**THEORY OF COMPUTATION AND COMPILER DESIGN**

**SLOT**-G2+TG2

**GUIDED BY: -** Prof. Shalini L

**PROJECT NAME: -**

**Traffic sign recognition on Indian roads using neural networks**

**GROUP MEMBERS: -**

1. Pranav Pramod Menon – 16BCE0090

2. Nishant Garg – 16BCE0088

3. Pratham Sharma – 16BCE0653

**Abstract**

Traffic signs are characterized by a wide variability in their visual appearance in real-world environments. For example, changes of illumination, varying weather conditions and partial occlusions impact the perception of road signs. In practice, a large number of different sign classes needs to be recognized with very high accuracy. Traffic signs have been designed to be easily readable for humans, who perform very well at this task. For computer systems, however, classifying traffic signs still seems to pose a challenging pattern recognition problem. Both image processing and machine learning algorithms are continuously refined to improve on this task. But little systematic comparison of such systems exists. For assessing the performance of state-of-the-art machine learning algorithms, we present a publicly available traffic sign dataset with more than 50,000 images of German road signs in 43 classes. The data was considered in the second stage of the German Traffic Sign Recognition Benchmark held at IJCNN 2011. The results of this competition are reported and the best-performing algorithms are briefly described. Convolutional neural networks (CNNs) showed particularly high classification accuracies in the competition. We measured the performance of human subjects on the same data—and the CNNs outperformed the human test persons.

**Introduction**

Traffic sign recognition is a multi-category classification problem with unbalanced class frequencies. It is a challenging realworld computer vision problem of high practical relevance, which has been a research topic for several decades. Many studies have been published on this subject and multiple systems, which often restrict themselves to a subset of relevant signs, are already commercially available in new high- and mid-range vehicles. Nevertheless, there has been little systematic unbiased comparison of approaches and comprehensive benchmark datasets are not publicly available. Road signs are designed to be easily detected and recognized by human drivers. They follow clear design principles using color, shape, icons and text. These allow for a wide range of variations between classes. Signs with the same general meaning, such as the various speed limits, have a common general appearance, leading to subsets of traffic signs that are very similar to each other. Illumination changes, partial occlusions, rotations, and weather conditions further increase the range of variations in visual appearance a classifier has to cope with. Humans are capable of recognizing the large variety of existing road signs in most situations with near-perfect accuracy. This does not only apply to real-world driving, where rich context information and multiple views of a single traffic sign are available, but also to the recognition from individual, clipped images.

**LITERATURE REVIEW**

**1.** Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition

- J.Stallkamp, M.Schlipsing, J.Salmen, C.Igel

2. Learning to Detect Traffic Signs: Comparative Evaluation of Synthetic and Real-world Datasets

- Andreas Møgelmose, Mohan M. Trivedi. Thomas B. Moeslund

METHODOLOGY

1. Load the positive samples data from a .mat file. The file names and bounding boxes are contained in an array of structures named 'data'.

2. Specify the folder for negative images.

3. Train a cascade object detector called 'stopSignDetector.xml' using HOG features.

4. Use the newly trained classifier to detect a traffic sign.

CODE

load('stopSigns.mat');

%%

% Add the images location to the MATLAB path.

imDir = fullfile(matlabroot,'toolbox','vision','visiondata','stopSignImages');

imDir1= fullfile(matlabroot,'toolbox','vision','visiondata','keeprightSignImages');

addpath(imDir,imDir1);

%%

% Specify the folder for negative images.

negativeFolder = fullfile(matlabroot,'toolbox','vision','visiondata','nonStopSigns');

neg1=fullfile(matlabroot,'toolbox','vision','visiondata','nonkeeprightSignImages');

%%

% Train a cascade object detector called 'stopSignDetector.xml' using HOG features. The following command may take several minutes to run:

trainCascadeObjectDetector('stopSignDetector.xml',data,negativeFolder,'FalseAlarmRate',0.2,'NumCascadeStages',5);

trainCascadeObjectDetector('keeprightsigndetector.xml',data,neg1,'FalseAlarmRate',0.1,'NumCascadeStages',5);

%%

% Use the newly trained classifier to detect a stop sign in an image.

detector = vision.CascadeObjectDetector('stopSignDetector.xml');

%%

% Read the test image.

img = imread('donotenter.jpg');

%%

% Detect a stop sign.

bbox = step(detector,img);

bbox1=step(detector1,img);

%%

% Insert bounding boxes and return marked image.

detectedImg = insertObjectAnnotation(img,'rectangle',bbox,'stop sign');

detectedImg1 = insertObjectAnnotation(img,'rectangle',bbox1,'Keep right');

% Display the detected stop sign.

figure;

imshow(detectedImg);

figure;

imshow(detectedImg1);

% Remove the image directory from the path.

RESULT