**INT-248 DEEP LEARNING PROJECT**

**Vehicle image dataset Classification using Convolutional Neural Networks**

Name:-Putta Bharath

Reg.no:-11714590

Mail:-[putta.bharathnaidu@gmail.com](mailto:putta.bharathnaidu@gmail.com)

Mobile.No:-8074229802

**Abstract: Fine-grained vehicle arrangement is the assignment of grouping the make, model, and year of a vehicle. This is an exceptionally testing task since vehicles of various kinds however comparable tones and perspectives can frequently look significantly more comparative than vehicles of a similar sort yet extraordinary tone and perspective. Vehicle makes, model, and year in blend with vehicle tone - are of significance in a few applications, for example, vehicle search, re-distinguishing proof, following, and traffic investigation. In this work, we research the appropriateness of a few late milestone convolutional neural organization (CNN) structures, which have demonstrated top outcomes for huge scope picture characterization assignments, for the undertaking of fine-grained arrangement of vehicles. CNN full form is convolutional neural network. It is used in deep learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other.**

**1.INTRODUCTION**

As a rule, the visual fine-grained arrangement can be exceptionally testing because of more unobtrusive contrasts between classes, contrasted with fundamental acknowledgment or coarse order, for example, on Image-Net. Perceiving the makes and models of vehicles is one such assignment. For people, this is generally a genuinely direct errand, particularly for vehicle devotees. Vehicles can typically be recognized by the natural eye because of certain key angles, for example, logos, hood adornments, or lettering. Nonetheless, because of the visual multifaceted nature of vehicles, this has generally been a hard undertaking for PCs. The intensity of CNNs is their ability for learning loads of highlights as well as the highlights themselves too. As of late, these CNNs have accomplished the cutting edge exactness on conventional picture characterization. In this undertaking, we make broad utilization of CNNs as our essential design of classifiers.

**2.PROPOSED WORK:**

**2.1 DATASET:**

This dataset contains 16,185 picture order sets of 196 unique classes, part into 8,144 preparing and 8,041 test pictures. Every one of the 196 classes is exceptionally fine-grained on the request for year, make, and model of a vehicle. In spite of the fact that the classes are fine-grained, each class is outwardly particular from each other; for instance, the dataset contains a 2012 Volkswagen Golf and a 1991 Volkswagen Golf, which are outwardly unmistakable, generally, however it doesn't contain a 2011 Volkswagen Golf, which is basically indistinguishable from the 2012 model.

Dataset Link: <https://ai.stanford.edu/~jkrause/cars/car_dataset.html>

**2.2PREPROCESSING:**

There are a great deal of pictures of vehicles here with various models and they are generally being given a class freely. I could relegate them their own class and train my model dependent on that. In any case, that will take me quite a while and will most likely cost me a great deal of cash on any sort of cloud administration which I utilize, for example, an AWS or Azure. Along these lines, here is the thing that I will do:

Choose three vehicle organizations that I need to clasify and prepare a model to search for them. Three vehicles that I will prepare the model to search for are :

* Audi
* BMW
* Mercedes-Benz

**2.3MAIN MODEL**

In the code, I am making a limit called cars\_to\_label. This limit takes a pandas information outline and does the going with on that dataFrame. It kills all the areas which are not Audi, BMW, or Mercedes. Since there numerous models of Audi, BMW, and Mercedes and I am really keen on requesting a vehicle as either paying little notice to the model. Hence, this limit changes over these marked vehicles to the going with marks:

Audi : 0

BMW: 1

Mercedes : 3

Experience the results Data Frame and check the quantity of Audis were given the imprint 0, the quantity of BMW's was given the name 1, and the quantity of Mercedes was given the name of 3. By then segment, it by the full-scale number of the results data diagram entries and get the accuracy.

**2.4 REFERENCE OF MODEL:** In these reference of the main model is done in flowchart form using util.

* 1. **COMPILER:** Here code is built using model.compiler and finding accuracy and loss

**2.6 RUN MODEL:** By, using Epoch run the data how many times you want (initialized in epoch for suppose epoch=10 then from 1 to 10 loss and accuracy is printed.

**3.0 RESULT AND DICUSSION:**

## For this model, we fed it pictures for Audi, BMW, Mercedes and asked

## it to predict images as either cars after training on 1036 images with

## 345 images in validation set.

At the point when we tried this model, we found that:

• It anticipated 113 out of 589 pictures of Audis effectively, making it 19.185% exact for Audi.

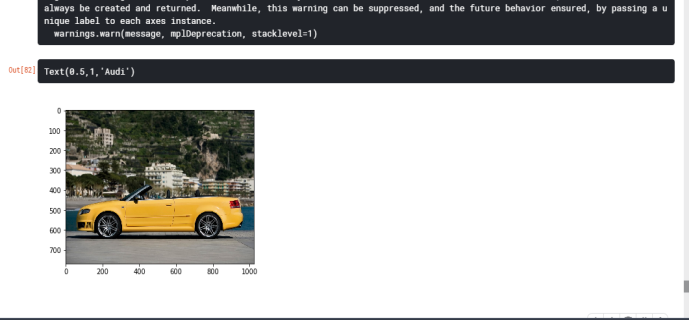
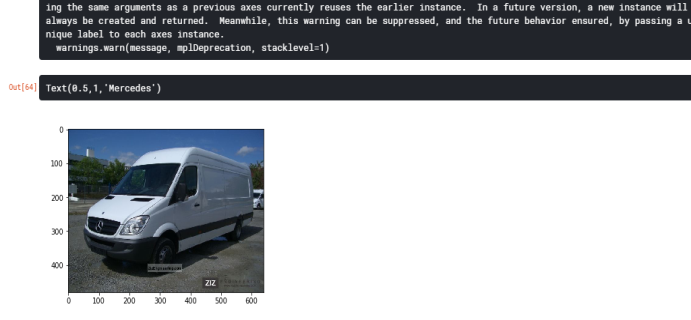
• It anticipated 423 out of 531 pictures of BMWs effectively, making it 79.661% precise for BMW

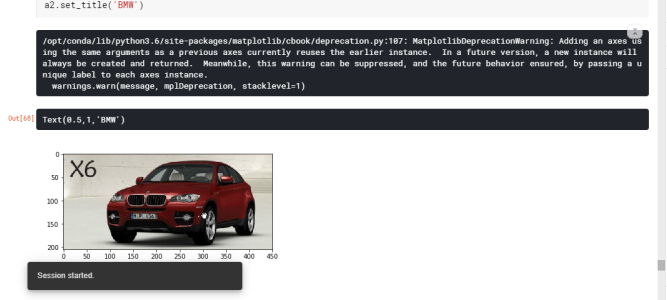
• It anticipated 151 out of 261 pictures of Mercedes effectively, making it 57.854% precise for Mercedes.

Not with standing, if we somehow happened to see this model from a brand freethinker perspective, it:

• It arranged 113 Audis, 423 BMWs and 151 Mercedes accurately out of 589+531+261=1381 pictures effectively.

• That makes this model 47.74% precise on this dataset**.**





WHY SUCH A LOW ACCURACY?

## The fundamental explanation that I can see is the size of the dataset that I utilized. The first dataset is 8144 pictures, yet I just centered around Audi, BMW, and Mercedes which weeded the dataset to 1036 pictures and that is simply insufficient to get all the subtleties. You would require a superior, more expanded dataset.

## WHAT CAN WE DO NOW?

• The obvious answer is get more pictures in the event that we simply need to zero in on these three brands.

• WE could likewise utilize Keras' inbuilt information growth methods to make our dataset better.

• We can tune the hyper boundaries, for example, clump size, the quantity of layers utilized.

**4.0 CONCLUSION**

CNN is inspired from biological process. It is popular techniques very dependent on size and quality of training data. Over fitting is reduced by using CNN model by using dropout. With CNN models accomplishing cutting edge results for fine-grained characterization undertakings, it turns out to be progressively critical to give developed and all around reported baselines to contrast new methodologies with. This work researches ongoing milestone CNN designs on the Stanford Cars-196 dataset. Because of the little size of the dataset, it was indicated that preparation a CNN without any preparation won't yield agreeable exactness. Adjusting of existing cutting edge structures prepared on Image-Net prompts much better outcomes which could be additionally improved through information growth.

Github Link : <https://github.com/bharathcan/Vehicle-Detection-using-CNN>