Module 5: Executive Summary

Roadsigns for Self-driven Cars Analysis

Presented by:

Joaquin Solis

Scott Townsend

Zach Holcomb

Isaac Weyland

I. Introduction, Data, and Insights

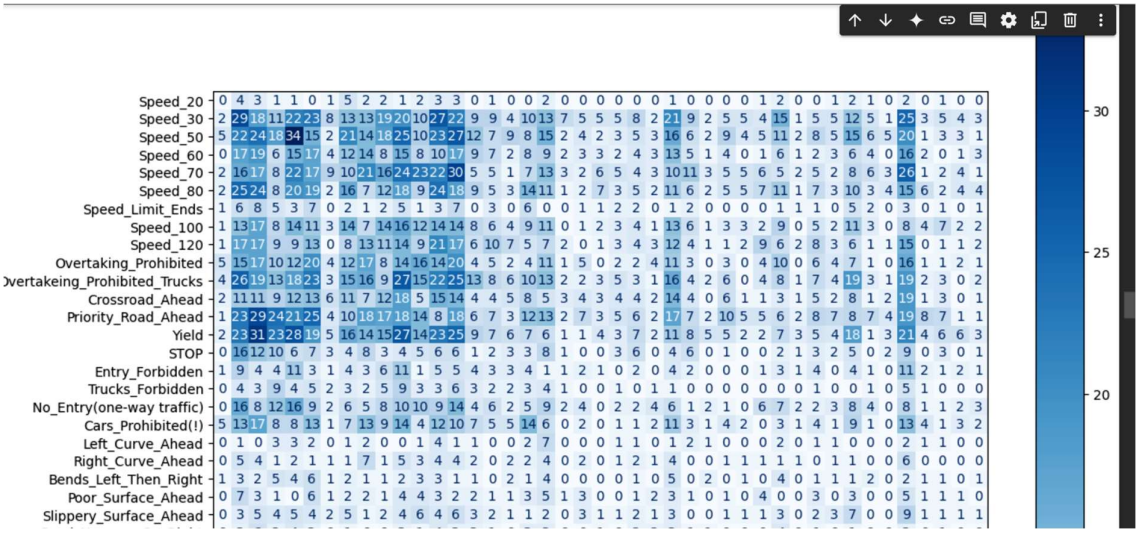
Road sign recognition is a cornerstone of intelligent transportation systems, enabling vehicles to accurately interpret and adhere to traffic regulations. This case study focused on creating and testing a Convolutional Neural Network (CNN) to classify German road signs into 43 distinct categories. The dataset consisted of RGB images organized into subdirectories matching to specific classes. Each image filename included a track number (indicating the physical sign) and a sequential number (indicating order).

While the model made significant strides in developing a reliable system, several challenges emerged, including optimizing model performance for robust and scalable classification. This case study highlights these challenges and explores solutions to improve the system's efficiency for real-world application in self-driving cars.

Some insights gained include data augmentation, including transformations like adding Gaussian noise and random rotations, which enhanced the model's ability to generalize by simulating real-world variations. Additionally, transfer learning with VGG16 demonstrated its potential to improve performance significantly, underscoring its value for image classification tasks.

Confusion Matrix

In this confusion matrix, we can see detailed insights into the types of errors the model is making by showing how often predicted labels match actual labels by measuring predicted and true labels. The purpose of this confusion matrix is to identify strengths and weaknesses in the classifier's performance across individual classes and highlight which classes are frequently confused, allowing us to focus on model improvements for those areas. Below is a snipped of our confusion matrix:



The darker and lighter colors represent the intensity or frequency of predictions for the classes. The darker colors indicate higher values (more predictions) while the lighter colors show the opposite. The shading essentially depicts how frequently the predictions are occurring. Looking at the confusion matrix, we can see that there are a plethora of light spots, which indicates how those specific areas may not be performing very well and if there was more time to focus on the model and its accuracy, these would be the spots to focus on.

II. ML Model

The initial approach involved building a custom CNN to classify traffic signs. However, due to limited performance, we transitioned to using a pre-trained VGG16 model, which proved more effective. Throughout the process, we tested various configurations using accuracy as a primary metric to evaluate changes and the overall performance of how well our model was fitting the data, and how well the model was doing in general.

Model Performance:
The VGG16 model we created achieved significant performance, **with accuracy levels exceeding 96%,** compared to the custom model's 90%-93%. Despite this success, challenges such as long training times (some epochs taking 3-4 minutes) and overfitting persisted. Addressing these issues required careful tuning and applying data augmentation techniques. One notable limitation was the model's ability to generalize beyond the training dataset.

Challenges: A major challenge in developing the model was how long it took to train it. We had to wait several minutes for each epoch we used with some attempts up taking to 20 minutes to complete which slowed down our progress in testing its performance. In addition, it was difficult to get the generators working properly. We also had problems with overfitting, the model would be good at the training data but fail to work as well with the holdout set.

Best Use Case: The model we produced could prove to be well suited for a configuration with the camera systems in self-driving cars. It can process live images of road signs, accurately classify them, and assist vehicles in making navigation decisions that ensure complete safety.

III. Python Notebooks

Notebook

IV. Discussion Responses

1. Will you be using an existing architecture as a starting point, or do you think it'll be better to design your own?

We recommend using **VGG-16** as a starting point due to its proven performance in image classification tasks. However, experimentation with other architectures or a custom-designed model may reveal a better fit for the specific dataset and problem at hand. Factors such as training time, accuracy, and adaptability to variations in the dataset will guide the final decision.

2. Do you think we need to do any preprocessing before using the data to train the model?

While the images appear to be in the correct format, additional preprocessing could enhance the model's robustness. Without preprocessing, the model may only recognize signs that closely resemble the training images. To address this, we could apply data augmentation techniques such as random cropping, translation, and zoom. These transformations simulate real-world variations, ensuring the model performs well in diverse conditions, such as different angles, lighting, and distances.

3. This seems like one of those cases where straight accuracy might not be the best metric for model evaluation. What do you think?
Accuracy alone may not fully capture the model's effectiveness in critical scenarios. An **F-measure** could be more suitable, as it balances the trade-offs between false positives and false negatives. For traffic signs, ensuring high recall for essential signs like stop signs is particularly crucial to avoid missed detections that could compromise safety.

4. What performance would you require before you'd ride in the car?
Perfect recall (1.0) for stop signs, as missing even one could have severe consequences.

V. Summary

This case study showed the value of good data preparation and the benefits of using transfer learning for traffic sign recognition. By adding data with random noise and rotations, the model became better at handling different real-world conditions and avoided overfitting. Switching from a custom CNN to the pre-trained VGG16 model greatly improved performance, achieving over 96% accuracy. These results highlight that using simple preprocessing steps, transfer learning, and practical evaluation methods like the F1 score can help build reliable systems for self-driving cars. Additionally, the confusion matrix displayed the areas that needed some improvement as they were performing poorly. This matrix can be used as a guide to enhance the model further and develop its capabilities.