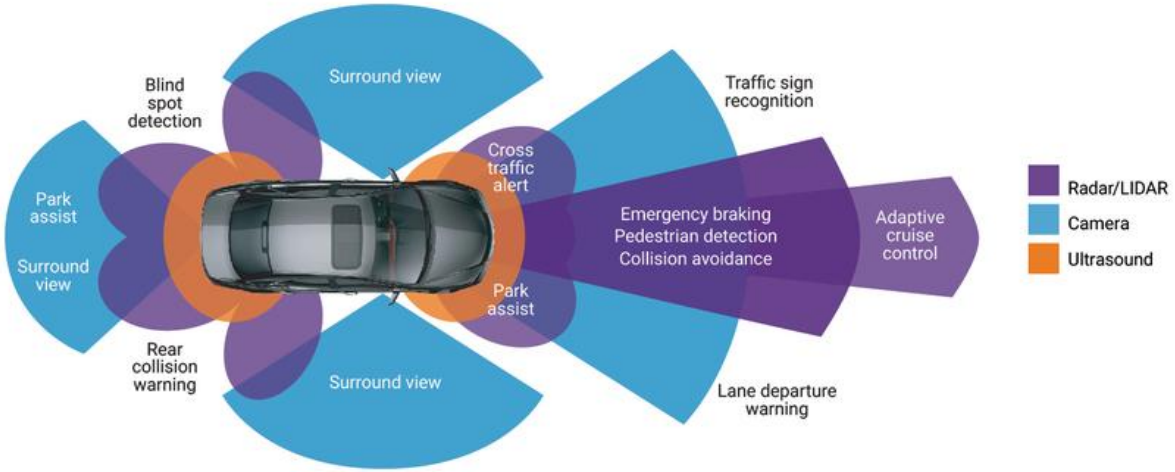
Traffic Sign Recognition Using Deep Neural Networks

By: Darrel Belen and Oleksandr Shashkov

Advanced Driver Assistance Systems (ADAS)

- Prevent deaths/injuries by reducing the impact of car accidents which can be avoided

- Several applications make up ADAS
  - Automatic Braking
  - Blind Spot Detection
  - **Traffic Sign Recognition**



ADAS in a modern vehicle [8]

Goals

- Explore deep learning network architectures and evaluates their performance on traffic sign recognition
- Combine networks architectures into an ensemble
- Evaluate performance of ensemble compared to its components (models)

General Approach

- Gather results from previous work
- Create multiple networks that are structured differently from the network found in background material
- Evaluate performance of networks
- Create an ensemble of networks by combining network architectures
- Evaluate performance of ensemble

Previous Work

- “Deep Learning Approach for U.S. Traffic Sign Recognition” [2]
  - Exploring the capabilities of the traffic sign recognition system
  - Focuses on US traffic sign data sets
  - Provides structure of model being tested

- Network Architectures
  - “Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning” [1]
  - “MobileNetV2: Inverted Residuals and Linear Bottlenecks” [7]
  - “Xception: Deep Learning with Depthwise Separable Convolutions” [3]
  - “Very Deep Convolutional Networks for Large-Scale Image Recognition” [6]

Data

Subset of pictures from the “German Traffic Sign Detection Benchmark” [4]

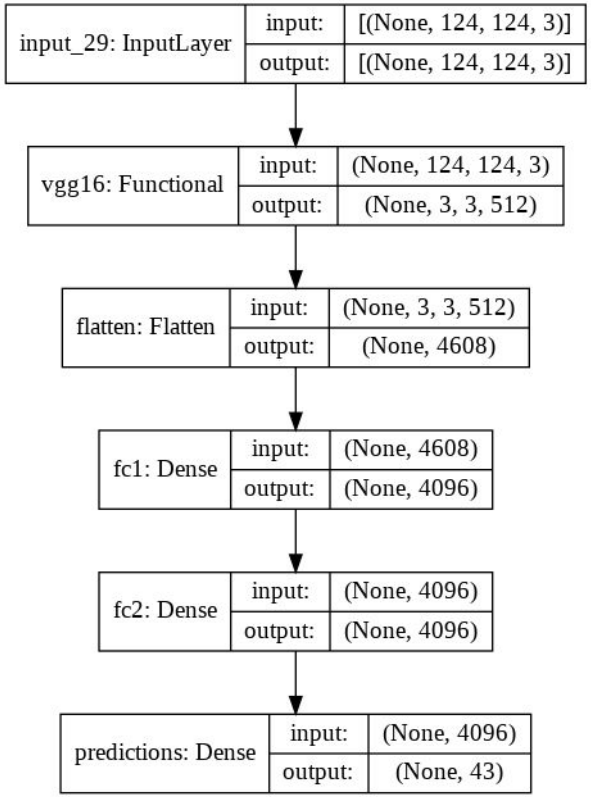
- Single-image pictures of German traffic signs
- 43 classes
- 861 total images
- One directory per class, each directory contains sample images

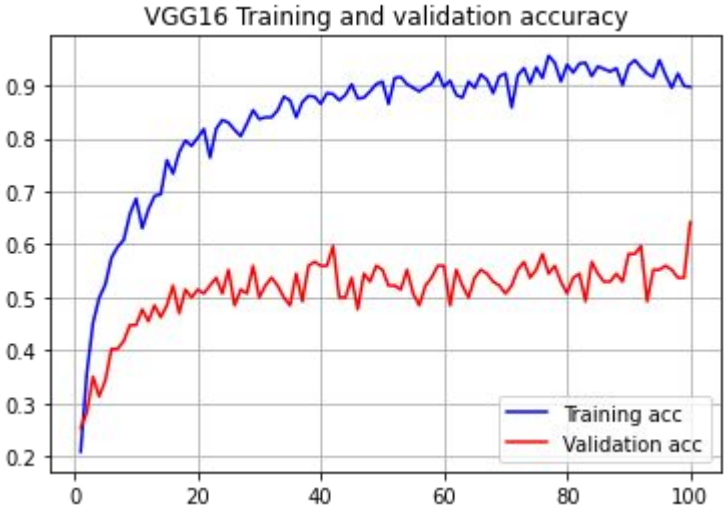
Traffic sign subset from Kaggle [5]

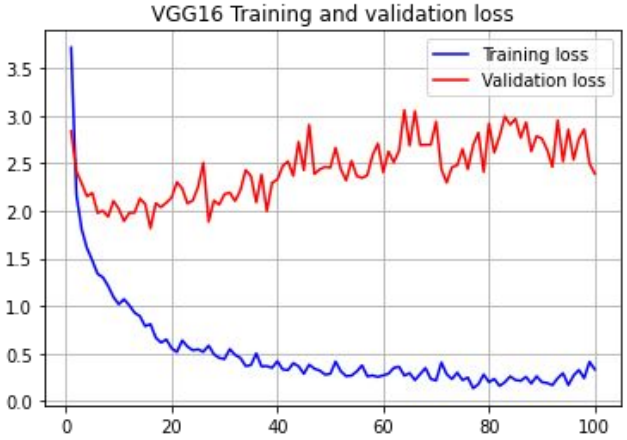
Implementation

- Four convolutional network architectures (found in Keras)
  - VGG16
  - MobileNetV2
  - Xception
  - InceptionResNetV2

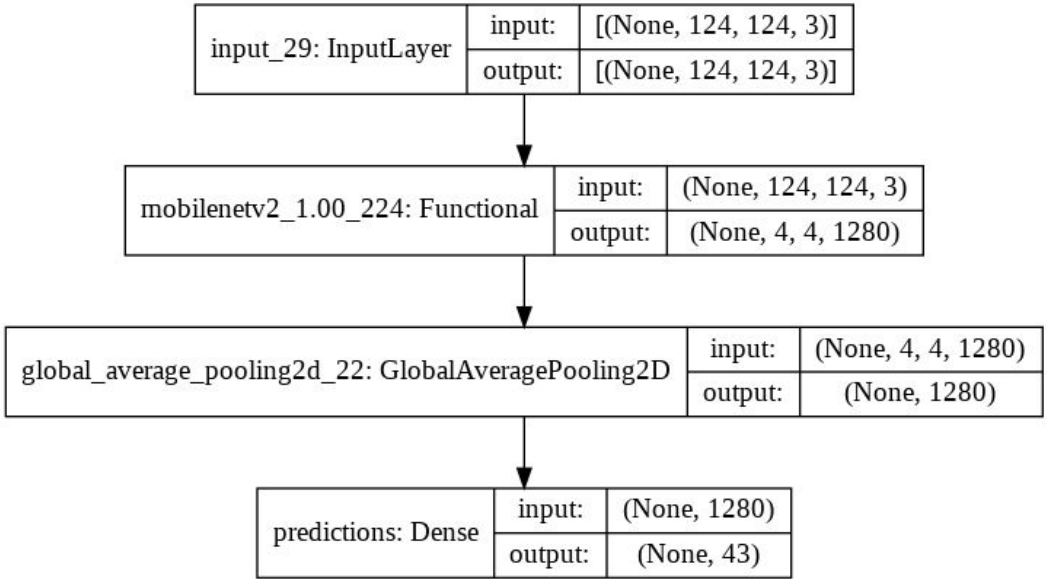
VGG16

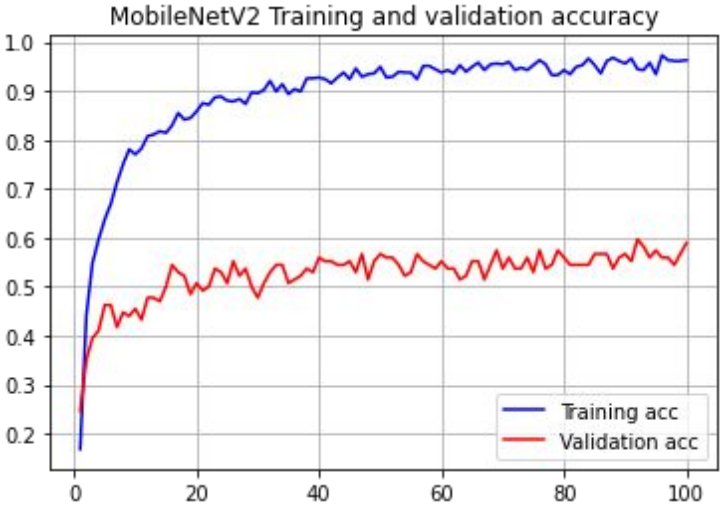

input_29: InputLayer
input: [(None, 124, 124, 3)]
output: [(None, 124, 124, 3)]
vgg16: Functional
input: (None, 124, 124, 3)
output: (None, 3, 3, 512)
flatten: Flatten
input: (None, 3, 3, 512)
output: (None, 4608)
fc1: Dense
input: (None, 4608)
output: (None, 4096)
fc2: Dense
input: (None, 4096)
output: (None, 4096)
predictions: Dense
input: (None, 4096)
output: (None, 43)

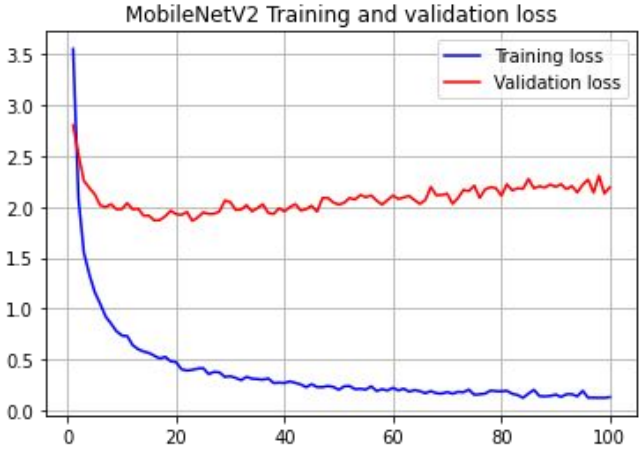

VGG16 Training and validation accuracy
Training acc
Validation acc
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0
20
40
60
80
100


VGG16 Training and validation loss
Training loss
Validation loss
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
20
40
60
80
100

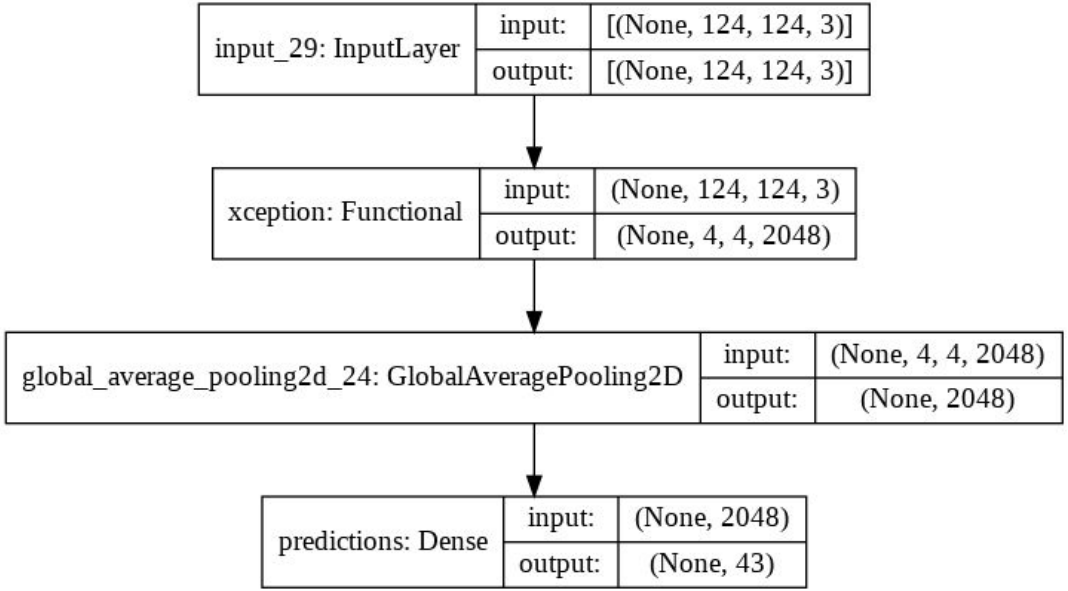
MobileNetV2

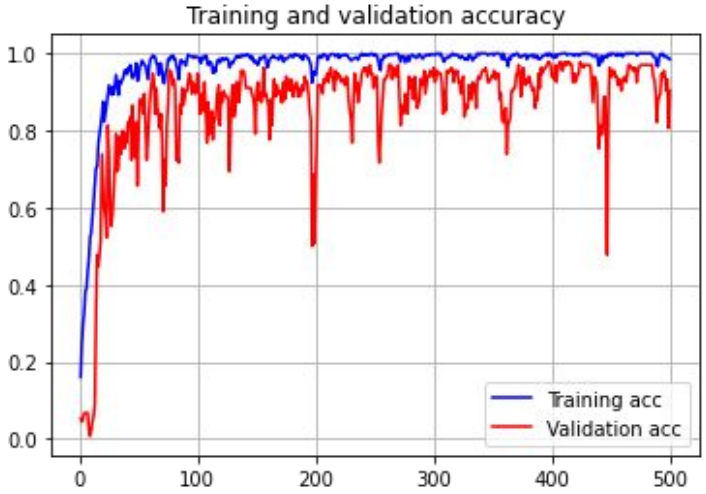


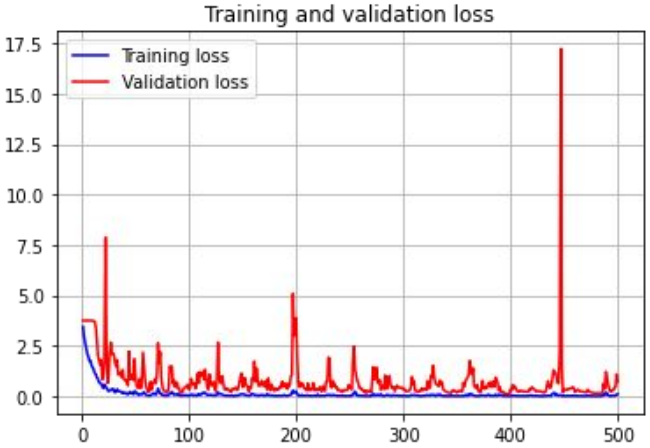




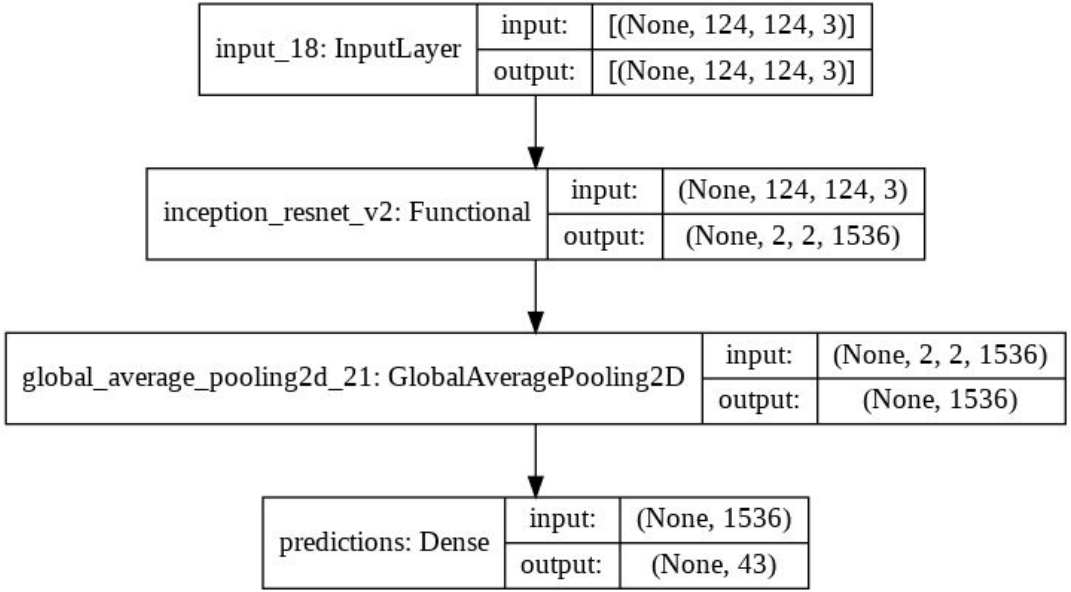
Xception

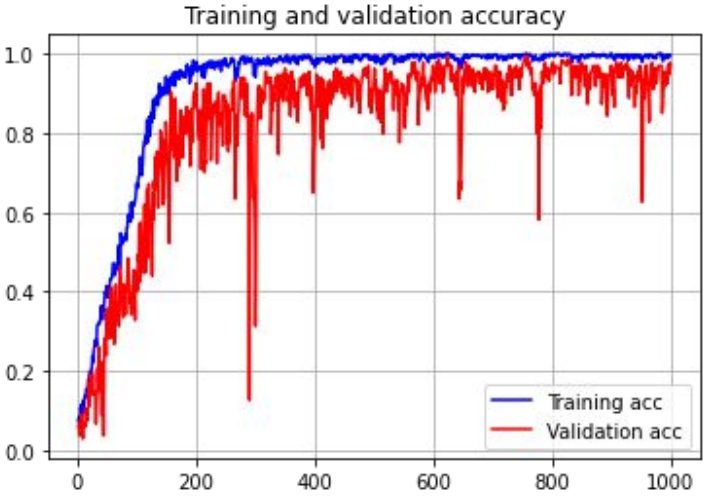

input_29: InputLayer
input: [(None, 124, 124, 3)]
output: [(None, 124, 124, 3)]
xception: Functional
input: (None, 124, 124, 3)
output: (None, 4, 4, 2048)
global_average_pooling2d_24: GlobalAveragePooling2D
input: (None, 4, 4, 2048)
output: (None, 2048)
predictions: Dense
input: (None, 2048)
output: (None, 43)


Training and validation accuracy
Training acc
Validation acc
0.0
0.2
0.4
0.6
0.8
1.0
0
100
200
300
400
500


Training and validation loss
Training loss
Validation loss
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
0
100
200
300
400
500

InceptionResNetV2





Analysis

- VGG16 and MobileNet did not learn on limited the dataset we had available
- Xception and InceptionResNet demonstrated outstanding learning capabilities and achieved 90%+ accuracy on the same data
- Ensembles demonstrated accuracy comparable with Xception and IncpetionResNet as standalone networks.

Conclusion

- Evaluated the performance of known deep network architectures
- Compared performance of ensembles with individual models
- Highly accurate network members dominate ensemble output
- Traffic Sign Recognition
    - Use an individual model with high accuracy score OR
    - Use an ensemble of less accurate, but less computationally costly members.

References

[1] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi, “Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning,” *Cornell University,* 2016, Available: arXiv.org, https://arxiv.org/abs/1602.07261. [Accessed April 28, 2021].

[2] Emmanuel B. Nuakoh, Kaushik Roy, Xiaohong Yuan, and Albert Esterline, “Deep Learning Approach for U.S. Traffic Sign Recognition,” in *ICDLT 2019: Proceedings of the 2019 3rd International Conference on Deep Learning Technologies, July 2019*, pp. 47-50

[3] François Chollet, “Xception: Deep Learning with Depthwise Separable Convolutions,” *Cornell University*, 2016, Available: arXiv.org, https://arxiv.org/abs/1610.02357. [Accessed April 25, 2021].

[4] The German Traffic Sign Detection Benchmark: https://benchmark.ini.rub.de/gtsdb_news.html

[5] GTSDB - German Traffic Sign Detection Benchmark. https://www.kaggle.com/safabouguezzi/german-traffic-sign-detection-benchmark-gtsdb

[6] Karen Simonyan and Andrew Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” *Cornell University*, 2015, Available: arXiv.org, https://arxiv.org/abs/1409.1556. [Accessed April 25, 2021].

[7] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” *Cornell University*, 2018, Available: arXiv.org, https://arxiv.org/abs/1801.04381. [Accessed April 25, 2021].

[8] Synopsys, “What is ADAS?,” *Synopsys*, 2020, Available: synopsys.com, https://www.synopsys.com/automotive/what-is-adas.html. [Accessed March 30, 2021].

Thank you very much!