AI Task Theoretical part

- Convolutional Neural Networks (CNN) is used to solve the given task as CNN performs better on image classification and recognition. First Training dataset is created by scraping google images using python image scraper script. Approximately 235 images we used for each class (Audi cars: 235 , Benz cars: 236, BMW cars: 236). A subset of some training data is used to make validation dataset (107 images from all classes in total). Test set is created by randomly downloading 5 images per class. Data augmentation is performed to increase the size of the data. Confusion matrix is used to check the accuracy of the model.

  In order to obtain the most similar image, first intermediate vector has to be extracted for the training data and test data(or image from url) after the flatten layer from the network and then Cosine similarity is used to calculate the most similar image.

  Images from URLs has to be downloaded and temporarily stored as ImageDataGenerator() cannot load images from urls. Next matching probabilities can be predicted using the saved model from training. Then cosine similarity is used to find the closest image in the training data.

- Images from the URLs are different from the training images(which are mostly full pictures of cars) where as images from URLs are most of the time not complete picture rather a closeup or small part of the car. Training data size is also one of the challenging part of the task.

- Increasing the size and quality of the training dataset can help to improve performance. If the training images close to the real world data as in test images, helps to improve the performance. Increasing the dimensions of the images can be helpful. Hyperparameter optimization helps in finding the right architecture for this problem and data. Experimentation using Transfer learning techniques can be done. Checking for overfitting and underfitting conditions gives the idea how well the model is trained.