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

Pavani Samala

1. Introduction

In this final project, we conducted an experiment to compare the performance of a Capsule Neural Network with a traditional CNN. We built and fine-tuned both models to perform image classification on a German Traffic Sign dataset and explained their predictions using LIME.

2. Background and Description of Individual work

For my contribution, I worked on understanding the Capsule Neural Network architecture and assisted in repurposing previously built code to align with out dataset. CapsNet is a complex model that tries to preserve the spatial information of features that is otherwise lost in a CNN. I adjusted the code to account for the number of classes is our dataset and the dimensions of our image. I also worked on creating a CNN model that could be trained using the same dataset.

Another portion of our project was using tools to explain our model. I initially worked on SHAP; however, we later learned that it is not compatible with CapsNets. SHAP works by using prior knowledge as a base value and adding the present data's features to understand its impact. It calculates SHAP values for each feature, telling us how much each feature impacts the model's decision.

Much of the other work I did was building the PowerPoint and writing the report.

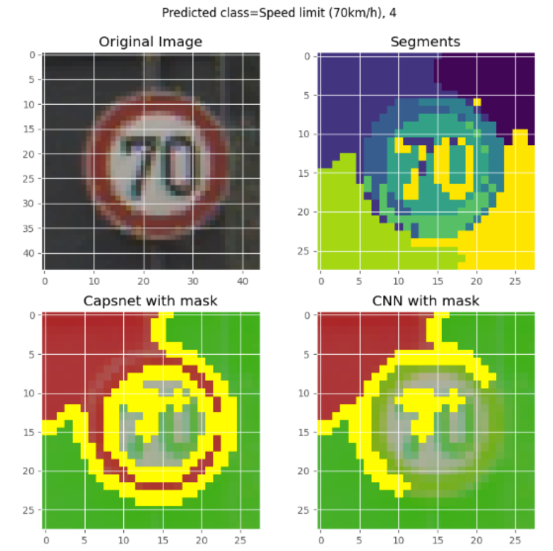
3. Results

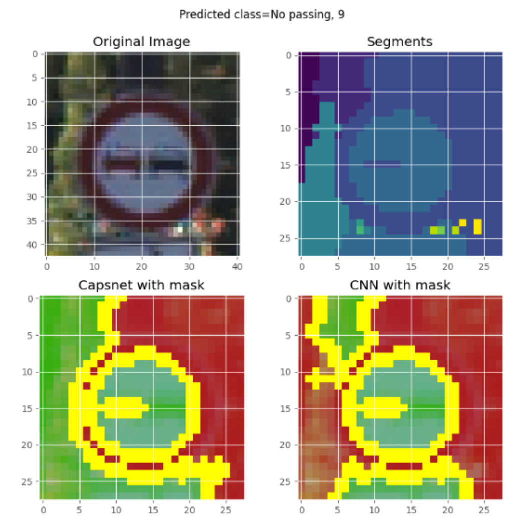
The tables below summarize each model's loss, accuracy, and f1-score for each number of epochs that was tested. You can see that the CapsNet and the CNN model approximately had the same accuracy and f1-scores. CapsNets, however, has a lower loss for all epochs tested.

Model	Epoch	Validation Loss	Validation Accuracy	Validation F1-Macro
Capsule Networks	8	4.6e-3	0.943	0.903
	9	4.5e-3	0.947	0.913
	10	4.4e-3	0.948	0.914

Model	Epoch	Validation Loss	Validation Accuracy	Validation F1-Macro
Baseline CNN	8	0.265	0.94	0.91
	9	0.236	0.943	0.91
	10	0.249	0.947	0.923

The figures below show a two random LIME samples. Since the green region positively impacted model's prediction and the red negatively impacts the model's prediction, some of the results were unexpected. Both the CapsNet and CNN used the background in their predictions rather than the traffic sign in the sample analyzed on the left. This does not make sense. The CapsNet does the same for the sample in the right; however, the CNN correctly uses the inside of the traffic sign to use for the image classification.





5. Summary

From conducting our experiment, we saw that the CapsNet did not perform better than the CNN. The accuracy and f1-score were approximately the same around 95% and .92. While explaining our model, we saw that both models had difficulty identifying the correct region at times. Although we don't see extraordinary results from the CapsNet, it can still be a good tool to have depending on the dataset used.

6. Code

I took 206 lines of code, modified 10, and added 6, so (206-10/206+6) x100 = 92%

7. Reference

https://pechyonkin.me/capsules-1/
https://eudl.eu/pdf/10.4108/eai.13-7-2018.158416
https://github.com/jindongwang/Pytorch-CapsuleNet