Bannari Amman Institute of Technology

Sathyamangalam 638 401

Stay Ahead

Final Year Students Project Handbook

Regulations 2018

TRAFFIC SIGN CLASSIFICATION USING DEEP LEARNING

PROJECT REPORT

Submitted by

AISHVARYA R(182IT103)
BAVADHARANI M P(182IT119)

In partial fulfilment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

BANNARI AMMAN INSTITUTE OF TECHNOLOGY
(An Autonomous Institution Affiliated to Anna University, Chennai)
SATHYAMANGALAM-638401

ANNA UNIVERSITY: CHENNAI 600 025

JANUARY 2022

BONAFIDE CERTIFICATE

Certified that this project report **"TRAFFIC SIGN CLASSIFICATION USING DEEP LEARNING"** is the bonafide work of "**BAVADHARANI M P(182IT119), AISHVARYA R(182IT103)**" who carried out the project work under my supervision**.**

SIGNATURE	**SIGNATURE**
Dr. DANIEL MADAN RAJA	Mr. ARUN KUMAR R
HEAD OF THE DEPARTMENT	**ASSISTANT PROFESSOR**
Department of Information Technology	Department of Information Technology
Bannari Amman Institute of Technology	Bannari Amman Institute of Technology
Sathyamangalam	Sathyamangalam
Erode – 638401.	Erode – 638401.

Submitted for Project Viva Voce examination held on MAY 31, 2022

Internal Examiner External Examiner

DECLARATION

We affirm that the project work titled **“TRAFFIC SIGN CLASSFICATION USING DEEP LEARNING”** being submitted in partial fulfilment for the award of the degree of **Bachelor of Technology** in **Information Technology** is the record of original work done by us under the guidance of **Mr Arun Kumar R**, Assistant professor, Department of Information Technology. It has not formed a part of any other project work(s) submitted for the award of any degree or diploma, either in this or any other University.

(Signature of candidate)	(Signature of candidate)
Aishvarya.R	Bavadharani.M.P
182IT103	182IT119

I certify that the declaration made above by the candidates is true.

(Signature of the Guide)

Mr ARUN KUMAR R

ACKNOWLEDGEMENT

We would like to enunciate heartfelt thanks to our esteemed Chairman **Dr.S.V.Balasubramaniam**, and the respected Director **Dr.M.P.Vijaykumar**, for providing excellent facilities and support during the course of study in this institute.

We are grateful to **Dr. S Daniel Madan Raja**, **Associate Professor and Head**, Department for his valuable suggestions to carry out the project work successfully.

We wish to express our sincere thanks to **Ms. Ponnila P**, **Assistant professor**, for her constructive ideas, inspirations, encouragement and much needed technical support extended to complete our project work.

We wish to express our sincere thanks to **Mr. Arun Kumar R**, **Assistant professor**, Department of Information Technology, for his constructive ideas, inspirations, encouragement, excellent guidance and much needed technical support extended to complete our project work.

We would like to thank our friends, faculty and non -teaching staff who have directly and indirectly contributed to the success of this project.

AISHVARYA R(182IT103),
BAVADHARANI M P(182IT119).

ABSTRACT

Traffic sign is the important part of the traffic system. It is mandatory to follow traffic sign and obey traffic rules to ensure surety. Road safety is important for both pedestrian and drivers. There are many factors affecting road safety like road conditions, traffic signs, weather conditions, conjected roads and the most important thing is people who are not following the traffic signs. The traffic sign recognition system plays a vital function and it provides timely road information to the driver, and may ensure the safety of life on the road. Here, selected signs was taken and worked for traffic sign detection, we used a convolutional neural network (CNN) individually to detect and recognize the traffic signs. Driver will recognize the traffic sign and get alert so unwanted accidents might be avoided.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1 INTRODUCTION ON ROAD SAFTEY SIGNS

Traffic Sign is a crucial piece of the trafficking framework. It is basic to follow the traffic signs and comply with traffic rules to guarantee security. Street wellbeing is a significant worry for the two people on foot and drivers. Plenty of factors are there which influence the well-being of the street like street conditions, traffic signs, weather patterns, blocked streets, and so on. Among them, the most crucial one is traffic signs. At times drivers misread these traffic signs which is one of the contributing variables to mishaps. With the blast of the auto business, they are zeroing in on executing independent vehicles. It is expected to be more secure and more effective. Traffic sign grouping is a significant part of independent vehicles. It ought to be done effectively to control street mishaps and to encourage the believability of independent vehicles. In any case, this could imperil the trafficking framework. Characterizing traffic signs is as yet a difficult example acknowledgment issue for PC frameworks. Ordinary Methods like AI Algorithms have been utilized in rush hour gridlock sign characterization. AI is a field that had outgrown the field of man-made consciousness, is of most extreme significance since it empowers the machines to acquire human-like insight without express programming. Because of the misfortune in handmade elements, a heap of profound learning calculations have been advanced to naturally become familiar with the elements through the various secret layers in a profound brain organization. Profound learning is the investigation of fake brain organizations and related AI calculations that contain multiple secret layers. Profound learning permits computational models that are made out of different handling layers to learn portrayals of information with different degrees of deliberation Deep learning has reliably worked in its capacity to give precise acknowledgment and forecast. So Deep learning outperforms Machine Learning calculations. This paper proposes a CNN model of profound nature CNN Model by adding four thick layers or completely associated layers. Since a completely associated layer learns highlights from every one of the mixes of the elements of the past layers, it advances profoundly.

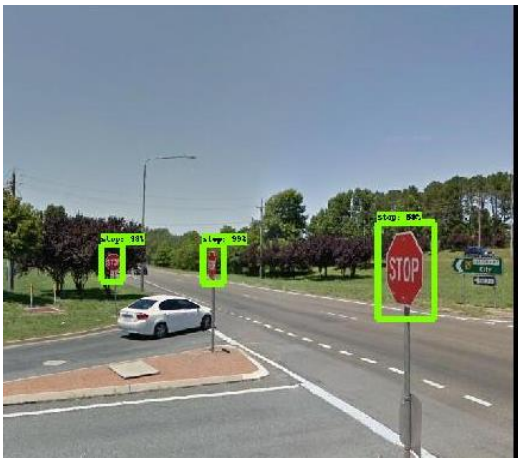


Figure 1.1.1 Diagram of Traffic sign classification.

1.2 PROJECT OBJECTIVE:

- The ultimate aim of the project is to alert the driver by the automatically recognizing the traffic signs which is present along the road using deep learning. This project is to detect the traffic sign ang help the driver to drive carefully and also to avoid accidents
- As per available literature, very little body of research is attempted to detect the traffic sign automatically.

1.3 PROJECT SCOPE:

- The Scope of this task is to make things simple for the individual who is driving the vehicle, which offers the data about the traffic hints he will cross.
- The traffic signs improve traffic security by illuminating the driver regarding speed cut-off points or potential perils, for example, cold streets, approaching street works or passer-by intersections.

Chapter 2

LITERATURE SURVEY

V.abdul Rahiman et al[4] proposed about The detection and identification of traffic signs is handled by a vision-based vehicle guiding system. The system collects data from the road ahead and supports the driver in making quick decisions, making driving safer and easier.

Arturo de la Escalera et al [5] recommend about traffic signs give drivers with extremely useful road information in order to make driving safer and easier. Traffic signs, we believe, should serve the same purpose for autonomous cars. They are meant to be quickly identified by human drivers due to their distinct colour and form from the surrounding surroundings.

Kedkarn Chaiyakhan et al [1] recommended about A traffic sign categorization system is a feature of a driving assistance system that warns and advises the driver on the meaning of traffic signs. We proposed the idea of automatically classifying each type of data in this study of directional signs A publicly available dataset was used to test the proposed approach.The main contribution offered in this research is traffic sign categorization using support vector machines and image segmentation. For classification, the image segmentation algorithm region of interest (ROI) with normal direction feature utilising SVM linear kernel function is effective.

Min Shi,Haifeng, et al [2] proposed about One of the key tasks of many traffic sign identification systems is to classify the shapes of traffic signs. They used support vector machines to create a shape-based classification model. The major goal of this project was to recognise seven different types of traffic sign shapes and five different types of speed restriction signs. The data was represented to the

SVM for training and testing using two types of features: binary image and Zernike moments.

Min Shi et al [3] suggested about The Intelligent Transportation System (ITS) is a system that uses sophisticated technology to create a safe, efficient, and integrated transportation environment. The detection and recognition of road signs is a key aspect of ITS, which provides ways to collect real-time traffic data for processing at a central location.This project will create a road sign recognition model based on AI and image analysis technologies that uses the Support Vector Machines machine learning method to recognise road signs.

Chapter 3

PROBLEM STATEMENT

A good answer to this problem would be to create machines that are environmentally conscious. As a result, safe vehicle driving is becoming a hot concern in a variety of areas, from tiny ventures to huge automobile manufacturing. However, this subject presents several concerns and issues. It is necessary to determine the width of the road's margins, recognise road signs, traffic signals, pedestrians, and other items that contribute to safe driving. There are several approaches to solve these problems.

Chapter 4

SYSTEM ANALYSIS

4.1 EXISTING SYSTEM

The past exploration was a picture classifier in view of an AI calculation (KNN). AI (ML) is the specialized investigation of calculations and measurable models that PC frameworks use to do an undertaking effectively without using unequivocal directions, rather depending on models and derivation.

Disadvantages:

- Less accuracy
- It’s not work well with image data

4.2 PROPOSED SYSTEM

This research implemented a CNN model to enhance the accuracy of traffic sign classification. Initially, Data pre-processing was done, which consists of Gray Scale conversion, Histogram Equalization, and Normalization on the GTSRB dataset. Data pre-processing is followed by Data augmentation. Data augmentation increased the number of training data to prevent overfitting. After that, the TS CNN model was implemented and trained with the GTSRB dataset.

Data Pre-processing:

- Gray Scale Conversion
- Histogram Equalization
- Normalization

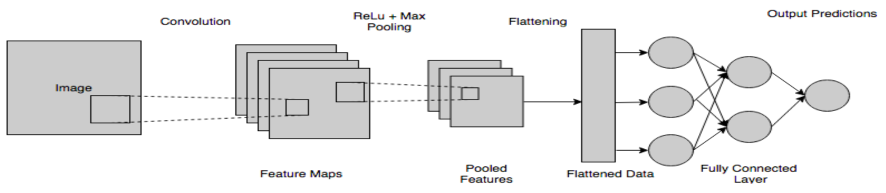


Figure 4.1.1 CNN algorithm layers

Chapter 5

SYSTEM REQUIREMENTS

5.1 HARDWARE REQUIREMENTS

- Laptop/System
- Minimum 4GB RAM.
- i5/i7 Processors
- Stable Internet connection
- 1GB free of Hard disk space / 500mb free of SSD

5.2 SOFTWARE REQUIREMENTS

- Operating System: Windows (Any version)
- Python Idle version 3.7
- Python packages
- Web browser: Microsoft Edge/Firefox/Google Chrome.

Chapter 6
METHODOLOGY

6.1 STEPS

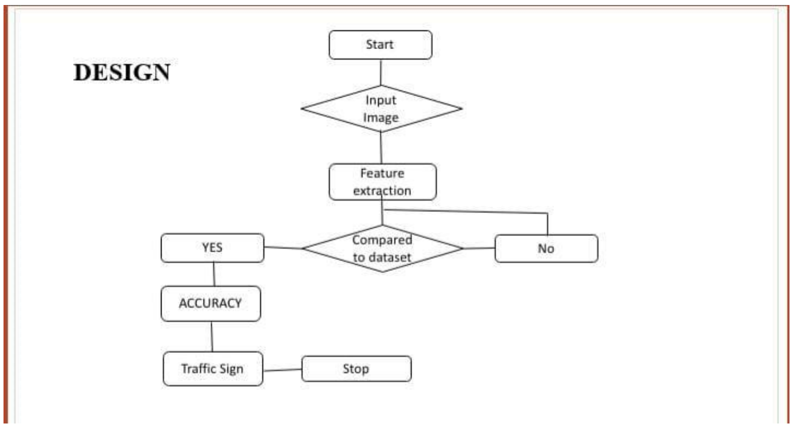


Figure 6.1.1 Design for Traffic sign classification

- Gather the information needed for training. After identifying some of the traffic signs, and gathered the datasets of traffic signs.
- After collecting the data pre-processing is done, it is nothing but image resizing. Removing the missing or irrelevant information
- After that, data is trained in order to anticipate the outcome. They have trained the data that was acquired after using the pre-processing approach.
- After training is completed, all the trained data will be stored in a file called pickle file. once this file is created, A file can be used for testing purpose how many times they want and it is also need not to be trained again
- Then it will predict what kind of traffic sign the input image is and also it will predict the accuracy, that is how much percentage the input image matches to the dataset image and then the accuracy will be calculated.

6.2MODULES

6.2.1 DATA COLLECTION

In data collection process we collect a different types of traffic sign images for Kaggle training purpose.

6.2.2 DATA PRE-PROCESSING

The adjustments we do to our data before passing it to the algorithm are referred to as pre-processing. Data The raw data is converted into an understandable data set using a pre-processing approach. In other words, anytime data is received from numerous sources, it is collected in a raw format that makes analysis impossible.

6.2.3 DATA TRAIN

The data used to train an algorithm or deep learning model to anticipate the outcome you want it to predict is referred to as training data. The performance of the algorithm you're applying to train the machine is measured using test data.

6.2.4 MODEL CREATE

A neural network is a simplified model of how the human brain processes information. It models a large number of interconnected processing units that resemble abstract representations of neurons. The processing elements are organised into layers.

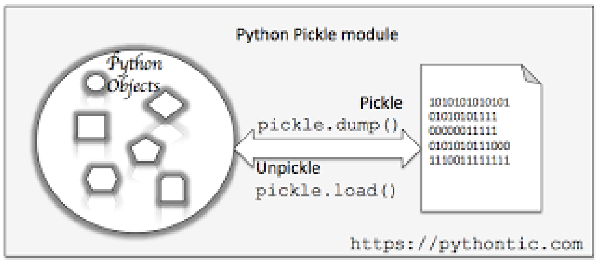


Figure 6.2.1 Creating the model

6.2.5 MODEL PREDICTION

Predictive analysis is a mathematical procedure that analyses patterns in a collection of input data to predict future events or outcomes. It's an important part of predictive analytics, a sort of data analytics that forecasts activity, behaviour, and trends using based on prior data.

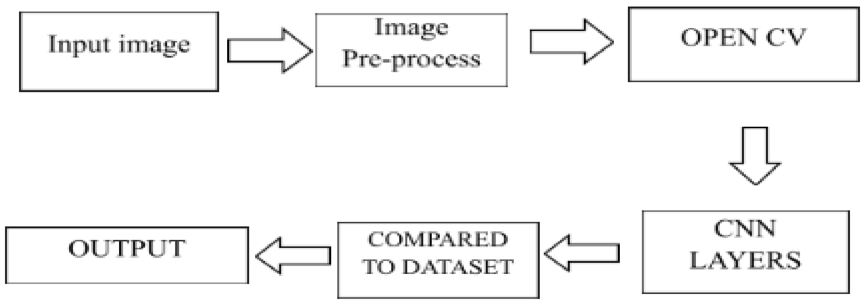


Figure 6.2.2 Block Diagram

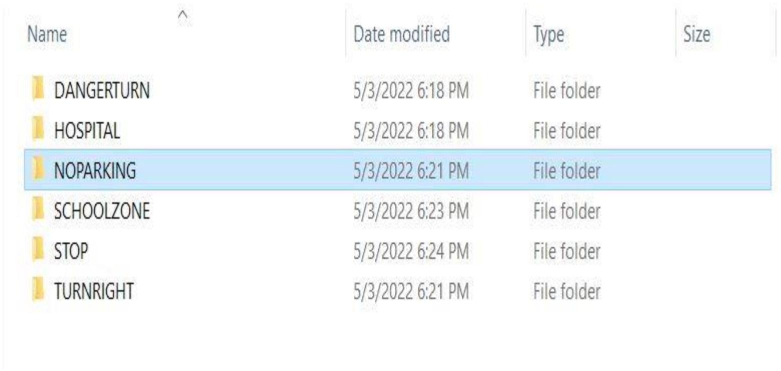
Chapter 7

MODEL & TRAINING

7.1 DATA COLLECTION

Collecting Datasets of traffic sign symbols from a particular platform like Kaggle. (https://www.kaggle.com/datasets/shanmukh05/traffic-sign-cropped).

Figure 7.1.1 Traffic sign dataset



Name	Date modified	Type	Size
DANGERTURN	5/3/2022 6:18 PM	File folder	
HOSPITAL	5/3/2022 6:18 PM	File folder	
NOPARKING	5/3/2022 6:21 PM	File folder	
SCHOOLZONE	5/3/2022 6:23 PM	File folder	
STOP	5/3/2022 6:24 PM	File folder	
TURNRIGHT	5/3/2022 6:21 PM	File folder	

Figure 7.1.2 Important traffic signs

Chapter 8
INPLEMENTATION

8.1 LIBRAIES USED

- Open CV
- TensorFlow
- Keras
- NumPy

8.1.1 OPEN CV

OpenCV is an open-source computer vision, machine learning, and image processing toolkit. Python, C++, Java, and more programming languages are supported by OpenCV. It can analyse photos and movies to recognise items, people, and even human handwriting.

In the categorization of brain strokes, an open CV is employed to interpret the images. The imread syntax is used in the image reading process in CV2.

8.1.2 TENSORFLOW

TensorFlow is an open-source machine learning application development framework. It's a symbolic math toolkit that performs many tasks including deep neural network training and inference using dataflow and differentiable programming. Developers can use multiple tools, frameworks, and community resources to construct machine learning applications. Google TensorFlow is now the most well-known deep learning library on the planet. Machine learning is used by Google in all of its products to enhance search, translation, picture captioning, and recommendations.

8.1.3 KERAS

Keras is a high-level neural network API that may be used with Tensor Flow, Theano, and CNTK. It has a high-level, user-friendly, modular, and extendable API that allows for quick experimentation. Keras may be used on both the CPU and the GPU. Francois Chollet created and maintains Keras, which is part of the Tensor Flow core and is Tensor Flow's preferred high-level API.

Keras includes the two most popular Keras models (Sequential and Functional), as well as the core layers and certain Pre-Processing features.

8.1.4 NUMPY

NumPy, or Numerical Python, is a library that contains multidimensional array objects and a collection of functions for manipulating them. NumPy allows you to conduct mathematical and logical operations on arrays.

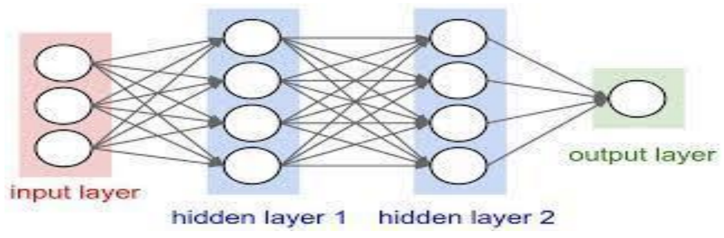
8.2 ALGORTHIUM USED

CNN: The input picture is transformed using a convolution layer in order to extract features from it. The picture is convolved with a kernel in this transformation. A kernel is a tiny matrix that is smaller in height and width than the picture to be convolved. A convolution matrix or convolution mask is another name for it.

It requires input data, a filter, and a feature map, among other things.

Assume the input is a colour picture composed of a 3D matrix of pixels. This implies the input will have three dimensions, which correspond to RGB in a picture.

A feature detector, also known as a kernel or a filter, will traverse over the image's receptive fields, checking for the presence of the feature. Convolution is the term for this procedure.



8.2.1 Convolution neural network algorithm

8.3 LAYERS IN CNN

- Pooling layer
- Flattern layer
- Fully connected layer

8.3.1 POOLING LAYER

Down sampling, also known as pooling layers, is a dimensionality reduction technique that reduces the number of factors in the input. The pooling process sweeps a filter across the whole input, similar to the convolutional layer, however this filter does not contain any weights. Instead, the kernel uses an aggregation function to populate the output array from the values in the receptive field.

8.3.2 FLATTERN LAYER

The flatten function reduces multi-dimensional input tensors to a single dimension, allowing you to model your input layer and create your neural network model, then effectively transfer those inputs to each and every neuron in the model.

8.3.3 FULLY CONNECTED LAYER

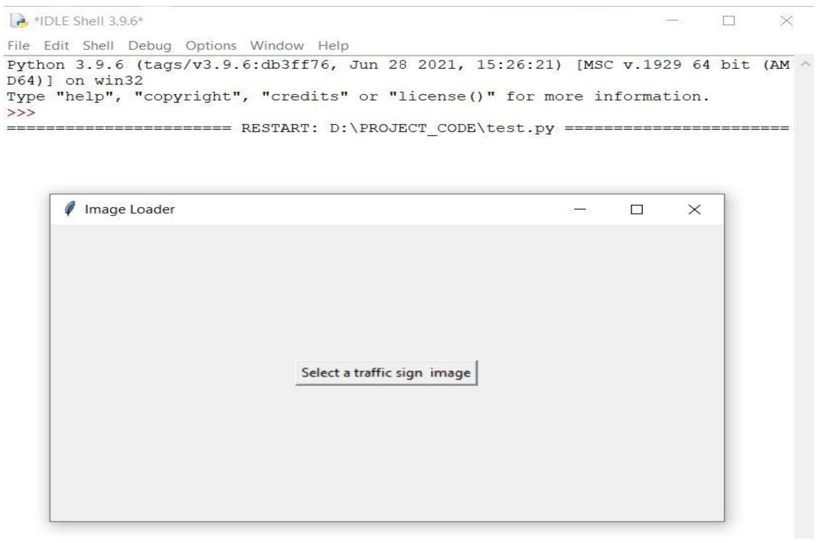
- The full-connected layer's name is self-explanatory. In partly linked layers, the pixel values of the input picture are not directly connected to the output layer, as previously stated.

- Each node in the output layer, on the other hand, links directly to a node in the preceding layer in the fully-connected layer.
- This layer performs classification tasks based on the characteristics retrieved by the preceding layers and their various filters.

Chapter 9

RESULTS

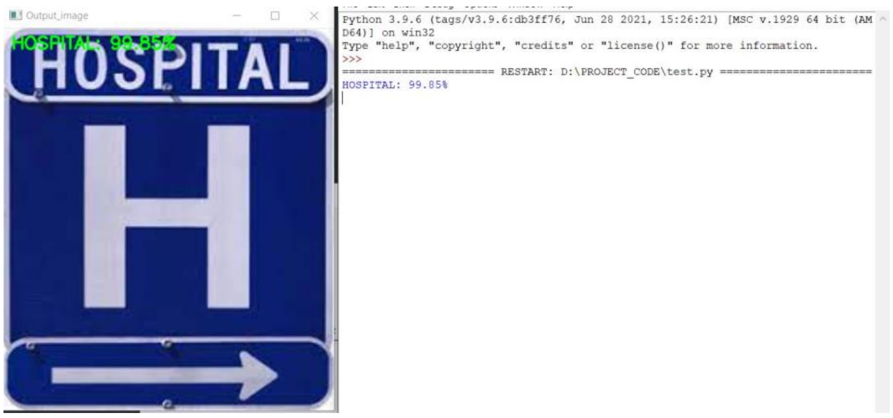
Traffic sign characterization is utilized to perceive traffic signs which is available along the road. Some guidelines are additionally shown for not many traffic signs. If the result picture is peril sign, then an alarm sound will be produced. If the result picture is school zone or medical clinic zone, then, at that point, a voice message like "go sluggish school is close" and "Kindly don't horn" will be produced. So, it will order the traffic signs and caution the driver so the gamble of mishaps can be diminished.



```
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
======================= RESTART: D:\PROJECT_CODE\test.py =======================
```

IDLE Shell 3.9.6
File Edit Shell Debug Options Window Help
Image Loader
Select a traffic sign image

Figure 9.1.1 Result Analysis



```
Python 3.9.6 (tags/v3.9.6:db3ff76, Jun 28 2021, 15:26:21) [MSC v.1929 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
======================= RESTART: D:\PROJECT_CODE\test.py =======================
HOSPITAL: 99.85%
```

Figure 9.1.2 Traffic Sign detection

Chapter 10
CONCLUSION

The GTSRB dataset was used to develop the Traffic Sign Classification using the TS CNN Model. Data pre-processing procedures were applied to the dataset to increase contrast and resolution. Following Data Pre-processing, Data Augmentation is used to enhance the number of pictures in the training samples in order to minimise overfitting. On the test data, our model had a significantly better accuracy of 98.44%. This method out performs current state-of-the-art algorithms. In the future, The aim is to improve the accuracy of the traffic sign classification by employing CNN variations.

Chapter 11
REFERENCES

1. S.Maldonado-Bascon, S. Lafuente-Arroyo, P. Gil-Jimenez, H. GomezMoreno, and F. Lopez-Ferreras, “Road-sign detection and recognition based on support vector machines,” IEEE Transactions on Intelligent Transportation Systems, vol. 8, no. 2, pp. 264–278.
2. G. Wang, G. Ren, Z. Wu, Y. Zhao, and L. Jiang, “A robust, coarse-tofine traffic sign detection method,” in Proceedings of IEEE International Joint Conference on Neural Networks.
3. C. G. Keller, C. Sprunk, C. Bahlmann, J. Giebel, and G. Baratoff, “Realtime recognition of u.s. speed signs,” in Proceedings of the Intelligent Vehicles Symposium.
4. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324.
5. Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980.

APPENDIX:

TRAIN.PY

```
from keras.models import Sequential
from keras.layers.convolutional import Conv2D
from keras.layers.convolutional import MaxPooling2D
from keras.layers.core import Activation
from keras.layers.core import Flatten
from keras.layers.core import Dense
from keras import backend as K
from keras.preprocessing.image import ImageDataGenerator
from keras.preprocessing.image import img_to_array
from sklearn.preprocessing import MultiLabelBinarizer
from sklearn.model_selection import train_test_split
from imutils import paths
import keras
import numpy as np
import random
import pickle
import cv2
import os

# Parameters:-
EPOCHS = 100
IMAGE_DIMS = (96, 96, 3)


# Input image:-
imagePaths = sorted(list(paths.list_images("dataset")))
random.seed(42)
random.shuffle(imagePaths)

# Create an list:-
data=[]
labels=[]

# loop over the input images
for imagePath in imagePaths:
        image = cv2.imread(imagePath)
        image = cv2.resize(image,(IMAGE_DIMS[1],IMAGE_DIMS[0]))

        image = img_to_array(image)
        data.append(image)

        l = label = imagePath.split(os.path.sep)[-2].split("_")
```

```
		labels.append(l)

# Convert into numpy both input & lables:-
data = np.array(data, dtype="float") / 255.0
labels = np.array(labels)

# binarizer implementation
mlb = MultiLabelBinarizer()
labels = mlb.fit_transform(labels)
print(labels.shape)
print(labels.ndim)

#  loop over each of the possible class labels and show them
for (i, label) in enumerate(mlb.classes_):
		print("{}. {}".format(i + 1, label))

# Split Training & Testing
(trainX, testX, trainY, testY) = train_test_split(data,
		labels, test_size=0.2, random_state=42)


# initialize the model
model = Sequential()

# Rows & Columns
imgRows=IMAGE_DIMS[0]
imgCols=IMAGE_DIMS[1]
numChannels=IMAGE_DIMS[2]
numClasses=6
inputShape = (imgRows, imgCols, numChannels)

activation="relu"
weightsPath=None

# define the first set of CONV => ACTIVATION => POOL layers
model.add(Conv2D(20, 5, padding="same",
	input_shape=inputShape))
model.add(Activation(activation))
model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))

# define the second set of CONV => ACTIVATION => POOL layers
model.add(Conv2D(50, 5, padding="same"))
model.add(Activation(activation))
model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))
```

```
# define the first FC => ACTIVATION layers
model.add(Flatten())
model.add(Dense(500))
model.add(Activation(activation))

# define the second FC layer
model.add(Dense(numClasses))

# lastly, define the soft-max classifier
model.add(Activation("softmax"))

# if a weights path is supplied (inicating that the model was
# pre-trained), then load the weights
if weightsPath is not None:
        model.load_weights(weightsPath)

#compile
model.compile(loss = keras.losses.categorical_crossentropy,optimizer = 'SGD',metrics
= ['accuracy'])

# fitting the model
hist = model.fit(x=trainX,y=trainY,epochs =EPOCHS ,batch_size =
128,validation_data =(testX,testY),verbose = 1)
history = model.fit(x=trainX,y=trainY,epochs =EPOCHS ,batch_size = 128)
print (history.history)
# evaluate the model
test_score = model.evaluate(testX,testY)
print("Test loss {:.5f},accuracy {:.3f}".format(test_score[0],test_score[1]*100))



print(accuracy)

print(loss)

# Save the model
model.save('TRAINING_EXPERIENCE.h5')

f = open("MLB.PICKLE", "wb")
f.write(pickle.dumps(mlb))
f.close()
print(trainX.shape)
print(testX.shape)

print(trainY.shape)
print(testY.shape)
```

TEST.PY

```
from keras.models import load_model
from keras import models
from keras.preprocessing.image import img_to_array
import cv2
import numpy as np
import pickle
import imutils
from tkinter import *
from PIL import Image

from tkinter import filedialog
# loading Python Imaging Library
from PIL import ImageTk, Image

root = Tk()
# Set Title as Image Loader
root.title("Image Loader")

# Set the resolution of window
root.geometry("550x300")


# Allow Window to be resizable
root.resizable(width = True, height = True)
##import button
def load():
    with open("output.img", "r") as f:
        output=f.read()
# Load Model:-
model = load_model("TRAINING_EXPERIENCE.h5")
mlb = pickle.loads(open("mlb.pickle", "rb").read())

# Read an Input image:-
def select_image():
   image1=filedialog.askopenfilename()
   image = cv2.imread(image1)

   output = imutils.resize(image,width=400)
   image = cv2.resize(image, (96, 96))
   image = image.astype("float") / 255.0
   image = img_to_array(image)
   image = np.expand_dims(image, axis=0)

   accuracy = model.predict(image)[0]
```

```
    idxs = np.argsort(accuracy)[::-1][:1]
    #print(idxs)

    for (i, j) in enumerate(idxs):
            label = "{}: {:.2f}%".format(mlb.classes_[j], accuracy[j] * 100)
            print(label)
            cv2.putText(output, label, (10, (i * 30) + 25),
                    cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 255, 0), 2)
    cv2.imshow('Output_image',output)
# create a button, then when pressed, will trigger a file chooser
# dialog and allow the user to select an input image; then add the
# button the GUI
btn = Button(root, text="Select a traffic sign  image", command=select_image)
btn.pack(side="bottom", expand="yes", padx="10", pady="10")
# kick off the GUI
root.mainloop()
```

REVIEWER COMMENTS AND QUERIES FROM SECOND REVIEW

- To improve in design part.

REVIEWER COMMENTS AND QUERIES FROM SECOND REVIEW

- Where will you implement this project in real time

Individual Contribution of Student 1:

NAME: BAVADHARANI M P
ROLLNO: 182IT119

1) Researched on algorithms and chose a better algorithm for our project.
2) Collected the datasets of traffic signs
3) Learned on python fundamentals

Individual Contribution of Student 2:

NAME: AISHVARYA R
ROLLNO: 182IT103

1) Researched about the procedures of using a software to get desired results
2) Trained the collected datasets
3) Learned about python.

ORIGINALITY SCORE (TURNITIN REPORT)

CONFERENCE CERTIFICATE AND JOURNAL PUBLICATION

https://drive.google.com/file/d/1S11Lt9N9qGdHEPm_01kQXjlELeYZOABt/view?usp=sharing