Vehicle Identification

Harsh Patel

Jill Hughes

Logan Murphy

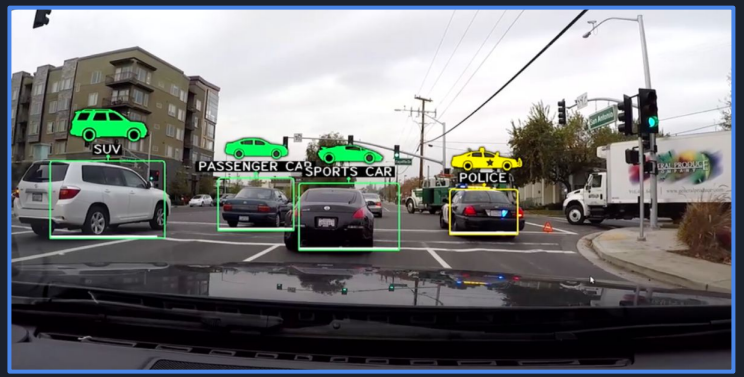
Mihai Anghel

Model for Vehicle Image Recognition

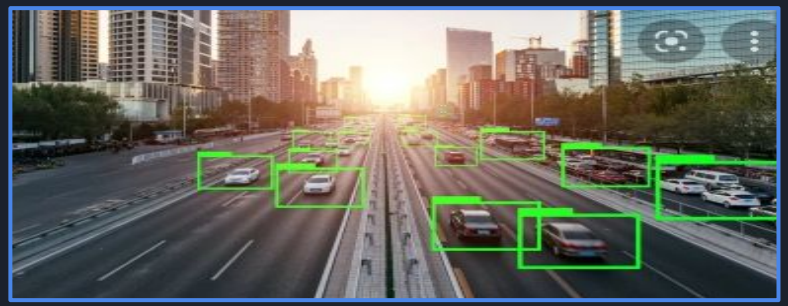
Why choose this topic?

Our team decided to pursue this topic for the various versatile utilizations a model such as this could have on a number of different transportation applications.

Applications such as monitoring traffic flow, automated parking systems and support for traffic law enforcement could integrate this machine learning model into their product to increase efficiency and results per their requirements.



Our Data



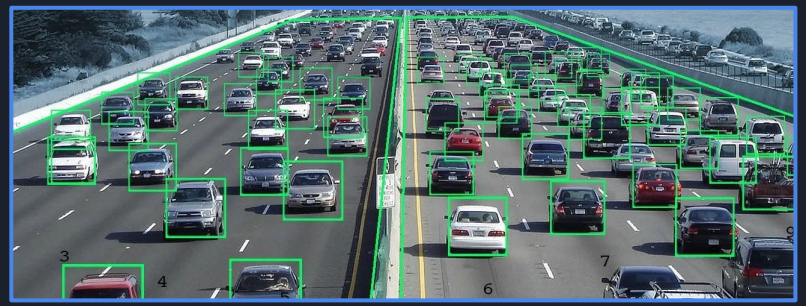
Our data source offers 16,185 images of 196 classes of cars. These are split between 8,144 training & 8,041 testing images, each class has been roughly 50-50 split.

The classes are at the level of Make, Model, Year
Ex: **2012 Tesla Model S** or **2021 BMW M3 coupe**

Citation:

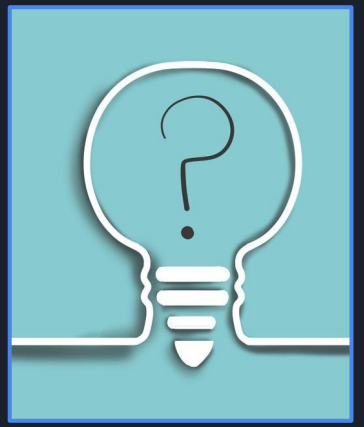
3D Object Representations for Fine-Grained Categorization
Jonathan Krause, Michael Stark, Jia Deng, Li Fei-Fei
4th IEEE Workshop on 3D Representation and Recognition, at ICCV 2013 **(3dRR-13)**. Sydney, Australia. Dec. 8, 2013.

Source: Stanford.edu ~jkrause Dataset



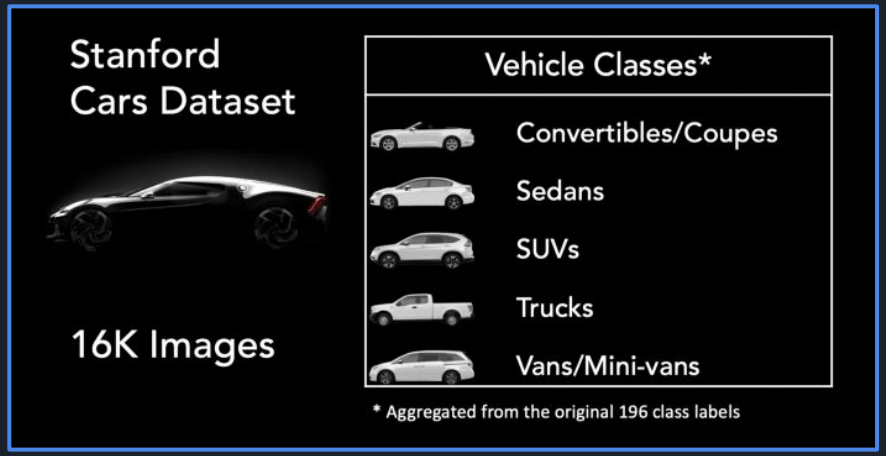
The primary question our team hopes to answer throughout the project:

➔ Can we take a dataset containing thousands of images, and create a model using machine learning in order to identify whether or not a vehicle is present in an image?



The second question our team hopes to answer throughout the project:

➔ Can the our prototype machine learning model identify the make, model and year of a targeted vehicle in an image?



Data Exploration

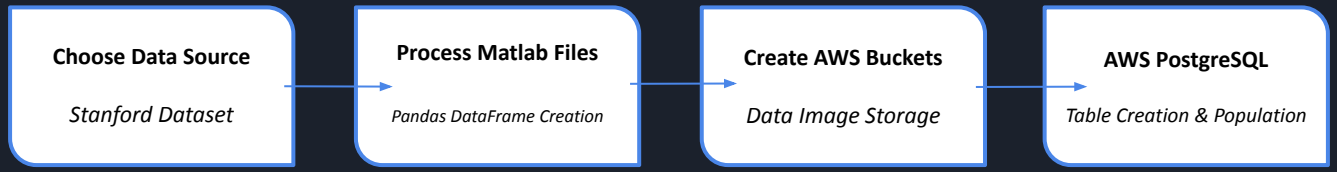
The Challenge

Unlike previous datasets we've encountered that were in CSV or Panda formats, this time our data came in MatLab format. The team had to find a way and transform this format into Pandas DataFrames. This would:

1. Allow its introduction into the AWS database
2. Allow effective use by our neural network model to process the images in an accurate and timely manner

The following outline explains the steps taken in order to clean, organize and set up our datasource that it can be used for analysis in our model

Data Preparation Outline



Dataset chosen for project presented here

Labels, df_train, df_test

Upload images to AWS buckets from PC using R3 Upload Tool

Creating and populating the tables to be used to run the Train/Test Machine Model

Analysis Phase (Part 1)

Initial Analysis

- Pandas was used to clean raw data from initial Matlab format
- Further analysis conducted with Python

PostgreSQL Database: How do we connect our data to the model?

- Read in database and create dataframe
  - Query merging images
  - Import dataset into model through query merging
- Images evaluated to determine what features will be captured
- Hosted on AWS

AWS R3 Storage Buckets

- Training images
- Testing images

Analysis Phase (Part 2 - Machine Learning Model)

Sequential (model)

- Hidden layers with activation set to reLu
- Using optimizer *adam* and *accuracy* metrics

SciKit Learn

- ML library for creating classifier

Neural Networks

- Popular choice for image processing in machine learning models
- Other choices included a Random Forest Classifier ML model
- More consistent concerning image classification
- More flexible for feature processing

Tested using cleaned data

- Query merging images
- IMport dataset into model through query merging
- Images evaluated to determine what features will be captured

AWS R3 Storage Buckets

- Training images
- Testing images

Two major packages/applications to consider: Tensorflow vs. Pytorch

- Both are successful image classifiers
- Tensorflow was the familiar choice
- Used to test Neural Networks Model

Put it into practice

- Read database and create dataframe
- Test model
- Update test image class & evaluate accuracy

How was data split into testing and training datasets?

- 50/50 split
- Adjusted as needed for the addition of future models

Looking ahead for final presentation…

Flask Application

- Created using pickle file of model
- Final deployed application will take an uploaded image and return: vehicle/ not vehicle

D3.js

- Fully functioning, interactive dashboard

Tableau

- Final deliverable
- Application hosted on Amazon Web Services

Tableau Blueprint

- Open with brief overview of process, database, and model
- Focus on README with link
- User input and results. Application will be done through a flask app
- A brief look at future plans
- Feedback