Capsule Neural Networks

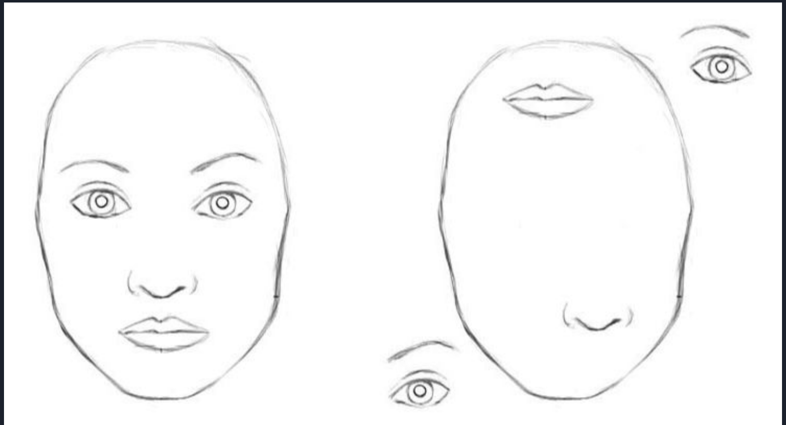
By: Siddharth Das, Kartik Jain, & Pavani Samala

Overview

- CNNs and their drawbacks
- Capsule Neural Nets
- Experiment/Methodology
- Explainability (LIME)
- Result/Conclusions

CNNs and their drawbacks

- Convolutional Neural Networks are the standard for image modeling
- Lower layers detect edges and color gradients and higher layers create complex combinations of the simple features
- Multiple stacked CNN layers and Max-pooling layers help increase the field of view
- Pooling Layers cause loss of information
- CNNs do not take into account important spatial hierarchies between simple and complex objects



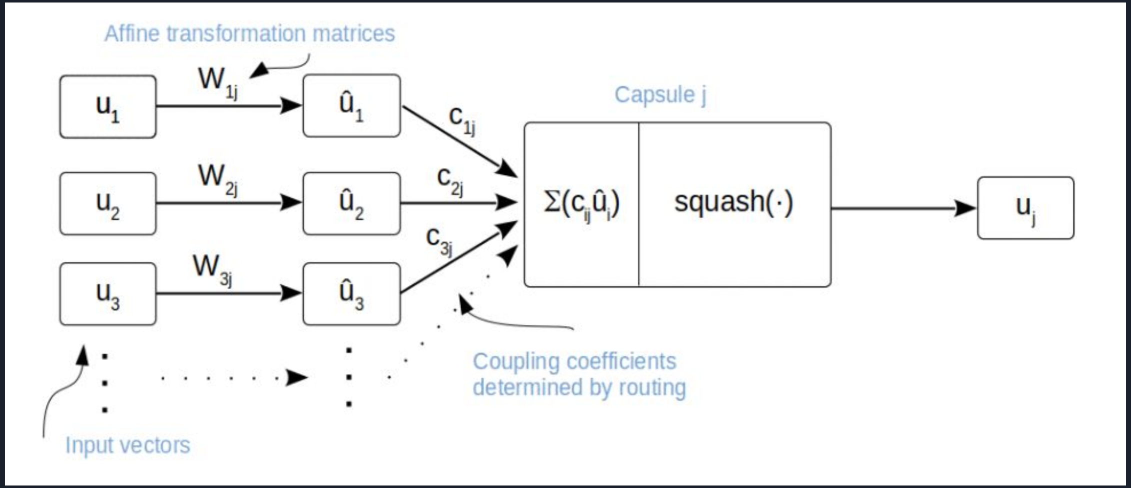
Capsule Neural Network

- Developed by Geoffrey Hinton and his team to overcome the shortcomings of CNNs
- Capsule: group of neurons that outputs a vector that encapsulates spatial information about a specific object or part of an object
- MNIST , CIFAR10
- Key Points: Dynamic Routing, Inverse Image Rendering, & Equivariance

How a Capsule Works

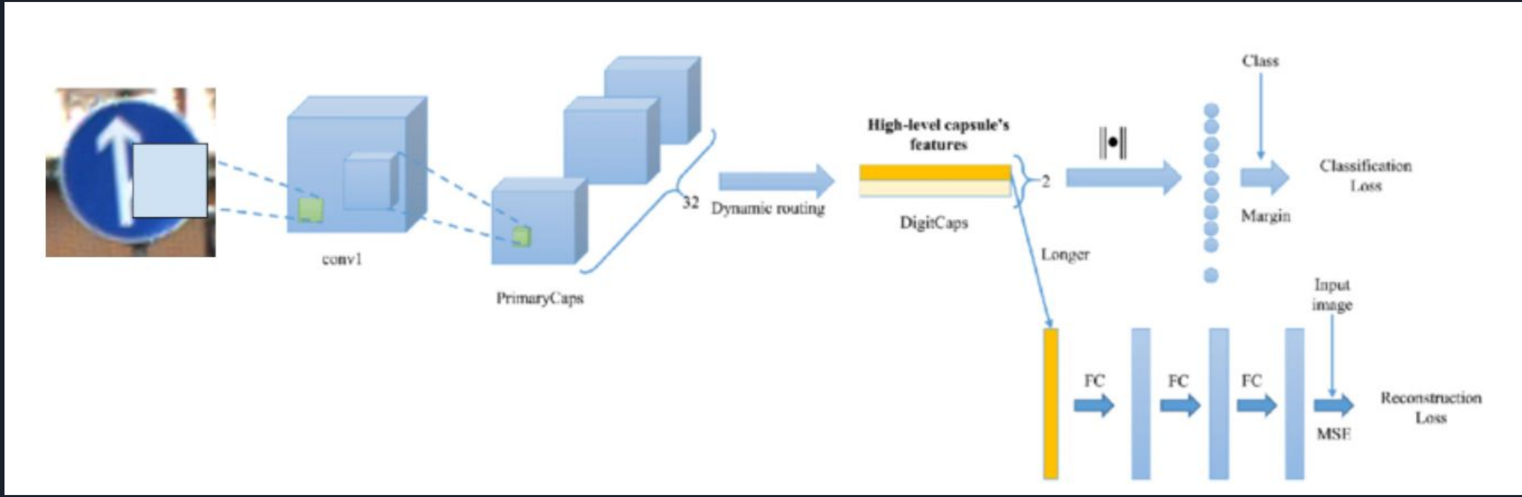
- Matrix Multiplication of Input Vectors
- Scaling Weights with Dynamic Routing
- Summing
- Squashing

$$v_j = \frac{\|s_j\|}{1+\|s_j\|^2} \cdot \frac{s_j}{\|s_j\|}$$



CapsNet Architecture

- Encoder: convolution layer, PrimaryCaps, and DigitCaps
- Decoder: 3 fully connected layers
- Total Loss = Margin Loss + *alpha* *reconstruction Loss

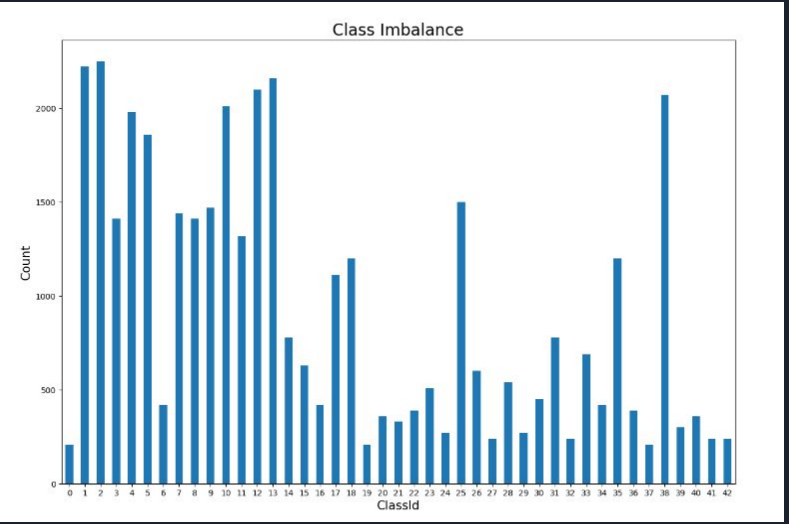


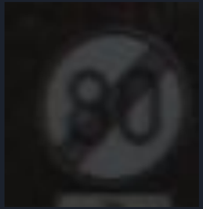
Experiment Scope

- Our objective was to train the same model on a traffic signs dataset and observe model performance
- Tuned hyperparameters and and model configs to get the best results
- Metric comparison against a baseline CNN architecture
- Explain the results from both models using LIME

Dataset

- [Dataset: German Traffic Sign Benchmark]
- More than 50k images and 43 classes
- Some labels include speed limit, stop signs, traffic lights, traffic signals, etc.
- Classes are not equally distributed

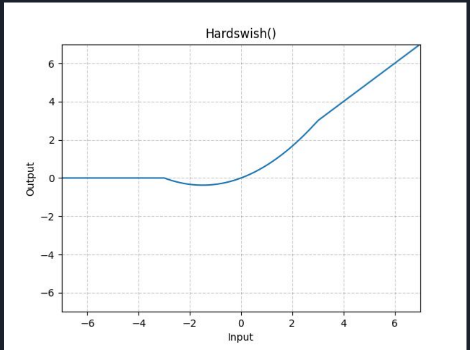




Fine-Tuning

Below are a list of hyperparameters that we tuned:

- Batch sizes: 16, 32, and 64
- Number of epochs: 5, 10, and 25
- Learning Rate: 1e-3, 1e-4, 1e-5
- Momentum with RMSProp optimizer: 0.5, 0.9

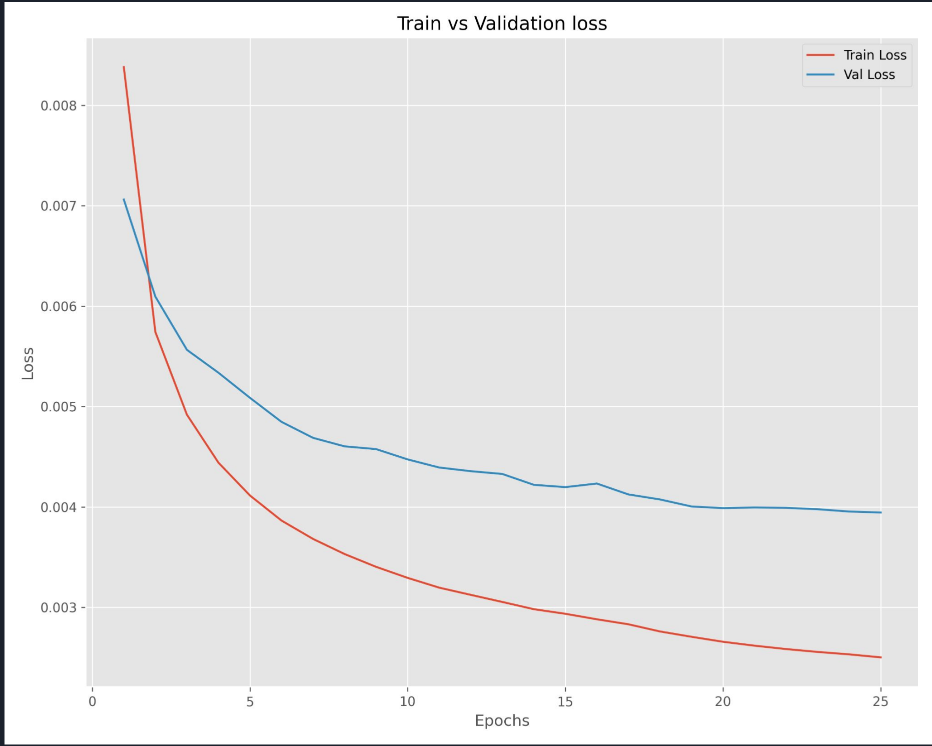


Model configs that we tuned

- Extra softmax (decoder)
- Hardswish activation function (decoder)
- Number of channels (Conv and Primary Capsule layer)
- Kernel size (Primary capsule layer)
- Number of dynamic routings (depends on kernel size)

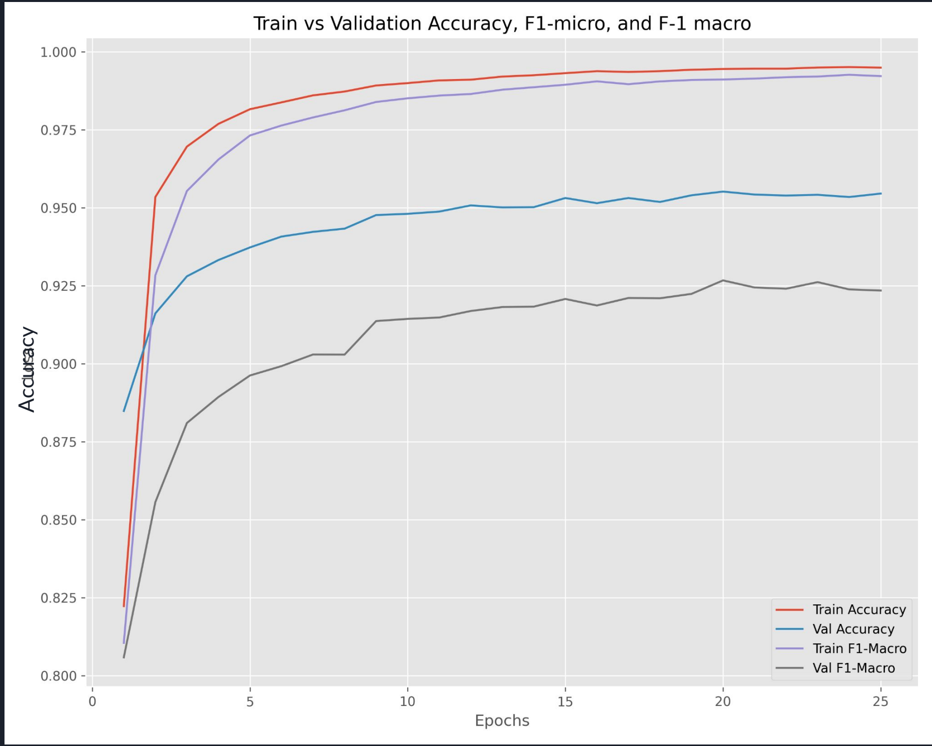
Model Results

Train and Validation Loss for Capsule Network model



Model Results

Train and Validation Accuracy and F-1 macro scores for Capsule Network model



Baseline CNN architecture

- We compared our model with a simple CNN architecture as a baseline
  - Three Convolutional layers with 32, 32, and 64 kernels
  - 2D max-pooling with size 2 after each Convolutional layer with
  - Dropout layers with 50% probability
  - A fully connected layer of size 256
  - Output layer with log softmax activation function
- We used the same parameters between capsnet and CNN to do our experiment
- Learning rate, number of epochs, optimizer

Post-Hoc Analysis

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

Explainability with LIME

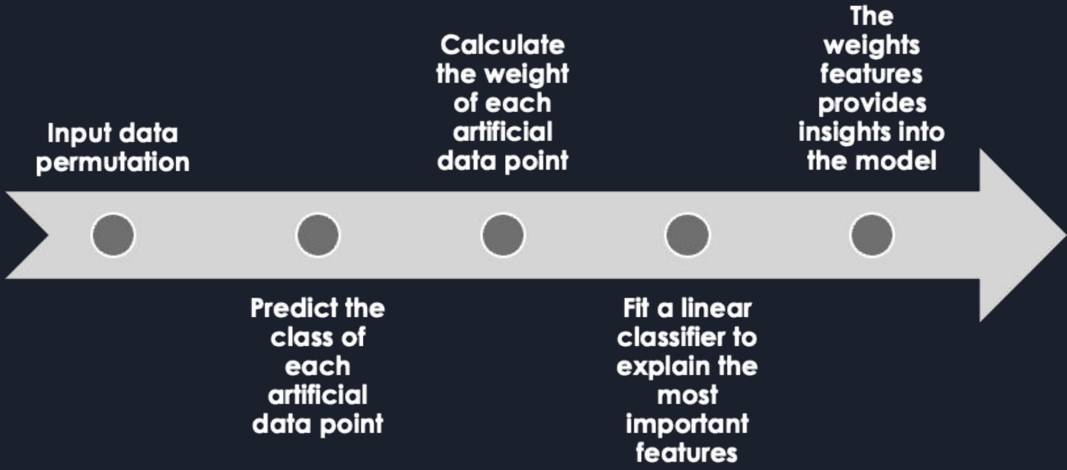
Original Image

Sample1

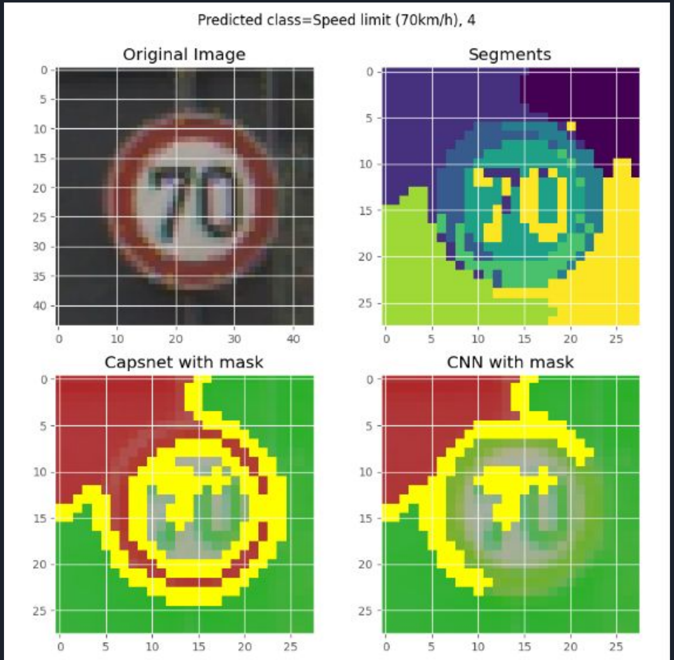
Sample 2

Sample 3

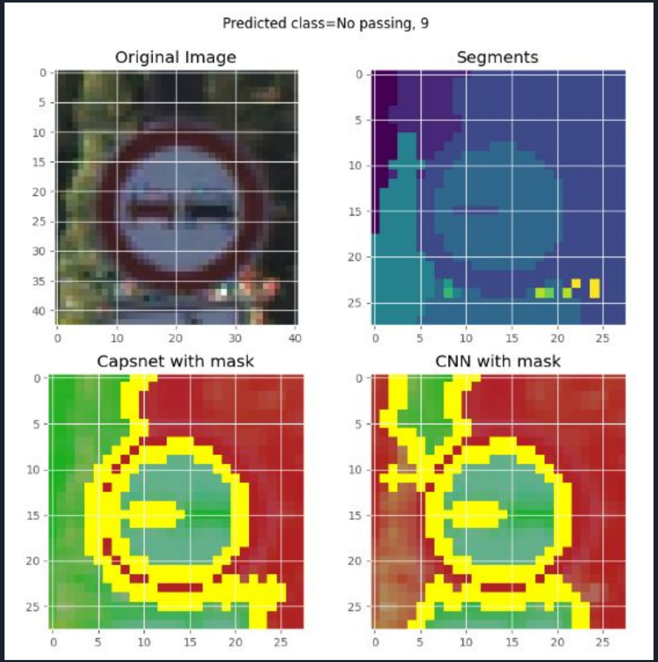
Generated Dataset Ex



Explainability Results



Example 1



Example 2

Conclusion

- CapsNet did not outperform the baseline CNN
- Would work better for Healthcare Image Classification
  - More clutter
  - Fewer colors
- There is potential for more exploration to justify the use of capsule networks

References

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