GehirnWagen Self Driving Car

Executive Summary

Presented by

Jed Chang
Bridger Norman
Celeste Popoca
Luke Russell
Kelin Tang
Ella Yang

I. Summary

As data analysts for GehirnWagen self-driving car company, we received 43 unique road sign designs that we were required to identify and classify (A sample of signs are depicted below in figure 1). We fed this road sign data into a convolutional neural network tailored for image recognition in order to create the software for the cars. Our calculations lead us to get an Accuracy score of **97.1%**. This means our model was able to accurately identify 97% of all road signs. We also had precision and recall of **1** for all stop signs. With all of this information, we are confident that our self-driving cars will have the ability to recognize, classify and react to various road signs, especially danger and control signs in varying conditions.

(Figure 1)

II. Methodology

Process:

1. View images and class labels
2. Imagine preprocessing and Imagine augmentation
3. Create a vanilla model
4. Compared ResNet, DenseNet, and Inceptions model
5. Testing the model we create
6. Determined how well the model identify the image
7. Found out DenseNet model has the best prediction

We used the DenseNet model to create a base to compare how well we increased the probability of guessing outcomes correctly. To increase the accuracy we adjusted images' rotation,brightness and zoom range in the model. The model achieved an accuracy of **95.7%** on the validation set and **97.1%** on the test set, demonstrating its effectiveness in accurately classifying images. The training process utilized early stopping and model checkpointing to ensure the best performing model was saved. The model can be deployed for real-time image classification tasks, offering valuable insights and enabling automated decision-making processes. The implementation of data augmentation techniques helps in handling variations in real-world images, leading to improved generalization. Overall, the model shows great potential for enhancing operational efficiency and decision-making in image-related business processes.

III. Results, Action Items, and Limitations

We chose to test three different Convolutional Neural Networks (CNN): Inception, Vanilla, TensorFlow's Keras, and DenseNet. The Inception model had a Accuracy score of 0.98, the vanilla model had a accuracy score of 0.94, and the sequential API from TensorFlow's Keras model had a accuracy score of 0.93, we decide the DenseNet model had a high accuracy score of 0.97, which means it has roughly a 97% correlation with the variation in all the road signs [see Table 1]. This parameter comes from testing the neural network on a set of data the model was not previously trained on.

	DenseNet	Inception	Vanilla	TensorFlow's Keras
Accuracy	97.1%	97.01%	94.53%	93.2%
Precision	92%	86.37%	82.56%	81.43%
Recall	91%	86.88%	81.84%	80.25%
F1-score	0.94	0.93	0.92	0.90

(Table 1)

Note:

Accuracy: *Number of correct predictions out of Total number of predictions*
Precision: *"...How often, when a model makes a positive prediction, this prediction turns out to be correct" and "how confident we can be that an instance predicted to have the positive target level actually has the positive target level."*

Recall: *"...Tells us how confident we can be that all the instances with the positive target level have been found by the model."*

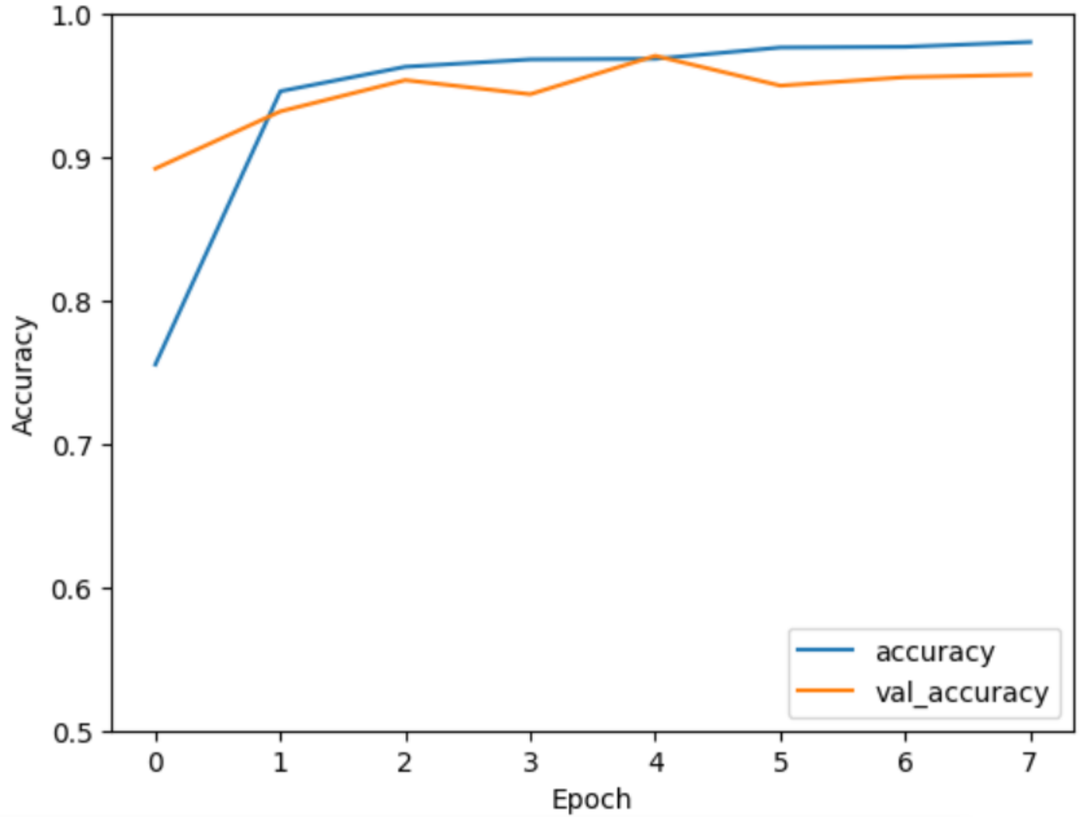
F1-score: *is calculated as the harmonic mean of precision and recall*

The range for precision and recall is 0 to 1. Higher numbers mean better model performance.

Based on these results, the CNN model that worked the best was the DenseNet architecture. This
demonstrates a promising future for self-driving cars in the company. The model can provide reasonably accurate estimates for determining what each road sign means. It will assist the company in providing safe self-driving cars and aid them in decision-making processes on the road.

A. Action Items

Based on the results, GehirnWagen should continue to accumulate more road sign data. It would be especially helpful if it was taken from a car's point of view. Most self-driving cars have cameras to look at the environment around them. The data should include driving in different day times, areas, and weather conditions (lighting).



(Figure 2)

B. Limitations

One of the many limitations we had in our model development was the lack of data similar to the real conditions the car would experience on the road. The data size and variability was increased by changing, for example, the angle and brightness of the images. However, some of the new may never be encountered in a real situation.

We also were limited by time. There are convolutional neural networks that already exist. However, the model could have been more accurate if we had time to include architectures of other networks. Not only could this make our network more efficient, but also specific to identifying roads signs.

IV. Q&A

What data augmentation strategy would you suggest to improve the performance of the model in this problem?

Our group has implemented a DenseNet architecture and employed data augmentation techniques to improve the model's performance on this problem. Data augmentation involves applying transformations like rotations, flips, zooms, shifts, and changes in brightness to the training data, increasing its diversity and helping the model generalize better. DenseNet, known for its dense connections between layers, efficiently captures intricate patterns and facilitates gradient propagation. By combining data augmentation with DenseNet, our model becomes more robust to variations in the dataset and gains a deeper understanding of image patterns. This approach aims to enhance the model's accuracy and overall performance on the classification task.

V. Python Notebooks

Below is Github Gist link to the notebook we used during this case study:

https://colab.research.google.com/gist/jluke-russell/d4c96f92d7298ccb85dd8b127f7ea0c6/copy-of-starter_signs_v2_student.ipynb

VI. References:

[1] Kelleher, J. D., Mac, N. B., & D'Arcy, A. (2015). Fundamentals of machine learning for predictive data analytics: Algorithms, worked examples, and case studies. MIT Press.