**Synopsis**:

* Human factor remains the most common cause of road mortality.Traffic signs are an essential part of our day to day lives. They contain critical info that ensures the safety to all the people around us.
* Without traffic signs,all the drivers would be clueless about what might be ahead to them and roads can become a mess.
* In order to avoid this mess we have come up with the idea of “**Traffic Sign Recognition**”
* Every country has some standards set for the design of different traffic signs like U-turn, Left-turn, Right-turn, No-entry, etc. Traffic-sign recognition (TSR) is a technology by which a vehicle is able to recognize the traffic signs put on the road e.g. "speed limit" or "children" or "turn ahead".
* The earlier Computer Vision techniques required lots of hard work in data processing and it took a lot of time to manually extract the features of the image.
* Now, deep learning techniques have come to the rescue and we will be using convolution neural network model .
* The principle of driving assistance systems aiming at road signs recognition is to detect signs, interpret their meaning, then transmit the information to the driver (by a projection on a windshield, a screen or a smartphone).

Implementation:

1. Load the dataset
2. Split it into test, train such that test part contains a ratio of .2% of data.
3. Applying categorical function to get fixed values
4. Applying sequential model
5. First layer is a 2d convolution layer which has 32 filters and relu as activation function
6. Next layer is a maxpool layer with pool size of (2,2) and a dropout of .25% is added
7. Next two layer are 2d convolution layer which has 64 filters and relu as activation function
8. Next layer is a maxpool layer with pool size of (2,2) and a dropout of .25% is added
9. Finally 2 dense layers are applied one with relu activation and other with softmax.
10. Then the model is made to fit with 25 epochs.
11. At this stage we will ge a loss of 0.5 and accuracy of 0.9250 on training data and validation loss of 0.307 and validation accuracy of .9534.

**GUI Construction**:

1. Importing the model
2. Declare a dictionary to store the names of all the symbols.
3. Use cv2.VideoCapture for capturing live images.
4. We are using infinite while loop to capture video elements.
5. After capturing  the image some preprocessing is done and then the sign is predicted.
6. :**Technology used:**

* To implement this project we will be using-
  + Tensorflow and Keras which is a popular deep learning framework for python and some additional library, Scikit-learn, Matplotlib, Numpy, PIL for image processing, Pandas, CV2.
  + Pycharm community version and Web camera
  + Language used - python3.8.6
  + We have used python for both back end and front end development