

Project Phase-1 Report

on

TRAFFIC SIGN CLASSIFICATION

Submitted to

NMAM INSTITUTE OF TECHNOLOGY, NITTE

(An Autonomous Institution Under VTU, Belagavi)

In partial fulfilment of the requirements for the award of the

Degree of Bachelor of Engineering

in

Computer Science and Engineering

by

KEERTAN KINI B

4NM17CS085

KISHAN

4NM17CS090

MANJUNATHA PATKAR

4NM17CS100

MANUKASHYAP U V

4NM17CS101

Under the guidance of

Mr. RAMESHA SHETTIGAR

Asst. Professor

Dept. of CSE, NMAMIT, NITTE



NITTE
EDUCATION TRUST

N.M.A.M. INSTITUTE OF TECHNOLOGY

(An Autonomous Institution affiliated to Visvesvaraya Technological University, Belagavi)

Nitte – 574 110, Karnataka, India

CERTIFICATE

Certified that the project work entitled TRAFFIC SIGN CLASSIFICATION is a bonafide work carried out by Keertan Kini B(4NM17CS085), Kishan(4NM17CS090), Manjunatha Patkar(4NM17CS100), Manukashyap U V(4NM1CS101) in partial fulfilment for the award of Degree of Bachelor of Engineering in Computer Science and Engineering of the Visvesvaraya Technological University, Belagavi during the year 2020-21. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of Project Phase- 1 prescribed for the said Degree.

Name & Signature of Guide

Signature of HOD

Signature of the Principal

External Viva

Name of the Examiners

Signature with Date

1. _____
2. _____

ACKNOWLEDGEMENT

The satisfaction that accompanies the completion of any task would be incomplete without the mention of all the people, without whom this endeavour would have been a difficult one to achieve. Their constant blessings, encouragement, guidance and suggestions have been a constant source of inspiration

First and foremost, my gratitude to my project guide, **Mr. Ramesha Shettigar** for his constant guidance throughout the course of this project Phase-1 and for the valuable suggestions.

I also take this opportunity to express a deep sense of gratitude to the project coordinators for their valuable guidance and support.

I acknowledge the support and valuable inputs given by, **Dr. Uday Kumar Reddy** the Head of the Department, Computer Science and Engineering, NMAMIT, Nitte.

My sincere thanks to our beloved principal, **Dr. Niranjan N Chiplunkar** for permitting us to carry out this project at our college and providing us with all the needed facilities.

Finally, thanks to staff members of the Department of Computer Science and Engineering and our friends for their honest opinions and suggestions throughout the course of our project Phase-1.

Keertan Kini B (4NM17CS085)

Kishan(4NM17CS090)

Manjunatha Patkar (4NM17CS100)

Manukashyap U V (4NM17CS101)

ABSTRACT

With the advent of electric vehicles and drive by wire, the idea of adding autonomy to motor vehicles became mainstream. Even though traffic signs have been around for ages now, people tend to ignore most of them, of which the speed limit is the most prominent. If we can create a machine learning model that will detect and classify the traffic signs, the ECU can take over, under the scenario where the driver ignores the signs.

CONTENTS

CHAPTER NO.	CHAPTER NAME	PAGE NO.
1.	INTRODUCTION	6
	1.1 Computer Vision	6
	1.2 Convolution Neural Network	6
	1.3 Computer Vision in Automobiles	6
	1.4 Traffic Sign Classification	6
2.	LITERATURE SURVEY	7-11
	2.1 Paper - 1	7
	2.2 Paper - 2	8
	2.3 Paper - 3	9
	2.4 Paper - 4	10
	2.5 Paper - 5	11
3.	PROBLEM DEFINITION	12
	3.1 Problem Statement	12
	3.2 Definition	12
4.	SYSTEM REQUIREMENTS SPECIFICATION	13
	4.1 Model Creation	13
	4.2 Model Execution	13
	4.3 Python Libraries	13
5.	SYSTEM DESIGN	14
6.	IMPLEMENTATION	15-18
	6.1 Importing the images	15
	6.2 Creating test - train split	15
	6.3 Validate the test - train split	16
	6.4 Displaying the sample image from all classes	16
	6.5 Displaying the test data distribution chart	17
	6.6 Preprocessing the images	17
	6.7 Image augmentation	18
7.	RESULTS AND CONCLUSION	19
8.	FUTURE WORKS	20
	REFERENCES	21

INTRODUCTION

Computer Vision

Definition: "Computer vision is an interdisciplinary scientific field that deals with how computers can gain a high-level understanding from digital images or videos. From the perspective of engineering, it seeks to understand and automate tasks that the human visual system can do."

Computer vision enables the computer to partially interact with the real world and gain valuable data over time that will enable next level of automation in the day to day tasks leading to a safer and comfortable world for humans while also decreasing the global energy footprint.

Convolution Neural Network

Definition: "In deep learning, a convolutional neural network is a class of deep neural networks, most commonly applied to analyzing visual imagery. They are also known as shift invariant or space invariant artificial neural networks, based on their shared-weights architecture and translation invariance characteristics."

Neural networks try to mimic the behaviour of the human brain to learn from real-world data and to solve real-world problems in the process. Convolution neural networks are one of the more efficient ways of using machine learning on images since they are capable of differentiating and identifying the different characters in each image, thereby accurately predicting the image class.

Computer Vision in Automobiles

Lately, the use of computer vision in the field has skyrocketed. Computer vision provides unprecedented security and comfort for the automobile and its occupants. Some of these services are collision detection, traffic sign detection, driver awareness detection, side collision detection, traffic awareness etc. But these features require a huge amount of Research and Development (R&D) budget and also highly capable processing units to be installed on the vehicles. Thus these features are only available in luxury vehicles whose cost would be the entire lifetime earnings of a common man a century ago.

Traffic Sign Classification

Traffic sign classification is the process of automatically recognizing traffic signs along the road, including speed limit signs, yield signs, merge signs etc. Being able to automatically recognize the traffic sign enables us to build smarter and safer automobiles.

LITERATURE SURVEY

Paper - 1

Paper Name: Towards Reliable Traffic Sign Recognition

-Benjamin Höferlin

Intelligent Systems Group Universität Stuttgart Stuttgart, Germany

benjamin.höferlin@vis.uni-stuttgart.de

-Klaus Zimmermann

European Technology Center (EuTEC) Sony Deutschland GmbH

Stuttgart, Germany klaus.zimmermann@sony.de

Abstract:

- System architecture for the reliable detection of circular traffic signs.
- 80s first research into computer-aided traffic sign detection.
- How old research is using colour segmentation, colour thresholding, Bayesian classification of colour.
- Hough transform and its derivatives.

Proposed Method:

- Detection
 - Twofold detection stage
 - Shape-based detection
 - Content-based detection
- Refinement
 - Contracting Curve Density Algorithm
- Classification
 - Two Multi-Layered Perceptrons

Conclusion:

- The 30-minute long test yielded 96.4% correct detections.

Future Work:

- Detection of non-circular signs
- Better methods for classification [1].

Paper - 2

Paper Name: Detection And Recognition of Indian Traffic Signs

- Pritika Priya, Dhara Modha, Mansi Agrawal Department Of
Information Technology, Bharati Vidyapeeth College Of Engineering
For Women, Pune-43

Abstract:

- An automatic system which would detect, recognize and interpret the meaning of the traffic signs for the driver.
- Use of several image processing techniques to enhance the efficiency and speed of the system.
- Disburden the drivers and reduce road accidents for better and safe driving, hence implementing the concept of the intelligent vehicle.

Proposed Method:

- Detection
 - Image blurring algorithm
 - Colour filtering
 - Blob detection
- Classification
 - Based on shapes: circle, rectangle and triangle.
- Recognition
 - Use of pattern matching algorithms to compare extracted ROI with standard templates.
 - If a pattern is found, sound notification is given to the driver otherwise the image is discarded.

Conclusion:

- Describes the system that is strictly used to differentiate Indian Traffic Signs that is subdivided into three classes according to the shapes.

Future Work:

- Improve the robustness of the system.
- Make the system work for highly tilted signs [2].

Paper - 3

Paper Name: Deep Learning for Large-Scale Traffic-Sign Detection and Recognition

-Domen Tabernik and Danijel Skočaj Faculty Computer and Information Science, University of Ljubljana Večna pot 113,1000 Ljubljana {domen.tabernik,danijel.skocaj}@fri.uni-lj.si

Abstract:

- Use of Mask R-CNN for Traffic sign detection and recognition.
- Using deep learning method for detection of traffic signs with large intra-category appearance variation.
- This approach is used for the detection of 200 traffic-sign categories.

Proposed Method:

- Detection and Recognition using Mask R-CNN
 - Online hard-example mining (OEHM)
 - Distribution of selected training sample
 - Sample weighting
 - Adjusting region pass-through during detection
- Data augmentation technique
 - This technique is used to generate several instances of traffic-signs and hence provide diverse data for the deep learning model.

Conclusion:

- Describes the system that is strictly used to differentiate Indian Traffic Signs that is subdivided in three classes according to the shapes.

Future Work:

- Average precision of 97.5% achieved in correct detections with an error rate of just 2%-3%.
- Improving the system to achieve ideal performance [3].

Paper - 4

Paper Name: Traffic Sign Classification Using Deep Inception Based Convolutional Networks

-Mrinal Haloi, IIT Guwahati

mrinal.haloi11@gmail.com

Abstract:

- Use of spatial transformer layers and a modified version of inception module specifically designed for capturing local and global features together.
- Classify precisely intraclass samples even under deformations.
- This approach addresses the concern of exploding parameters and augmentations.

Proposed Method:

- Transformation invariant
 - Localisation network
 - Grid generator
 - Sampling unit
- Proposed Pipeline
 - A modified version of GoogLeNet Inception module is used for the classification task.
- GTSRB data set is used for training and testing.

Conclusion:

- Achieves the state-of-the-art performance of 99.81% on GTSRB dataset.

Future Work:

- Improving the system to achieve ideal performance [4].

Paper - 5

Paper Name: Indian Traffic Sign Detection and Classification Using Neural Networks

-Arun Nandewal, CSE Department, arunnandewal@gmail.com

-Abhishek Tripathi, IT Department, abhishek.tripathi2421@gmail.com

-Satyam Chandra, EEE Department, satyam9871@gmail.com

NITK Surathkal

Abstract:

- This paper presents an automatic Indian Road Traffic Sign Detection and Classification system based on Multiple Neural Networks.
- Validated on a standard data set of Indian Traffic Signs.
- The proposed methodology works with real-time images invariant to rotation, illumination and partially distorted and occluded images.

Proposed Method:

- The proposed system has 4 stages:
 - Image procurement and preprocessing
 - Colour segmentation
 - Blob Detection using Binarization and Otsu Thresholding.
 - Classification using Multiple Neural Networks to decide the type of sign.

Conclusion:

- When the NN is trained over a standard database, the recognition of ROI has high accuracy.

Future Work:

- Real-time implementation requires a more robust system which has reduced proceeding time [5].

CHAPTER - 3

PROBLEM DEFINITION

Problem Statement

Creation of an automatic, fast and light system that is capable of real-time detection, classification and interpretation of the traffic signs.

Definition

Traffic sign classification is one of the most important uses of computer vision in automobiles. It has several aspects that include differentiating the road and surroundings, identifying the sign poles, closing in on the traffic sign and classifying it. We plan to tackle the issue of classifying the traffic signs once we have the sign in the view of the camera since it is the basic block of a traffic sign classification system.

Our goal is to create a machine learning model that is capable of classifying most of the traffic signals and also enable real-time interpretation of those signals. We aim to create an end product that can see the traffic signal through the camera and can interpret it onto the display while also being light on its resources.[8]

We plan to use OpenCV for the image input part of the system and the CNN model for the classification and interpretation of the traffic signs.[6]

SYSTEM REQUIREMENTS SPECIFICATION

Model Creation

- Any one of the following
 - Google Collab Notebook with GPU acceleration enabled.
 - Kaggle Notebook with GPU acceleration and Network enabled.
 - PC or Laptop with
 - Intel i5 or Ryzen 5 1600 or higher.
 - Nvidia GTX or RTX or Radeon series graphics card.
 - 1GB of storage space.
 - 8GB or more RAM
 - Anaconda or PyCharm IDE installed

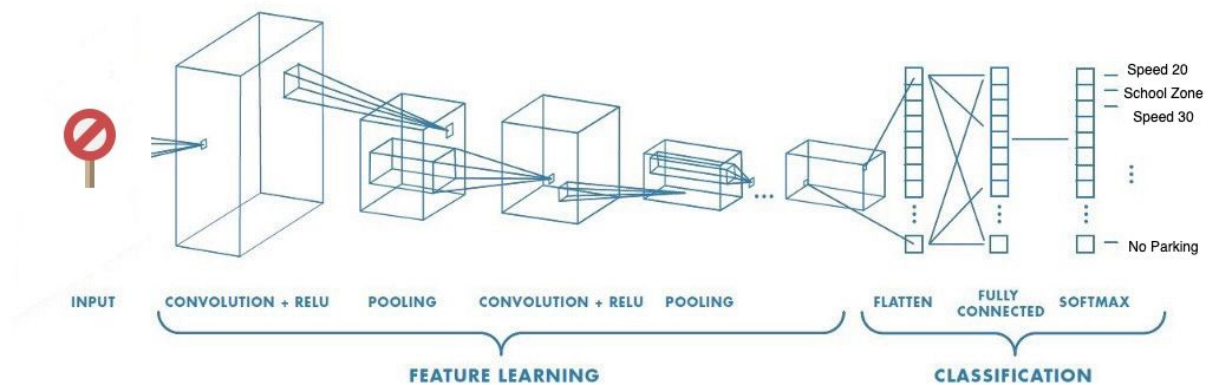
Model Execution

- Average PC or Laptop with WebCam.

Python Libraries

- Python libraries such as Numpy, Matplotlib, Tensorflow, Keras, Cv2, Sklearn etc are used.

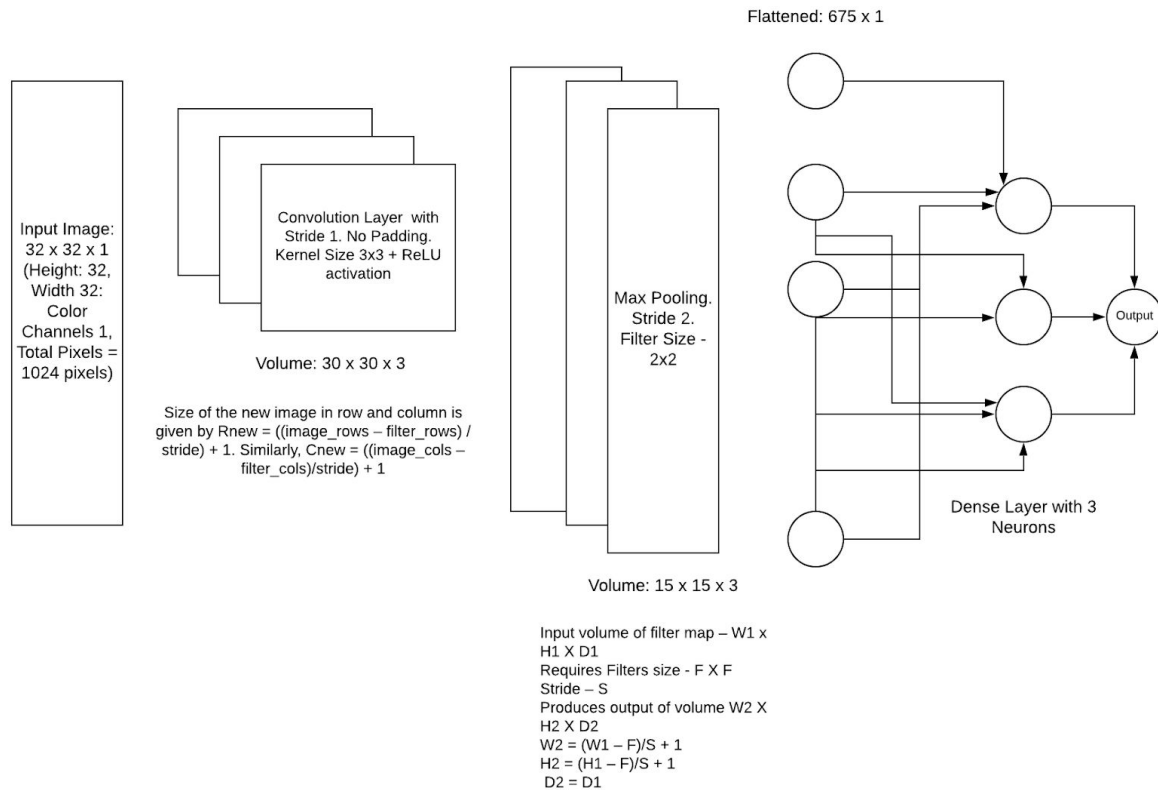
SYSTEM DESIGN



CNN is a machine learning algorithm that can take an input image, assign importance to various aspects or objects in the image, and be able to differentiate one from another.[7]

[6]CNN works by extracting the features from the images. Any CNN consists of the following:

1. The input layer which is a grayscale image.
2. The output layer is a binary or multi-class label.
3. Hidden layers consist of convolution layers, ReLU (Rectified Linear Unit) layers, the pooling layer, and a fully connected Neural Network.



CHAPTER - 6

IMPLEMENTATION

Importing the images

```
#Importing the images
count = 0 #stores the number of images in the dataset
images = [] #image list
classNo = [] #class list
imgCount = 0 #counter
myList = os.listdir(path) #This will list all the files and folders
inside the folder containing dataset
print("Total classes detected: ",len(myList))
noOfClasses = len(myList)
print("Importing the classes....")
for x in range(0, (len(myList) - 1)): #traverse the directory, find and
store the images and classes
    myPicList = os.listdir(path + "/" + str(count))
    for y in myPicList:
```

```

        curImg = cv2.imread(path + "/" + str(count) + "/" + y)
        images.append(curImg)
        classNo.append(count)
        imgCount += 1
    print(count, end = " ")
    count += 1
print("\nTotal number of Images detected: ",imgCount)
images = np.array(images)
classNo = np.array(classNo)

```

Creating test - train split

```

# Creating the split for train, test and validation from the data set
X_train, X_test, y_train, y_test = train_test_split(images, classNo,
test_size=testRatio)
X_train, X_validation, y_train, y_validation = train_test_split(X_train,
y_train, test_size=validationRatio)

# X_train = ARRAY OF IMAGES TO TRAIN
# y_train = CORRESPONDING CLASS ID

```

Validate the test - train split

```

#Validating the training and validation values created in the earlier
code
print("Data Shapes")
print("Train",end = "");print(X_train.shape,y_train.shape)
print("Validation",end =
 "");print(X_validation.shape,y_validation.shape)
print("Test",end = "");print(X_test.shape,y_test.shape)
#assert is a keyword that is used during debugging. It raises an
AssertionError if the condition is not met.
assert(X_train.shape[0]==y_train.shape[0]), "The number of images in not
equal to the number of labels in training set"
assert(X_validation.shape[0]==y_validation.shape[0]), "The number of
images in not equal to the number of labels in validation set"
assert(X_test.shape[0]==y_test.shape[0]), "The number of images in not
equal to the number of labels in test set"
assert(X_train.shape[1:]==(imageDimensions)), " The dimensions of the
Training images are wrong "
assert(X_validation.shape[1:]==(imageDimensions)), " The dimensions of

```



```
the Validation images are wrong "  
assert(X_test.shape[1:]==(imageDimensions))," The dimensions of the Test  
images are wrong"
```

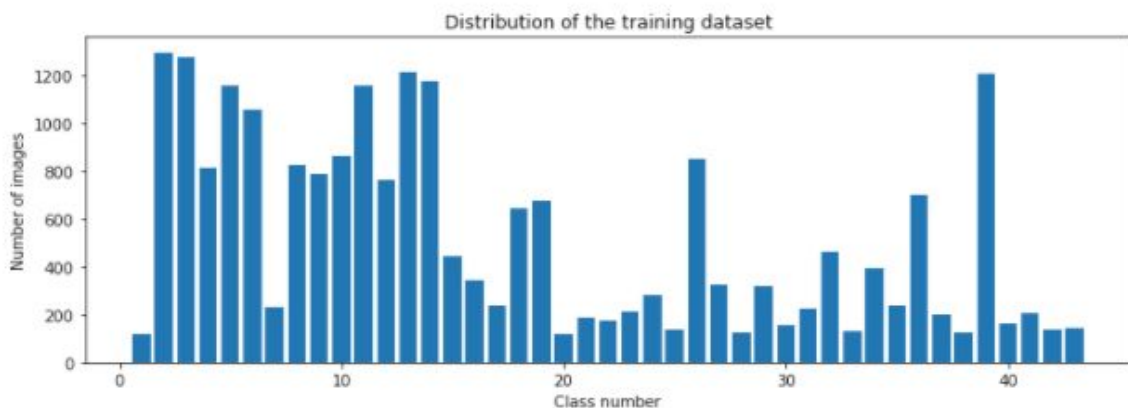
Displaying the sample images from all the classes

Output



Displaying the test data distribution chart

Output:



Preprocessing the images

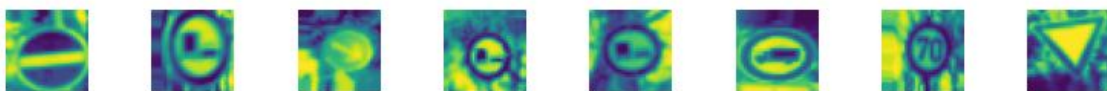
```
#Pre processing the images
def grayscale(img):
    img = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY) #Converting the color
space of the image
    return img
def equalize(img):
    img =cv2.equalizeHist(img) #Equalise the histogram of the images
(standard lighting)
    return img
def preprocessing(img):
    img = grayscale(img)
    img = equalize(img)
    img = img/255
    return img
X_train=np.array(list(map(preprocessing,X_train))) # Iterate to do the
pre processing on all the images
X_validation=np.array(list(map(preprocessing,X_validation))) #same for
the validation and test set
X_test=np.array(list(map(preprocessing,X_test)))
```

Image augmentation

```
#Augmenting the image to make it more generic
dataGen= ImageDataGenerator(width_shift_range=0.1,    # 0.1 = 10%    IF
MORE THAN 1 E.G 10 THEN IT REFERS TO NO. OF  PIXELS EG 10 PIXELS
                                height_shift_range=0.1,
                                zoom_range=0.2,    # 0.2 MEANS CAN GO FROM 0.8
TO 1.2
                                shear_range=0.1,    # MAGNITUDE OF SHEAR ANGLE
                                rotation_range=10)  # DEGREES

dataGen.fit(X_train)
batches= dataGen.flow(X_train,y_train,batch_size=20)  # REQUESTING DATA
GENERATOR TO GENERATE IMAGES  BATCH SIZE = NO. OF IMAGES CREATED EACH
TIME ITS CALLED
X_batch,y_batch = next(batches)
# Showing the augmented image samples
fig,axs=plt.subplots(1,15,figsize=(20,5))
fig.tight_layout()
for i in range(15):
    axs[i].imshow(X_batch[i].reshape(imageDimensions[0],imageDimensions[1]))
    axs[i].axis('off')
plt.show()
y_train = to_categorical(y_train,noOfClasses)
y_validation = to_categorical(y_validation,noOfClasses)
y_test = to_categorical(y_test,noOfClasses)
```

Output:



RESULTS AND CONCLUSION

Phase - 1 Results and Conclusion

1. The data set does not have any irregularities or corrupt data.
2. The test train split passed the initial validation.
3. The distribution chart indicates sufficient images in all classes.
4. The colour space conversion of the images is successful.
5. The images are properly augmented and ready to be fed to the model.

FUTURE WORKS

Future works include:

1. Train the model
2. Test the model
3. Validate the model
4. Save the model
5. Create an interface to interact with the model
6. Add real-time interaction with the model using a webcam

REFERENCES

- [1]. Towards Reliable Traffic Sign Recognition
Benjamin Hoferlin - Intelligent Systems Group Universitat Stuttgart Stuttgart, Germany benjamin.hoeferlin@vis.uni-stuttgart.de
Klaus Zimmermann - European Technology Center (ESTEC) Sony Deutschland GmbH Stuttgart, Germany klaus.zimmermann@sony.de

- [2]. Detection And Recognition Of Indian Traffic Signs
Pritika Priya, Dhara Modha, Mansi Agrawal Department Of Information Technology, Bharati Vidyapeeth College Of Engineering For Women, Pune-43

- [3]. Deep Learning for Large-Scale Traffic-Sign Detection and Recognition
Domen Tabernik and Danijel Skocaj Faculty of Computer and Information Science, University of Ljubljana Vecna pot 113, 1000 Ljubljana {domen.tabernik,danijel.skocaj}@fri.uni-lj.si

- [4]. Traffic Sign Classification Using Deep Inception Based Convolutional Networks Mrinal Haloi IIT Guwahati1, mrinal.haloi11@gmail.com

- [5]. Indian Traffic Sign Detection and Classification Using Neural Networks
Arun Nandewal - CSE Department, arunnandewal@gmail.com
Abhishek Tripathi - IT Department, abhishek.tripathi2421@gmail.com
Satyam Chandra - EEE Department, satyam9871@gmail.com
NITK Surathkal

- [6]. Saha, S. (2018). A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way.

- [7]. Stanford University (2018). Introduction to Convolutional Neural Networks.

- [8]. Visin, V.D., Francesco (2016). English: Animation of a variation of the convolution operation. Blue maps are inputs, and cyan maps are outputs.
[online] Wikimedia Commons.

- [9]. Bonner, A. (2019). The Complete Beginner's Guide to Deep Learning: Convolutional Neural Networks.