Dataset

Overview

- Single-image, multi-class classification problem
- More than 40 classes
- More than 50,000 images in total
- Large, lifelike database
- Reliable ground-truth data due to semi-automatic annotation
- Physical traffic sign instances are unique within the dataset (i.e., each real-world traffic sign only occurs once)

Structure

The training set archive is structures as follows:

- One directory per class
- Each directory contains one CSV file with annotations ("GT-<ClassID>.csv") and the training images
- Training images are grouped by tracks
- Each track contains 30 images of one single physical traffic sign

Image format

- The images contain one traffic sign each
- Images contain a border of 10 % around the actual traffic sign (at least 5 pixels) to allow for edge-based approaches
- Images are stored in PPM format (Portable Pixmap, PX)
- Image sizes vary between 15x15 to 250x250 pixels
- Images are not necessarily squared
- The actual traffic sign is not necessarily centered within the image.This is true for images that were close to the image border in the full camera image
- The bounding box of the traffic sign is part of the annotations (see below)

Annotation format

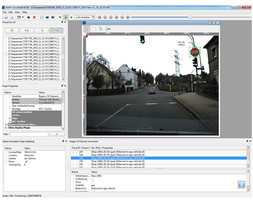
Annotations are provided in CSV files. Fields are separated by ";" (semicolon). Annotations contain the following information:

- **Filename**: Filename of corresponding image
- **Width**: Width of the image
- **Height**: Height of the image
- **ROI.x1**: X-coordinate of top-left corner of traffic sign bounding box
- **ROI.y1**: Y-coordinate of top-left corner of traffic sign bounding box
- **ROI.x2**: X-coordinate of bottom-right corner of traffic sign bounding box
- **ROI.y2**: Y-coordinate of bottom-right corner of traffic sign bounding box

The training data annotations will additionally contain

- **ClassId**: Assigned class label

Important note: Before January 17, 2011, there were some errors in the annotion data. These errors **only** affected the the *Width* and *Height* information of the whole image (being off 1 pixel in one or both directions), **not** other columns like *ClassId* or the *ROI* information. Thanks to Alberto Escalante for pointing this out. The annotations have been fixed and are included in the training image archive. For those of you who already downloaded the image date set, we provide a ZIP file which contains only the updated annotation data only.



The annotations were created with *Advanced Development & Analysis Framework (ADAF)* by Nisys GmbH

Result format

The results will be submitted as single CSV.

It contains two columns and no header. The separator is ";"(semicolon).

There is no quoting character for the filename.

First columns is the **Image Filename**, second column is the assigned class id.

The file must contain exactly one entry per element of the test set.

Example:

```
00000.ppm 4
00001.ppm 22
00002.ppm 14
00003.ppm 1
00004.ppm 5
00005.ppm 2
.....
```

Pre-calculated features

To allow scientists without a background in image processing to participate, we several provide pre-calculated feature sets. Each feature set contains the same directory structure as the training image set. For details on the parameters of the feature algorithm, please have a look at the file Feature_description.txt which is part of each archive file.

HOG features

The file contains three sets of differently configured HOG features (Histograms of Oriented Gradients). The sets contain feature vectors of length 1568, 1568, and 2916 respectively. The features were calculated using the source code from http://pascal.inrialpes.fr/soft/olt/. For detailed information on HOG, we refer to:

N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. IEEE Conference on Computer Vision and Pattern Recognition, pages 886-893, 2005

Haar-like features

The file contains one set of Haar-like features. For each image, 5 different types of Haar-like features were computed in different sizes for a total of 12 different features. The overall feature vector contains 11.584 features.

Hue Histograms

For each image in the training set, the file contains a 256-bin histogram of hue values (HSV color space).

Code snippets

Matlab

The Matlab example code provides functions to iterate over the datasets (both training and test) to read the images and the corresponding annotations.
Locations where you can easily hook in your training or classification method are marked in the code by dummy function calls.
Please have a look at the file `Readme.txt` in the ZIP file for more details

C++

The C++ example code demonstrates how to to train a linear classifier (LDA) using the Shark machine learning library.
This code uses the precalculated features. It was used to generate the baseline results.
Please have a look at the file Readme.txt in the ZIP file for more details

Python

The Python example code provides a function to iterate over the training set to read the images and the corresponding class id.
The code depends on matplotlib. Please have a look at the file Readme.txt in the ZIP file for more details

Citation

The data is free to use. However, we cordially ask you to cite the following publication if you do:

J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. The German Traffic Sign Recognition Benchmark: A multi-class classification competition. In *Proceedings of the IEEE International Joint Conference on Neural Networks*, pages 1453–1460. 2011.

```
@inproceedings{Stallkamp-IJCNN-2011,
    author = {Johannes Stallkamp and Marc Schlipsing and Jan Salmen and Christian Igel},
    booktitle = {IEEE International Joint Conference on Neural Networks},
    title = {The {German} {Traffic} {Sign} {Recognition} {Benchmark}: A multi-class classification competition},
    year = {2011},
    pages = {1453--1460}
}
```

Thank you.

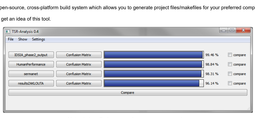
Result Analysis Application

We provide a simple application to facilitate result analysis. It allows you to compare different approaches, analyse the confusion matices and inspect which images were classified correctly.
The software is supplied under GPLv2. It depends on Qt 4.7, which is available here in source code and binary form. Qt is licensed under LGPL. Qt is a trademark of Nokia Corporation.

The software is provided as source code. The files can be found in the download section. The code is platform-independent, however, it has only been tested on Microsoft Windows with Visual Studio. So there might be a couple of issues left where GCC is more strict than Visual Studio. We appreciate any comments, patches and bug reports.

The project uses CMake, an open-source, cross-platform build system which allows you to generate project files/makefiles for your preferred compiler toolchain.

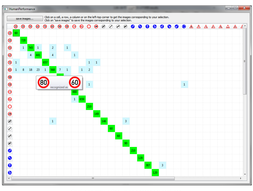
Here are some screenshots to get an idea of this tool.



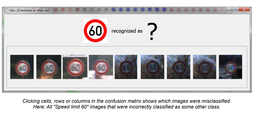
Main window: Performance of one or more approaches

Compare multiple approaches

Compare multiple approaches and on which images they erred



Confusion matrix: See which classes got confused.



Clicking cells, rows or columns in the confusion matrix shows which images were misclassified.
Here: All "Speed limit 60" images that were incorrectly classified as some other class.

Downloads

The GTSRB dataset is available via this link.

Code

- Example code for Matlab to read all training and test images including annotations: Download
- Example code for C++ to train a LDA classifier using the Shark machine learning library: Download
- Example code for Python to read all training images: Download

Result analysis application

The code is platform-independent, however, it has only been tested Visual Studio. So there might be a couple of issues left where GCC is more strict than Visual Studio.
We appreciate any comments, patches and bug reports. The code uses CMake, an open-source, cross-platform build system which allows you to generate project files/makefiles for your preferred compiler toolchain.

- Readme.txt
- Source code: tsr-analysis-src.zip (503 kB)

Make sure to check the news page regularly for updates. If you sign up, you will be notified about important updates by email.

Acknowledgements

SPONSORED BY THE

Federal Ministry of Education and Research

We would not have been able to provide this benchmark dataset without the extensive and valuable help of others.
Many thanks to Lukas Caup, Sebastian Houben, Lukas Kubik, Bastian Pietzka, Stefan TenBAUß, Marc Tschentscher for their annotation support, to Sebastian Houben for providing the Matlab code samples, Lukas Kubik and especially Bastian Pietzka for creation of this web site.

Furthermore, we thank Nisys GmbH for their support and for providing the Advanced Development & Analysis Framework.