Predicting Car Make and Model

James Wang
CSE140 Artificial Intelligence
University of California, Santa Cruz
Santa Cruz CA USA
jwang402@email.com

Billy Kwong
CSE140 Artificial Intelligence
University of California, Santa Cruz
Santa Cruz CA USA
bqkwong@ucsc.edu

Vincent Thai
CSE140 Artificial Intelligence
University of California, Santa Cruz
Santa Cruz CA USA
vikthai@ucsc.edu

ABSTRACT

With the proliferation of cars in society, such has brought convenience and independence to the masses but with such adoption comes crime and deceit. As such, the ability to distinguish vehicles based on make, model, and year through vision has crucial applications in law enforcement and security. Such methods of vision based surveillance has been proven in China's wide adoption of such technologies for law enforcement. The use of computer vision to identify cars based on specific models can help in instances where license plates are not legible or valid. To solve this multiclass classification problem, our team has created a program with an underlying model that successfully identifies different cars based on make, model, and year. We utilized Stanford's extensive Cars Dataset of over 16 thousand unique images. Through experimentation and research, we found success through the implementation and use of the ResNet34 model, providing approximately 80% accuracy.

MOTIVATION AND OBJECTIVE

The goal of our project is to train a model utilizing the Stanford Cars Dataset to identify and classify cars according to make, model, and year given an image.
Other parties have utilized this same dataset in an attempt to create a project with similar intent; one group implemented a multi-task learning scheme where they were able to improve test accuracy on car models by at least 0.1% while simultaneously performing high accuracy classification on car make and car type. The same group also implemented a different approach in an attempt to improve accuracy. They used a compression method by switching over from a ResNet architecture design to MobileNetV2. They didn't achieve a better test accuracy as test accuracy deteriorated by around 1%, however, they were able to reduce the model size by 10x. In the interest of accuracy, there is consensus that a Resnet is among the better implementations.

DATASET

The dataset consists of 16,185 images of cars divided into 196 classes. 8,144 of the images are training images and 8,041 are testing images. Each class is a specific year, make, and model of a car; each of these classes is mapped to a numerical value spanning from 1 to 196. Each numerical class value is mapped to a string in the format [Make Model Year] i.e.: Hyundai Sonata 2012.

The data is labeled by a series of matlab files with the coordinates for a bounding box, its corresponding class number, and filename. We used loadmat from scripy.io to perform regex operations on the files. We then wrote a series of algorithms for sorting the images in directories with hierarchies based on class for training the Resnet.

The coordinates for the bounding box for each image in pixels are important as some images are car dealer advertisements with extraneous elements in the image such as dealer logos and other advertising elements and watermarks.

Due to the broad scope of this dataset, we experimented with limiting the scope by cleaning the dataset by removing images that were laden with elements such as images that had watermarks or text overlays, shadows, and obstructed images of cars. We found that this made negligible impact on accuracy when benchmarked with the testing data. For development purposes, we utilized a smaller random subset of the data to develop and test the algorithm, and fed the entirety of the validated dataset when the project is nearing completion. The images range in resolution, and after bounding, many were not perfect squares.

To preprocess the dataset for training, we first wrote a program to read in the matlab file containing the labels and created 196 directories for the 196 classes, and put each of the testing images into the corresponding folder whilst also cropping each image by the given bounding box in the label. This decreased the runtime of the training algorithm as it removed some irrelevant portions of each image and sorted the images by class. We also preprocessed the bounding box problem as opposed to doing so on the fly.

```
01470.jpg
<matplotlib.image.AxesImage at 0x7fb1384ced30>
```

Figure 1: **Original Image**

```
8041
```

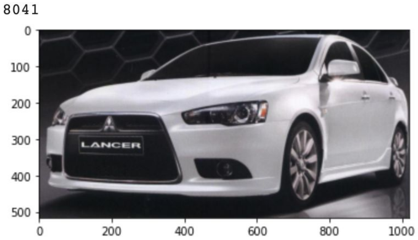


Figure 2: **Processed Image**

MODELS AND ALGORITHMS

This problem is a multiclass image classification project. Specifically, we are classifying images of cars into 196 classes; as such, we will be implementing a Convolutional Neural Network. We argue this project, to identify cars namely for law enforcement applications, is not a multi label classification project, as we are only interested in the specific model of a car for this project (each unique model is one class, and each unique model of car can only belong in one class), rather than a categorization such as a brand or a body style.

We considered other approaches such as specific feature extraction-identifying the car by its brand emblem for instance. This approach was not ideal since identifying which brand a car belongs to will not determine what the model of the car will be, and this might be computationally expensive and unnecessary. Also, there are cases that if an image of the car did not contain the brand emblem on the car, such as a top down or a side image of the car, then our model would not have worked: this model needs to be robust in its use case of being able to identify cars with non-ideal quality or angles as is the case in the real world.

We chose to implement this project with Resnet34. There is a philosophy that the deeper a network is, the better it will perform, which prompted us to want to implement a deep network. With classic CNNs, however, performance degrades after a certain threshold, so we wanted to approach our solution with transfer learning. Transfer learning is using the knowledge we gain from solving one problem and applying to a related problem. In our case, for example, analyzing the features of one car can be applied to another.

For our neural network, we implemented a resnet which allows for the training of deeper neural networks. The ResNet takes an image input and passes it through a convolution and pooling layer. The image input is first transformed to a perfect square and set to a fixed resolution. Then the images are all randomly rotated, flipped and shuffled. The image is then passed into 34 additional layers of similar convolution. Each layer performs a 3x3 convolution with a fixed feature dimension map of values ranging from 64, 128, 256, and 512. The algorithm uses skip connection to bypass the input every 2 convolutions. The general idea behind skip connection is that it adds the output from an earlier layer to a later layer which helps to mitigate the vanishing gradient problem by allowing an alternate shortcut path for the gradient to flow through. All of this also helps the network perform better by allowing the model to learn an identity function which will ensure that the higher layer will perform at least as good as the lower layers and not any worse. We trained the model based on the cropped images provided by our bounding box and utilized GPU hardware (NVIDIA CUDA) for the training.

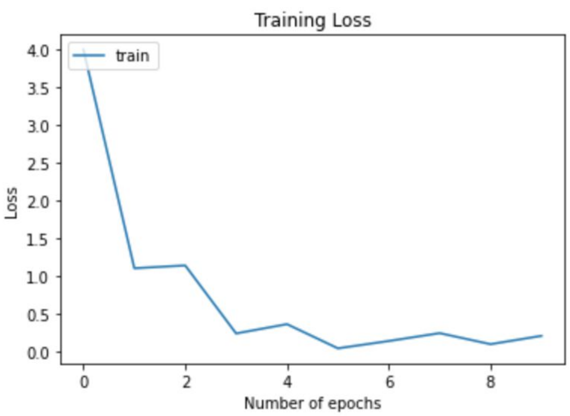


Figure 3: **Training Loss**

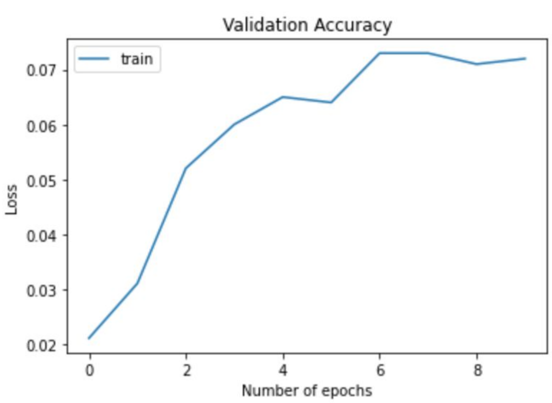


Figure 4: **Validation Accuracy**

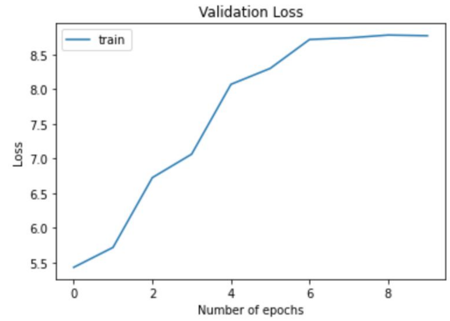


Figure 5: **Validation Loss**

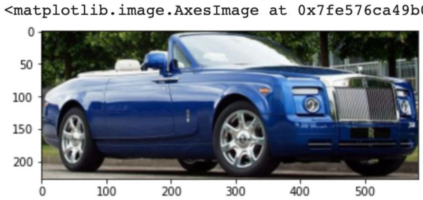
RESULTS AND ANALYSIS

Through the quarter, we were able to build and train a model that would classify images with their respective classes to an accuracy rate of 80%.

We were satisfied with the accuracy given the large number of unsatisfactory training images and the limited timeframe to work on the project given a late project change. We successfully achieved our goal of being able to identify different cars based on make, model, and year with a considerably high accuracy rate. Our project shows potential for improvement with further development given if we were to have more time to develop our model, and to train the updated model with more images and deeper networks.

```
Randomly chosen file to classify: /tmp/boundedTest/07431.jpg
Seconds elapsed for prediction: 0.12244701385498047
*******************************
FIRST GUESS:  Rolls-Royce Phantom Drophead Coupe Convertible 2012
*******************************
Might also be: Rolls-Royce Phantom Sedan 2012
Might also be: Ford F-450 Super Duty Crew Cab 2012
Might also be: BMW 1 Series Convertible 2012
Might also be: Spyker C8 Coupe 2009

<matplotlib.image.AxesImage at 0x7fe576ca49b0>
```



```
Randomly chosen file to classify: /tmp/boundedTest/00109.jpg
Seconds elapsed for prediction: 0.1431119441986084
******************************
FIRST GUESS:  BMW 3 Series Wagon 2012
******************************
Might also be: Daewoo Nubira Wagon 2002
Might also be: BMW 1 Series Coupe 2012
Might also be: BMW 3 Series Sedan 2012
Might also be: Acura Integra Type R 2001

<matplotlib.image.AxesImage at 0x7fe576ab0320>
```

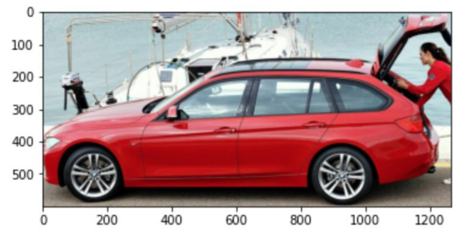


Figure 6,7: **Program Output**

CONTRIBUTIONS

All members met weekly to plan and discuss details on how to implement the project. Our group formed a group chat for the means of communication, and each member was easy to approach and communicate with. Communication was vital in the process of developing our project, as we were able to answer each other's questions and clarify any concerns that anyone had. Each member kept pace with any updates to the project. The project was hosted in a shared Google Drive with Colab, so the group could easily collaborate in the coding and debugging process. This also helped prevent issues arising in different development environments and can help keep track of who contributed to what modules in aiding debugging. In the process of constructing our final presentation video, we created a slide presentation where each member presented. For our slides, all members contributed to and presented the content. The edits to the final presentation video were made via iMovie.

FUTURE WORKS

We have landed a solid baseline of what the model, project, and dataset could achieve. For those who might want to build upon our current project, a deeper network in addition to a larger dataset could entail greater usability. It is clear that 196 classes is not enough to achieve our goal of helping identify all cars on the road for road enforcement. With every make, model and year being a unique class, the number of classes can seem endless. As society continues to migrate to the web, a good deal of car sales and research will likely shift to online retailing. A car image classification system could solve business issues such as the truth behind posted car images on peer-to-peer sales websites. As not many people are well versed in the details of specific car makes, further implementations of our project could address whether the images posted on these sales platforms are actually of the car that they specify. They can also address whether multiple exterior pictures actually represent the same car, or if they are using other images to mask the blemishes of the car they are actually selling. The car recognition model could also aid online retailing by recommending alternative cars that are available in an inventory that have similar looks/features and price. Of course, there are many other possibilities that can be achieved with further development other than online retailing. A simple classification of cars can help in identifying the fine-grained features important for 3D object detection for autonomous cars.

Image recognition is also becoming more prominent in society today which raises a lot of questions about the ethics of such a method. Image recognition methods installed in intersection cameras, etc, could be used to spy on the population which could raise many privacy issues. If work were to continue on this project and image recognition projects in general, regulations must be put in place to protect personal information. We must define a line of what is socially acceptable in the ownership of data. In the instance of intersection cameras, a solution could be to blur out license plates and potential faces during data collection.

REFERENCES

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

[2] Connor Shorten. 2019. Introduction to ResNets. *Towards Data Science*. https://towardsdatascience.com/introduction-to-resnets-c0a830a288a4

[3] Bhanu Yerra. Car Image Classification Using Features Extracted from Pre-trained Neural Networks. *Towards Data Science*. https://towardsdatascience.com/classifying-car-images-using-features-extracted-from-pre-trained-neural-networks-39692e445a14

[4] Grab AI For Sea Computer Vision challenge - Cars Dataset. https://github.com/kamwoh/Car-Model-Classification.

[5] Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. Bag of Tricks for Image Classification with Convolutional Neural Networks. https://arxiv.org/pdf/1812.01187v2.pdf.
[6] Ross Girshick, Jeff Donahue. Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. https://arxiv.org/pdf/1311.2524.pdf.
[7] Car Recognition. https://github.com/foamliu/Car-Recognition.
[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. *Cornell University.* https://arxiv.org/abs/1512.03385.