# SoC Summer Workshop Project

## Traffic Sign Recognition


Traffic-sign recognition (TSR) is a technology by which a vehicle is able to recognize traffic signs on the road e.g. "speed limit" or "children" or "turn ahead". This is a very important technology in self-driving cars.

This project will give you the chance to train different models using various features to classify traffic signs.

The project is divided into 3 difficulty levels. *Beginner, Expert* and *Bonus*.

### Beginner Level:

For this level we use the Chinese Traffic Sign Database (Traffic Sign Recogntion Database (ia.ac.cn)). This is available as a zip file in your project folder under the name *"Dataset_1.zip".*

This dataset consists of 5998 images belonging to 58 classes. Each image is named "XXX_yyyy.png". Here XXX represent the class (traffic sign type) and yyyy represents the image number within each class.

For the beginner level, we make use of the "starter.py" code, which you can find in the project directory.

Follow along with the tasks and fill in the blanks of the given code to complete beginner level.

## Tasks:

Follow the tasks with *"starter.py"* and fill in the missing code for each section.

**T1: Reading images.**

- Change the dataset_path to point to the unzipped Dataset_1/images folder in your computer.
- The given loop will go through all the files in the folder, variable i gives each file name.
- Complete the code to read the images and append them to list X
- The labels for each image have been already appended to list y for you

At the end of T1, you should have X, y with 5998 entries on each.

**T2: Pre-processing images.**

- Given loop will go through all images in X and resize them to 48x48 pixels.
- Complete the code to convert the images to grayscale. (Hint: use the cvtColor function in opencv)
- Complete the code to append the pre-processed images to X_processed list.

At the end of T2, you should have X_processed with 5998 entires of resized and grayscale images.

**T3: Calculating Features and Splitting train/test sets.**

- Install skimage using anaconda. (you can follow the same instructions given for installing sklearn with the package name "scikit-image")
- The given code will use skimage and extract hog features for you.
- Write code to split X_features and y into training and testing sets. Make use of the *"sklearn.model_selection.train_test_split"* to do this. Use a 80-20 split and make sure to shuffle the samples.

At the end of T3, you should have x_train, x_test, y_train and y_test. Training sets should have 4798 samples and the test sets should have 1200 samples.

**T4: Training and testing the classifier.**

- Use the sklearn SVM package to train a classifier using x_train and y_train.
- Use the x_test and y_test to evaluate the classifier and print the accuracy value.

# Expert Level:

We will build upon the beginner level code to try out different techniques and improve our model. The same dataset will be used here.

Complete the following tasks,

## Tasks:

**T1: Different pre-processing techniques**

What are other pre-processing steps you can use?

Examples: Perform gaussian blur to reduce noise; or convert to a different color space.

Try a few other pre-processing techniques and evaluate how they affect accuracy.

**T2: Different features**

What are other feature extraction methods you can use?

Explore some other feature extraction methods given in skimage: https://scikit-image.org/docs/stable/api/skimage.feature.html

Try few other feature extraction methods and evaluate how they affect accuracy. You can also try software packages other than skimage.

**T3: Different Classification Models**

What are other classification models you can use?

Try other classifiers including but not limited to: RandomForrest, kNN and Decision Tree.

For each classifier, change parameters and evaluate how the parameters affect accuracy.

## Bonus Level:

This level is for you to apply what you learned to a more challenging dataset from scratch. The dataset is, German Traffic Sign Recognition Benchmark (GTSRB) (German Traffic Sign Benchmarks (rub.de)). This is available in the *"Dataset_2.zip"* file.

This dataset is already split into training and testing sets for you.

**Task:**

Write a python program to load the images from this dataset, into X, y. Then do suitable pre-processing, feature extraction and model training to develop a Traffic Sign Recognition system.

Report on the methods used and the results obtained by your Traffic Sign Recognition system.

## Important Caveats:

1. You must complete Beginner Level. If you don't, you will get 0 for the project. The other levels are optional.
2. For the Expert Level, you may only use the classifiers provided in scikit-learn.
3. If you use any pre-trained model from the internet (eg ResNet, VGG), or any deep learning network, you may only do this for the Bonus Level.