Image Classification of German Traffic Signs Using Capsule Neural Network and Convolutional Neural Network

Project Report

Kartik Jain

Department of Data Science

The George Washington University

DATS_6203_10: Machine Learning II

Amir Jafari Dec 13, 2022

Introduction

In this project, we explored Capsnet Neural Networks (CapsNet) and compared its performance with a Convolutional Neural Network (CNN). Both models were fine-tuned to perform image classification on German Traffic Sign Image Dataset and were evaluated using precision, recall, and f-score. Lastly, we implemented Local Interpretable Model-agnostic Explanations (LIME) to understand how the input features of our model affects its predictions.

As a general overview of this paper, we will begin by discussing the results from our Exploratory Data Analysis (EDA). Next we will provide background information on CapsNet and describe the models we used in our experiment. Then, we will discuss our results and explainability. Finally, we will conclude with our findings and areas of improvement.

Interpretation with LIME

Evaluating models based on accuracy is not always sufficient; hence, we sought ways to explain and interpret our model's behaviors to understand the reasons behind why certain classes were predicted. LIME, as mentioned before, stands for Local Interpretable Model-agnostic Explanations. This tool explains a model by approximating the local linear behavior. It generates new data points around a given instance and uses the model to classify them. Then, it weighs the data points based on their distance to the original instance and fits them using a linear classifier. This can tell us whether increasing or decreasing a value can change the prediction. After running its algorithm, LIME masks a portion of the image in green if it has a strong positive correlation and contributes greatly to the prediction and masks the image in red if it has a negative correlation and contributes trivially to the prediction.

To avoid any bias, we built code that randomly chooses an image and utilized the lime package to generate masked images. Below are the results from using LIME on four random images that were predicted by the CapsNet and CNN model. Figures, a and b, are organized as follows: top left is the original image, top right is the segmented image, bottom left is the masked image using CapsNet model, and bottom right is the masked image from CNN model.

Looking at our result, we can see that the CapsNet and CNN model correctly predicted both images. However, when we take a closer look at the highlighted regions, some of the results are unexpected. In figure 7, both models use the image outside the traffic sign to classify the image. Intuitively, this does not make sense. In figure 8, the CapsNet model again uses the image outside the traffic sign; the CNN on the other hand correctly uses the inside of the traffic sign to make the correct prediction.

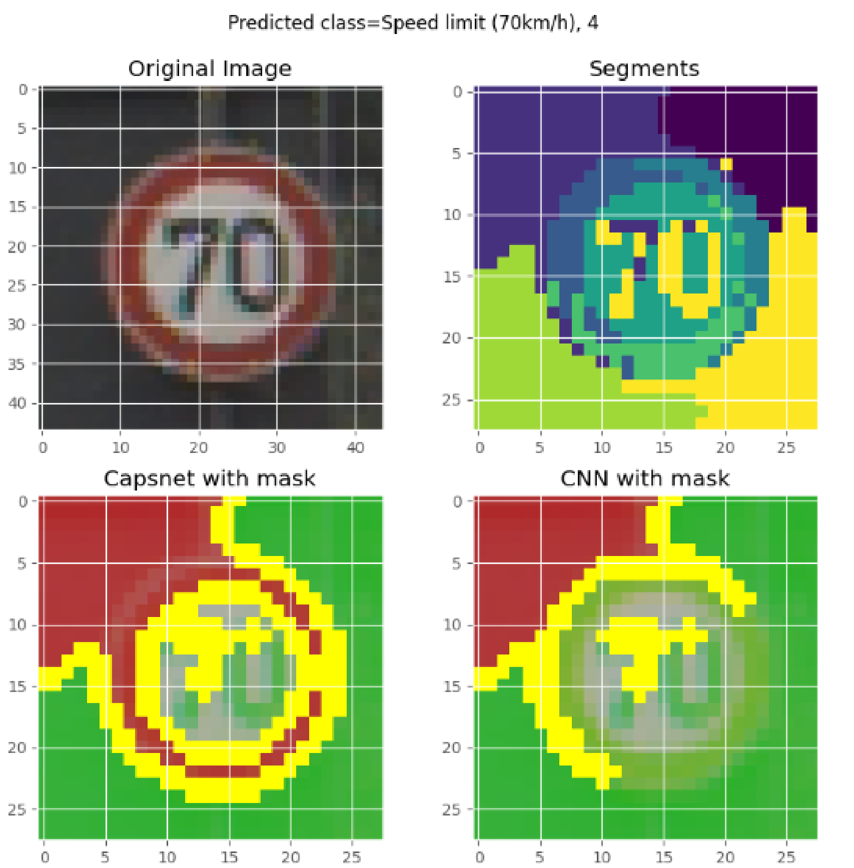
Predicted class=Speed limit (70km/h), 4
Original Image
Segments
Capsnet with mask
CNN with mask

Figure 6

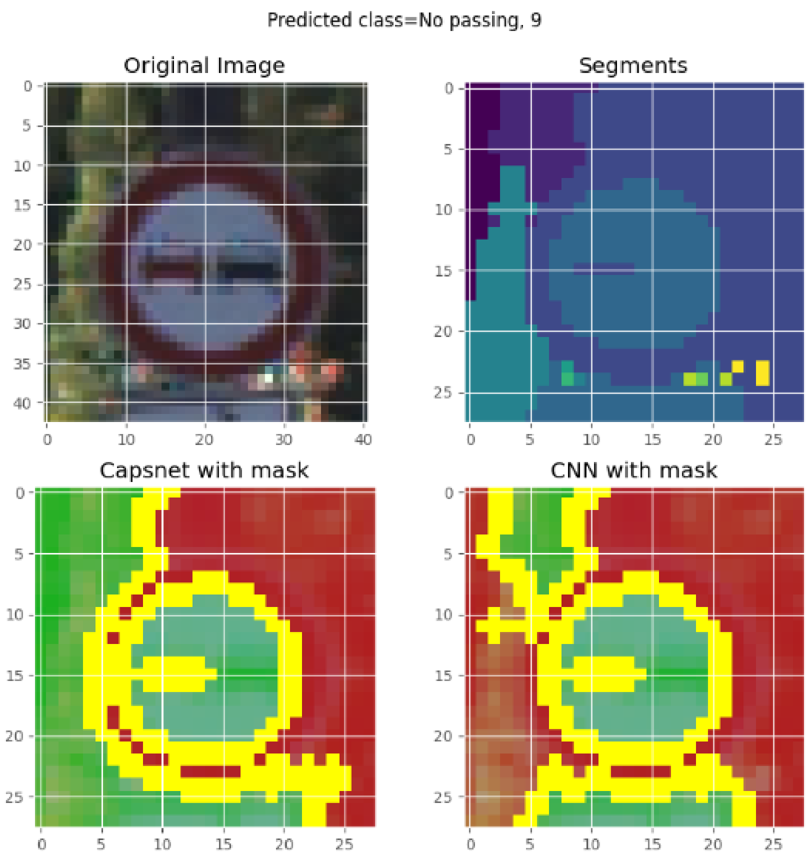
Predicted class=No passing, 9
Original Image
Segments
Capsnet with mask
CNN with mask

Figure 7

Summary and Conclusion

This project focused on comparing the performance of a Capsule Neural Network and a Convolutional Neural Network in classifying the German Traffic Sign Data. CapsNet was fine tuned by changing batch size, epochs, learning rate, and momentum. It was also fine-tuned by changing activation functions, number of in channels, kernel size, and number of dynamic routing. The CNN was built using the same number of parameters and was fine tuned by changing the batch size, epoch, learning rate. In the end, we saw that the CapsNet did not outperform the CNN, since they had similar accuracy and f1-scores.

After implementing LIME, we were able to see which regions were impacting the model prediction positively and negatively. The results were unexpected and can be further analyzed in future work.

Although we did not see improved results in the CapsNet, it is still important to keep them as a tool. It may be the case that CapsNets works more efficiently on certain types of dataset than others. For example, it has excelled in predicting healthcare images such as histological and retina images.

References

- https://pechyonkin.me/capsules-1/
- GitHub - jindongwang/Pytorch-CapsuleNet: An easy-to-follow Pytorch implementation of Hinton's Capsule Network
- https://christophm.github.io/interpretable-ml-book/lime.html
- https://proceedings.neurips.cc/paper/2017/file/2cad8fa47bbef282badbb8de5374b894-Paper.pdf
- https://eudl.eu/pdf/10.4108/eai.13-7-2018.158416
- https://arxiv.org/abs/1807.07559