Title: Traffic Sign Recognition

- Liam Brown 101076657
- Nick Truong 101070517

Challenge:

This task is challenging as while traffic signs do have their differences, they also have a number of similarities to each other (shape, colour), so it is necessary to develop accurate geometry and edge detection as well as image processing so that all traffic signs are distinguishable by our model. In addition, this ought to be fast as well.

This means that the image processing ought to differentiate as much important detail as possible from what's unimportant. The better this is done, the less work has to be done when training the model.

We hope to learn how to use opencv to take valuable information from images using image processing and machine learning.

Summary:

An application that detects traffic signs and recognizes the type of signs that it detects. Potential approaches could include:

- Matching test set images to a given stock image of a sign
- Classifying similar signs into groups

Goals / Deliverables:

We hope to have a model that can classify images quickly by preprocessing at least 2 features ("Cleaning") for accuracy and speed, then identify the "Sign", in the image, then use machine learning to classify them. We're hoping to, at minimum, achieve %80 accuracy.

To evaluate success, we want a high accuracy in detecting and recognizing signs, as well as a fast speed.

To show evidence of success, we can record videos of the program processing videos/live feeds of traffic signs, thus capturing its speed. We can test for its accuracy by testing the program over a large number of samples and recording the results.

We feel that it is very realistic to achieve what has been outlined above in the allotted time.

If things go *very well*, the hope is to make it fast and accurate enough to use in a realtime setting (via a camera feed). This would be if we are ahead of schedule.

Schedule:

Every 2 weeks, we will get some part of our project done, allocating the tuesday and thursday of both weeks for production.

We will use an Agile software development approach, meaning the work will be progressed and assigned by availability as it is required.

- Goal 1:
    - Set up a rudimentary image processing system using OpenCV

- Goal 2:
    - Set up an edge detection system
    - Finalize an approach to sorting the signs, potential ways:
        - Sort by type
        - Find signs with strongest association with a given png of a sign

- Goal 3:
    - Set up a geometry recognition system

- Goal 4:
    - Add image “cleaning” functionality (choose one)

- Goal 5:
    - Add image “cleaning” functionality (choose one)

- Goal 6:
    - Use machine learning to associate pictures with signs

- Goal 7 (extra):
    - Figure out how to hook up a camera feed to our program

Background:

We will be using opencv in c++ for computer vision problems. This will help optimize speed of preprocessing the images.

We will be using numpy in python for machine learning problems. This will help optimize simplicity.

The application itself will simply be written for a Windows machine.

For a dataset, we will use the German Traffic Sign Recognition Benchmark (GTSRB) dataset from the INI Benchmark Website: https://benchmark.ini.rub.de/