TRAFFIC SIGN RECOGNITION WITH VOICE ALERT SYSTEM

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted by:

KODURU POOJA (18011A0420)

SAKA SRINIVASULU (19015A0422)

RESHMA (18011A0437)

SHARAFUDDIN (18011A0459)

Department of Electronics and Communication Engineering

Jawaharlal Nehru Technological University Hyderabad

University College of Engineering Hyderabad (Autonomous)

Kukatpally, Hyderabad-500085

2022

TRAFFIC SIGN RECOGNITION WITH VOICE ALERT SYSTEM

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted by:

KODURU POOJA (18011A0420)

SAKA SRINIVASULU (19015A0422)

RESHMA (18011A0437)

SHARAFUDDIN (18011A0459)

Under the esteemed guidance of

Dr. Makkena Madhavi Latha

Professor in ECE & Director of Academic and Planning JNTUH

Department of Electronics and Communication Engineering

Jawaharlal Nehru Technological University Hyderabad

University College of Engineering Hyderabad (Autonomous)

Kukatpally, Hyderabad-500085

2022

Jawaharlal Nehru Technological University Hyderabad

University College of Engineering Hyderabad - 500085

Department of Electronics and Communication Engineering

CERTIFICATE BY THE SUPERVISOR

This is to certify that the project entitled "**Traffic Sign Recognition With Voice Alert System**" is being submitted by -

Ms. KODURU POOJA (18011A0420)

Mr. SAKA SRINIVASULU (19015A0422)

Ms. RESHMA (18011A0437)

Mr. SHARAFUDDIN (18011A0459)

in partial fulfilment of the requirements for the award of degree in Bachelor of Technology in Electronics and Communication Engineering at the Jawaharlal Nehru Technological University Hyderabad University College of Engineering during the academic year 2021-22 is a bona fide work carried out under my guidance and supervision.

SUPERVISOR

Dr. Makkena Madhavi Latha

Professor in ECE

Director of Academic and Planning JNTUH

Jawaharlal Nehru Technological University Hyderabad

University College of Engineering Hyderabad - 500085

Department of Electronics and Communication Engineering

CERTIFICATE BY THE HEAD OF THE DEPARTMENT

This is to certify that the project report entitled "**Traffic Sign Recognition With Voice Alert System**" is Submitted by

Ms. KODURU POOJA (18011A0420)

Mr. SAKA SRINIVASULU (19015A0422)

Ms. RESHMA (18011A0437)

Mr. SHARAFUDDIN (18011A0459)

In partial fulfilment of the requirements for the award of degree in Bachelor of Technology in Electronics and Communication Engineering at the Jawaharlal Nehru Technological University Hyderabad University College of Engineering Technological University during the academic year 2021-22 is a Bona fide work carried out by them.

HEAD OF THE DEPARTMENT

Dr. A Rajani

Associate Professor & Head

Electronics & Communication Engineering

JNTUH College of Engineering Hyderabad

Jawaharlal Nehru Technological University Hyderabad

University College of Engineering Hyderabad - 500085

Department of Electronics and Communication Engineering

DECLARATION OF THE CANDIDATES

We hereby declare that the project report title "**Traffic Sign Recognition With Voice Alert System**" is a bona fide record work done and submitted under the esteemed guidance of **Dr. M. Madhavi Latha**, Professor in ECE & Director of Academic and Planning JNTUHUCEH, in partial fulfilment of the requirements for Project in Electronics and Communication Engineering at the Jawaharlal Nehru Technological University Hyderabad University College of Engineering during the academic year 2021-22 is a bona fide work carried out by us. The results have not been submitted in any other University or Institute for the award of degree or diploma.

Ms. KODURU POOJA (18011A0420)

Mr. SAKA SRINIVASULU(19015A0422)

Ms. RESHMA (18011A0437)

Mr. SHARAFUDDIN (18011A0459)

ACKNOWLEDGEMENT

We take this opportunity to record our gratitude to all those who helped us in the successful completion of this project.

There are many people who helped us directly or indirectly to complete our project successfully. Firstly, we would like to express our deep gratitude towards our guide **Dr. Makkena Madhavi Latha** Professor in ECE & Director of Academic and Planning, JNTUHUCEH for her support in the completion of our dissertation.

We would like to express our sincere thanks to **Dr. A Rajani, Head, Department of ECE** for providing the facilities to complete the dissertation. We would like to thank all our faculty and friends for their help and constructive criticism during the project period.

Finally, we are very much indebted to our parents for their moral support and encouragement to achieve goals.

Ms. KODURU POOJA (18011A0420)

Mr. SAKA SRINIVASULU(19015A0422)

Ms. RESHMA (18011A0437)

Mr. SHARAFUDDIN (18011A0459)

ABSTRACT

There have been a lot of technological advancements and cars with auto-pilot mode have come up. Autonomous vehicles have come into existence. There has been a boom in the self driving car industry. However, these features are available only in some high end cars which are not affordable to the masses. This Dissertation proposes to devise a system which helps in easing the job of driving to some extent.

To ensure a smooth and secure flow of traffic, road signs are essential. A major cause of road accidents is negligence in viewing the Traffic signboards and interpreting them incorrectly. The proposed system helps in recognizing the Traffic sign and sending a voice alert through the speaker to the driver so that he/ she may take necessary decisions.

The proposed system is trained using Convolutional Neural Network (CNN) which helps in traffic sign image recognition and classification. A set of classes are defined and trained on a particular dataset to make it more accurate. The German Traffic Sign Benchmarks Dataset is considered, which contains approximately 43 categories and 51,900 images of traffic signs. Following the detection of the sign by the system, a voice alert is sent through the speaker which notifies the driver. The aim of this system is to ensure the safety of the vehicle's driver, passengers, and pedestrians.

CONTENTS

List of Figures

Chapter 1

INTRODUCTION

This Chapter deals with the introduction of the proposed dissertation. In this chapter an overview of the current scenario of traffic sign recognition systems is given. Then the proposed methodology which fulfils the goal is discussed.

1.1 Overview

Road safety is attracting the attention of many researchers around the world since it is indispensable in protecting human life. To ensure a smooth and secure flow of traffic, road signs are essential. A major cause of road accidents is negligence in viewing the Traffic signboards and interpreting them incorrectly.

Driver assistance systems have played a very important role. For several years now, systems for the detection, classification, and recognition of road signs have become a very important research topic for researchers. From one research project to another, the authors have tried to improve the accuracy and recognition rate of these systems. To achieve these improvements, some researchers have turned to deep learning models.

1.2 Motivation

In current traffic management systems, there is a high probability that the driver may miss some of the traffic signs on the road because of overcrowding due to neighbouring vehicles. With the continuous growth of vehicle numbers in urban agglomerations around the world, this problem is only expected to grow worse. A visual-based traffic sign recognition system can be implemented on the automobile with an aim of detecting and recognizing all emerging traffic signs.

Nowadays, there is a lot of attention being given to the ability of the car to drive itself. One of the many important aspects for self-driving car is the ability for it to detect the traffic sign in order to provide safety and security for the people not only inside the car but also outside of it. The traffic environment consists of different aspects whose main purpose is to regulate flow of traffic, make sure each driver is adhering to the rules so as to provide a safe and secure environment to all the parties concerned.

1.3 Goal

The main objective of this work is to recognize the Traffic sign and send a voice alert through the speaker to the driver so that he/ she may take necessary action.

1.4 Methodology

It is proposed to design and construct a computer-based system which can automatically detect the road signs so as to provide assistance to the user or the machine so that they can take appropriate actions.

It is implemented by first Uploading / Capturing the Image and then Pre-process it using various image pre-processing techniques. Later, using LENET-5 deep neural network the traffic sign is detected. Then by using softmax layer of the neural network the traffic sign is recognised, and a voice alert is sent.

Chapter 2

BACKGROUND

This Chapter deals with the background of the proposed dissertation. In this chapter a brief overview of existing systems is done. Then the need for traffic sign recognition is explained.

2.1 Existing Systems

Although traffic sign recognition has gained a plethora of popularity in driver assistant system, there are still numerous difficulties for identifying real-world traffic signs by using computer algorithms due to various size of targets, color deterioration and partial occlusion. In order to deal with these obstacles, many approaches and algorithms have been proposed -

- In the past, traffic sign detection mainly relied on traditional object detection algorithms. The pipeline of traffic sign detection normally utilized hand-crafted features to extract region proposals, and then combined classifiers to filter out the negatives.
- Recently, deep learning methods are emerging, and various cutting-edge approaches have been widely applied into this area, such as deep convolutional networks (CNNs).
- CNNs have brought possibility of learning features from giant amount of data without pre-processing, which avoids the process of designing hand-crafted features and absorbs more generalized features .

2.2 Need of Traffic Sign Recognition System

Traffic-sign recognition (TSR) is a technology by which a vehicle is able to recognize the traffic signs put on the road like "speed limit" or "children" or "turn ahead". This is part of the features collectively called ADAS. The technology is being developed by a variety of automotive suppliers. It uses image processing techniques to detect the traffic signs. The detection methods can be generally divided into color based, shape based and learning based methods.

Traffic signs can be analyzed using forward-facing cameras in many modern cars, vehicles and trucks. One of the basic use cases of a traffic-sign recognition system is for speed limits. Most of the GPS data would procure speed information, but additional speed limit traffic signs can also be used to extract information and display it in the dashboard of the car to alert the driver about the road sign. This is an advanced driver-assistance feature available in most high-end cars, mainly in European vehicles.

Need for TSR

Safety : Traffic sign recognition considerably enhances safety, as it allows drivers to concentrate on the traffic in complicated situations. The system also helps motorists to keep to the speed limit.

Function : Warning functions can be implemented on the basis of traffic sign recognised , for example before breaking the speed limit, overtaking where this is forbidden or ignoring stop/no entry signs. Automatic adjustment of vehicle speed to the applicable speed limit is also possible. This "Intelligent Speed Assist" and traffic sign display also form part of the Euro NCAP Ratings.

Alert the Driver : Whenever a traffic sign is recognised, a voice alert is sent to the driver via the speaker. By this way it facilitates the driver with required traffic sign information in hand. This way he/ she gets to know what they must do and also it ensures safety.

Affordable TSR system : The existing TSR systems are costly, and deployed only in very high priced cars like AUDI, BMW etc. So this calls for a traffic sign recognition system which can be included in all vehicles for both safety and better functionality.

Chapter 3

TRAFFIC SIGN RECOGNITION

SYSTEM DESIGN

This Chapter deals with the design of the proposed dissertation. In this chapter the various steps in the project design as shown in figure 3.1 are explained in detail.

3.1 System Design :

Firstly, a Dataset is collected as shown in the flow diagram figure 3.1. The Dataset can be either standard or developed. Then the dataset is used to train the LeNet cnn model. Once the training is done, the model will recognise the traffic signs which is sent as voice alert to the driver as shown in the last block of figure 3.1.

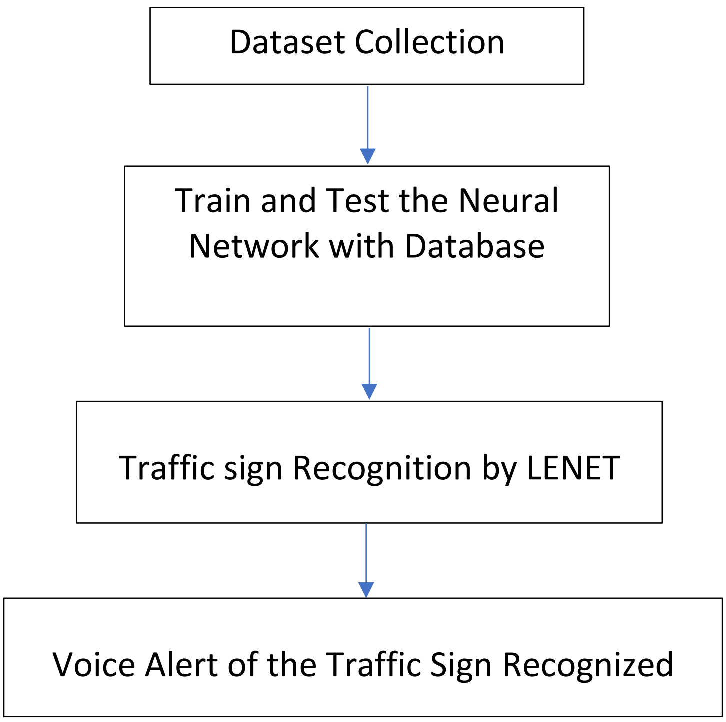


Figure 3.1 : Flow diagram

3.2 Dataset Collection

3.2.1 Gtsrb Dataset

The **German Traffic Sign Recognition Benchmark** (**GTSRB**) contains 43 classes of traffic signs, split into 39,209 training images and 12,630 test images. Image sizes vary between 15x15 to 250x250 pixels. The images have varying light conditions and rich backgrounds. The various classes in the dataset can be seen in figure 3.2**.**

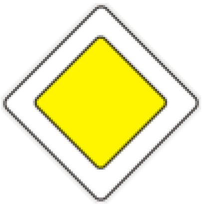
Figure 3.2 : GTSRB Dataset

3.2.2 Developed Dataset

Along with GTSRB Dataset, a model trained separately with our own custom dataset is also proposed. This dataset has 5 categories of traffic signs -

a) Stop Sign
b) Priority Road Sign
c) Speed Limit 20 Km/hr Sign
d) Ahead Only Sign
e) Pedestrians Sign

(a)



(b)

(c)

(d)

(e)

Figure 3.3 : Developed Dataset

This Dataset consists of 115 images, out of which 100 images are used for training the model and remaining are used for testing the model. The images are of varying sizes and lighting conditions. A custom dataset is also considered to verify the model's performance with a smaller dataset, in comparison to a large and standard dataset. The 5 categories in this dataset can be viewed in figure 3.3

3.3 Training and Testing

For training and testing purposes the model, the data is to be broken down into three distinct dataset splits as shown in figure 3.4.

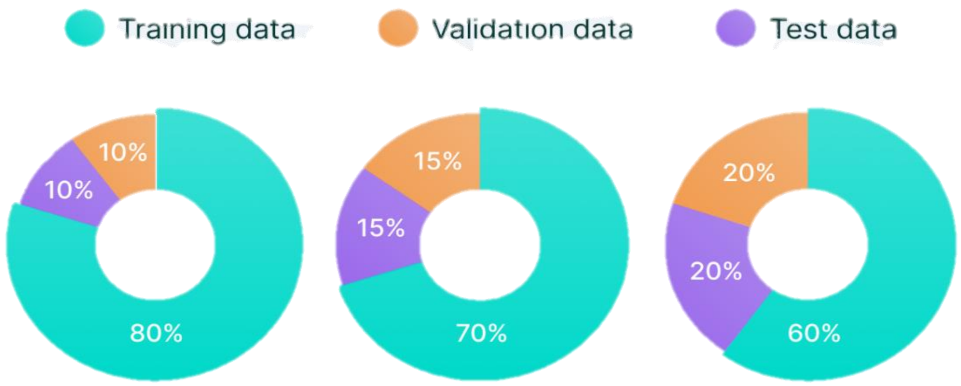


Figure 3.4 : Splitting of Dataset

(Reference : https://www.v7labs.com/blog/train-validation-test-set)

The Training Set

It is the set of data that is used to train and make the model learn the hidden features/patterns in the data. In each epoch, the same training data is fed to the neural network architecture repeatedly, and the model continues to learn the features of the data. The training set should have a diversified set of inputs so that the model is trained in all scenarios and can predict any unseen data sample that may appear in the future.

The Validation Set

The validation set is a set of data, separate from the training set, that is used to validate our model performance during training. This validation process gives information that helps us tune the model's hyperparameters and configurations accordingly. It is like a critic telling us whether the training is moving in the right direction or not. The model is trained on the training set, and, simultaneously, the model evaluation is performed on the validation set after every epoch. The main idea of splitting the dataset into a validation set is to prevent our model from overfitting i.e., the model becomes really good at classifying the samples in the training set but cannot generalize and make accurate classifications on the data it has not seen before.

The Test Set

The test set is a separate set of data used to test the model after completing the training. It provides an unbiased final model performance metric in terms of accuracy, precision, etc. To put it simply, it answers the question of "How well does the model perform?"

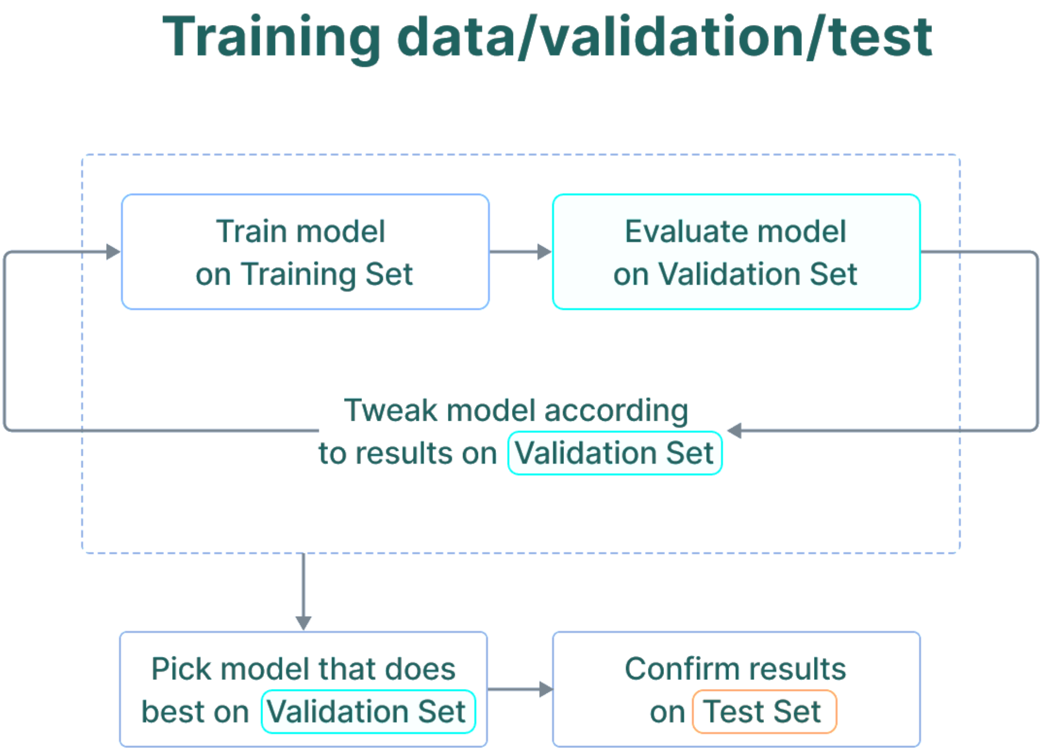


Figure 3.5 : Training and Testing a Model

(Reference : https://www.v7labs.com/blog/train-validation-test-set)

Performance Measures

A classification task in which a machine learning system observes tumors and has to predict whether these tumors are benign or malignant is being considered. **Accuracy (ACC)**, or the fraction of instances that were classified correctly, is an obvious measure of the program's performance. We can measure each of the possible prediction outcomes to create different snapshots of the classifier's performance. When the system correctly classifies a tumor as being malignant, the prediction is called a **true positive**. When the system incorrectly classifies a benign tumor as being malignant, the prediction is a **false positive**. Similarly, a **false negative** is an incorrect prediction that the tumor is benign, and a **true negative** is a correct prediction that a tumor is benign. These four outcomes can be used to calculate several common measures of classification performance, like accuracy, precision, recall and so on. Accuracy is calculated with the following formula −

$$\mathbf{ACC = (TP + TN)/(TP + TN + FP + FN)}$$

Where, TP is the number of true positives

TN is the number of true negatives

FP is the number of false positives

FN is the number of false negatives.

Precision (PREC) is the fraction of the tumors that were predicted to be malignant that are actually malignant. Precision is calculated with the following formula −

$$\mathbf{PREC = TP/(TP + FP)}$$

Recall (R)is the fraction of malignant tumors that the system identified. Recall is calculated with the following formula −

$$\mathbf{R = TP/(TP + FN)}$$

3.4 LeNet 5

Lenet-5 is one of the earliest pre-trained models proposed by Yann LeCun and others in the year 1998. They used this architecture for recognizing the handwritten and machine-printed characters. The main reason behind the popularity of this model was its simple and straightforward architecture. It is a multi-layer convolution neural network for image classification. It has four sets of convolution layers with a combination of max pooling and dropout layers. After the convolution and max pooling layers, we have flatten layer and two fully connected layers. At last, a Softmax classifier which classifies the images into respective class.

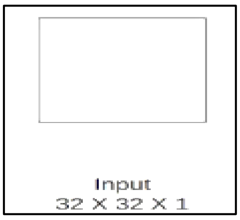


Figure 3.6 : LeNet Architecture

(Reference: https://www.analyticsvidhya.com/blog/2021/03/the-architecture-of-lenet-5/)

The input to this model is a 32 X 32 grayscale image hence the number of channels is one as shown in figure 3.6.

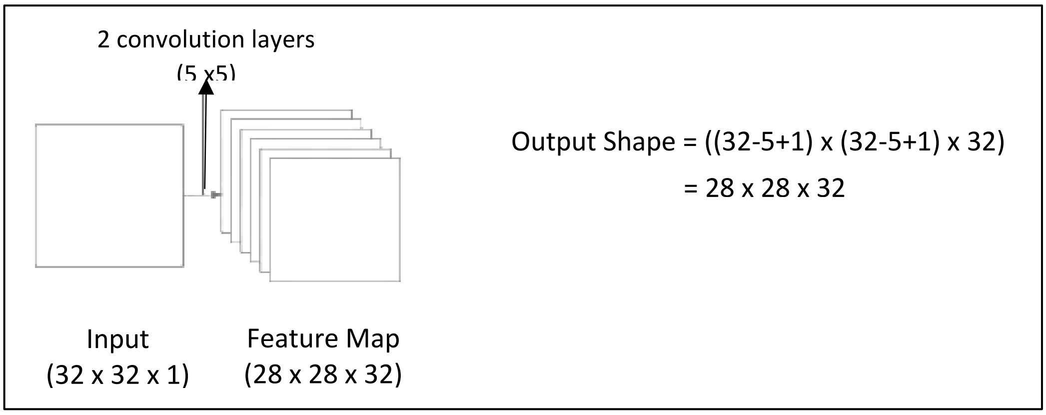


Figure 3.7 : LeNet Architecture

(Reference: https://www.analyticsvidhya.com/blog/2021/03/the-architecture-of-lenet-5/)

Then the first convolution operation with the filter size 5X5 is applied. And there are 32 such filters. As a result, the feature map is of size 28X28X32 as shown in figure 3.7. Here the number of channels is equal to the number of filters applied.

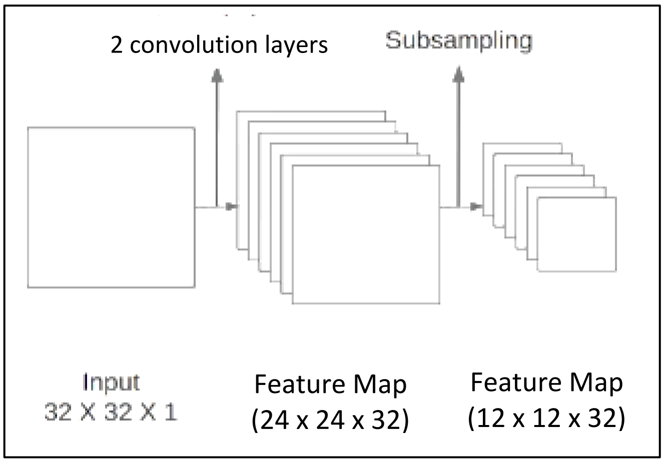


Figure 3.8 : LeNet Architecture

(Reference: https://www.analyticsvidhya.com/blog/2021/03/the-architecture-of-lenet-5/)

After the first convolution operation the size of feature map is 28 x 28 x 32. Then a second convolution layer with filter size 5 x 5 and 32 such filters is applied. As a result, we a feature map of size 24X24X32 is obtained. Then max pooling is applied and the size of the feature map is reduced by half .That is 12 x 12 x 32 as shown in figure 3.8. Note that, the number of channels is intact.

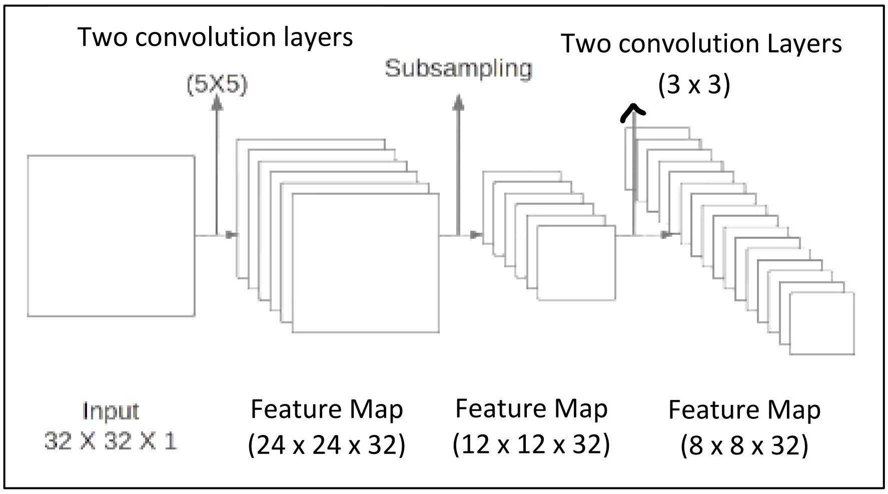


Figure 3.9 : LeNet Architecture

(Reference: https://www.analyticsvidhya.com/blog/2021/03/the-architecture-of-lenet-5/)

Next, two convolution layers with 64 filters of size 3X3 are applied. Again the feature map changed it is 8X8X64. The output size is calculated in a similar manner. After this, again a max pooling or subsampling layer is used, which again reduces the size of the feature map by half i.e 4X4X64 as shown in figure 3.9.

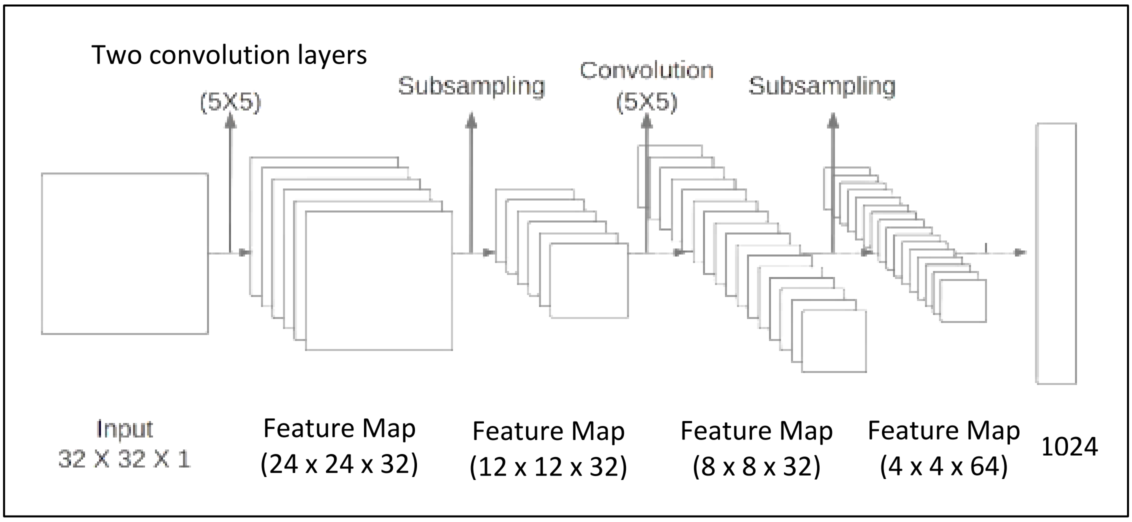


Figure 3.10 : LeNet Architecture

(Reference: https://www.analyticsvidhya.com/blog/2021/03/the-architecture-of-lenet-5/)

After which flatten result is 1024 values as shown in figure 3.10. After these convolution layers, there is a fully connected layer with 512 neurons. At last, there is an output layer with 43 neurons since the data have 43 classes. Here is the final architecture of the Lenet-5 model as shown in figure 3.11.

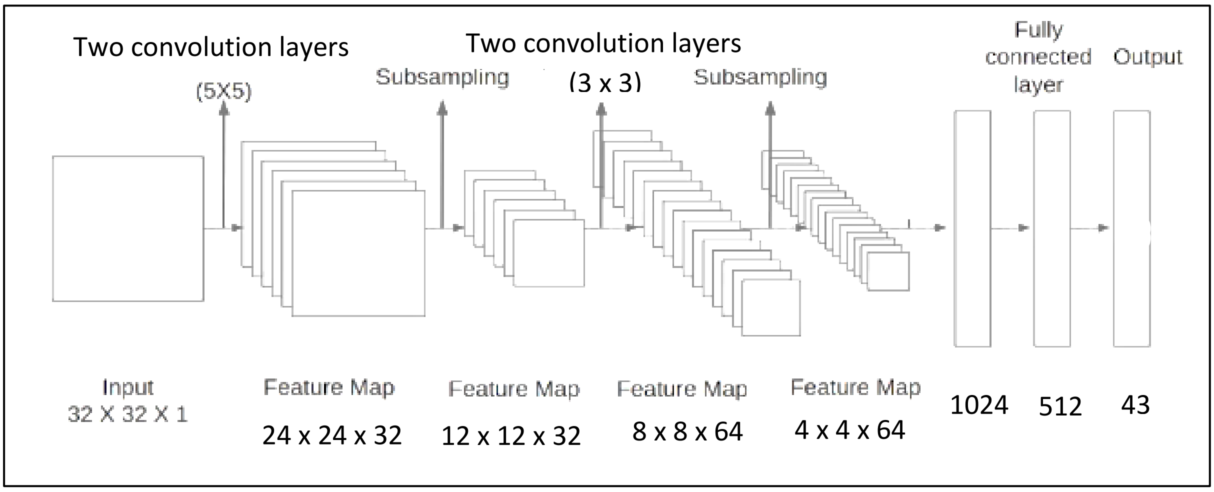


Figure 3.11 : LeNet Architecture

(Reference: https://www.analyticsvidhya.com/blog/2021/03/the-architecture-of-lenet-5/)

3.4.1 Classification

LeNet Architecture is as shown in figure 3.12. The Last layer of the CNN model is a softmax layer. Softmax assigns decimal probabilities to each class in a multi-class problem. Those decimal probabilities must add up to 1.0. This additional constraint helps training converge more quickly than it otherwise would.

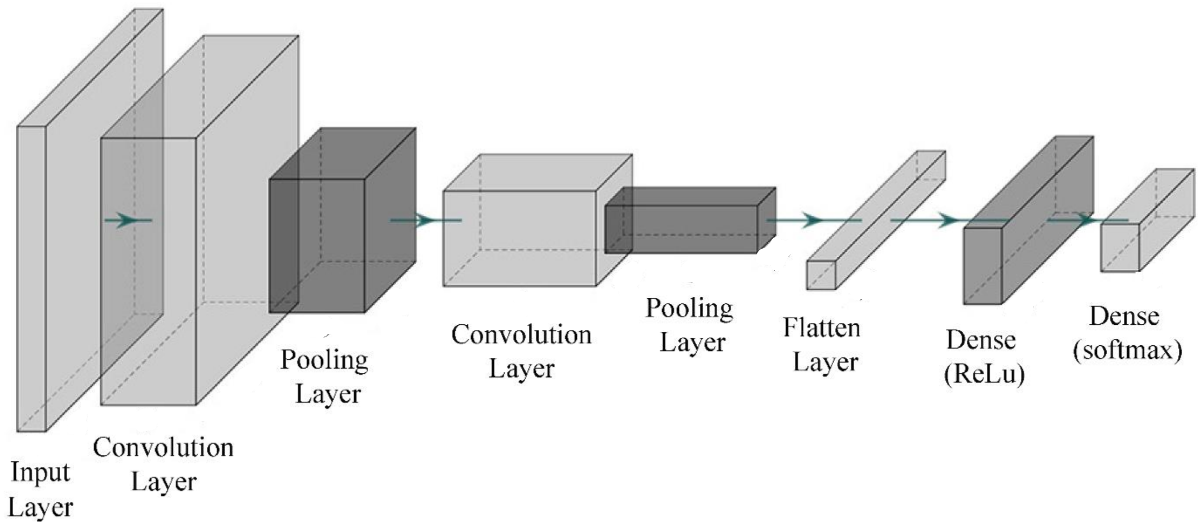


Figure 3.12 : LENET Model

(Reference : https://i.stack.imgur.com/Bt07P.png)

The Softmax Layer Classifies the input image into one of the total classes. The traffic sign recognised is then outputted through a Voice - Alert .

Chapter 4

IMPLEMENTATION

This Chapter Deals with the implementation of the proposed dissertation. In this chapter the various steps performed to execute the proposed project are explained in detail.

Step 1 : Importing All The Required Libraries

Here all the required python libraries like opencv , numpy , keras, tensorflow are imported successfully

Opencv

OpenCV provides a real-time optimized Computer Vision library, tools, and hardware. It also supports model execution for Machine Learning (ML) . OpenCV is an open-source BSD-licensed library that includes several hundreds of computer vision algorithms. OpenCV has a modular structure, which means that the package includes several shared or static libraries. The following modules are available:

- Object Detection (objdetect) - detection of objects and instances of the predefined classes (for example, faces, eyes, mugs, people, cars, and so on).
- Image Processing (imgproc) - an image processing module that includes linear and non-linear image filtering, geometrical image transformations, color space conversion, histograms, and so on.
- Video Analysis (video)
- Core functionality (core)
- 2D Features Framework (features2d)
- High-level GUI (high gui)
- Video I/O (video io)

Numpy

- Provides An array object of arbitrary homogeneous items
- Fast mathematical operations over arrays
- Main Functionalities are Linear Algebra, Fourier Transforms, Random Number Generation

Keras

Keras is a high-level, deep learning API developed by Google for implementing neural networks. It is written in Python and is used to make the implementation of neural networks easy. It also supports multiple backend neural network computation.

TensorFlow has adopted Keras as its official high-level API. Keras is embedded in TensorFlow and can be used to perform deep learning fast as it provides inbuilt modules for all neural network computations. At the same time, computation involving tensors, computation graphs, sessions, etc can be custom made using the Tensorflow Core API, which gives you total flexibility and control over your application and lets you implement your ideas in a relatively short time.

Step 2 : Extracting Dataset

- The ‘train’ folder contains 43 folders each representing a different class. The range of the folder is from 0 to 42. With the help of the OS module, iterate over all the classes and append images and their respective labels in the data and labels list.
- The PIL library is used to open image content into an array.
- Then open each folder and extract the images and store in the list data[]
- Once the data is extracted, the dataset is split into train and test data.

Step 3 : Building Cnn Model

- The architecture of the model is "LeNet" as discussed in **Section 3.4**
- First a “Sequential” model is created and then all the layers are added – convolutional, maxpool, dropout etc as per the CNN architecture desired.
- Once all the layers are added the model is compiled.

Step 4 : Training The Model

- The test data obtained in splitting is taken as validation.
- The model is trained with the training dataset divided and validate using validating dataset.
- Epoch is the number of times the neural network is trained with the whole dataset.
- To train the model 20 epochs are used.

Step 5 : Testing The Model

- The Test dataset is extracted in this step.
- The process is similar to training dataset extraction.
- Then prediction is made for each input in test dataset and compared with actual outputs to obtain the accuracy.

Step 6 : Performance Metrics

- In this step confusion matrix is calculated to obtain the TP,TN,FP,FN values.
- A seaborn is also plotted to visualise the confusion matrix .
- Using the TP,TN,FP,FN values Precision, Recall and Accuracy are calculated as discussed in **Section 3.3**

Step 7 : Developing GUI

- In this step a GUI is developed using the Tkinter library
- An upload image button is provided on the GUI
- Once the image is uploaded, the trained model is loaded to classify the traffic sign.

Step 8 : Capture Image

- The Image can be directly uploaded into the GUI.
- The image can also be captured from the webcam and further processed.
- Even a video input can be given. In this particular case frames are extracted, and output is obtained for each frame.

Step 9 : Preprocess The Image

- The image is read in RGB
- Then the red colour channel is split.
- The contours for sign are estimated by morphology and thresholding.
- The biggest sign contour is selected
- The coordinates x,y,w,h of boundary Box are calculated .
- The sign is cropped from the image by boundary Box Coordinate
- Then CNN is applied on the sign.

Step 10 : Classify The Sign

- The pre-processed image is given as input to the trained model.
- Then the softmax layer of the model classifies the traffic sign.

Step 11 : Voice Alert

In this step the label/ name of the traffic sign recognised is read out by the Python Speech Engine.

Chapter 5

RESULTS AND ANALYSIS

This Chapter Deals with the results of the proposed dissertation. In this chapter the results obtained in various stages of the project design are shown and analysed in detail.

5.1 Loading images from database

5.1.1 GTSRB Dataset

- GTSRB is a standard dataset downloaded from kaggle for training.
- There are 43 classes in total each with unique traffic sign as shown in figure 5.1.
- The images contain one traffic sign each.
- There are a total of 51,900 images in the dataset. 32,500 images are used for training
- Images contain a border of 10 % around the actual traffic sign (at least 5 pixels) to allow for edge-based approaches
- Image sizes vary between 15 x 15 to 250 x 250 pixels
- Images are not necessarily squared.
- The actual traffic sign is not necessarily centered within the image. This is true for images that were close to the image border in the full camera image.

5.1.2 Developed Dataset

- There are total 5 categories of traffic signs as shown in figure 5.2
- There are a total of 115 images, out of which 100 are used for training.
- The Images contain one traffic sign each and they are not squared.
- The Image sizes are not fixed but varying.
- The Images are collected from various websites.

Figure 5.1 : GTSRB Dataset

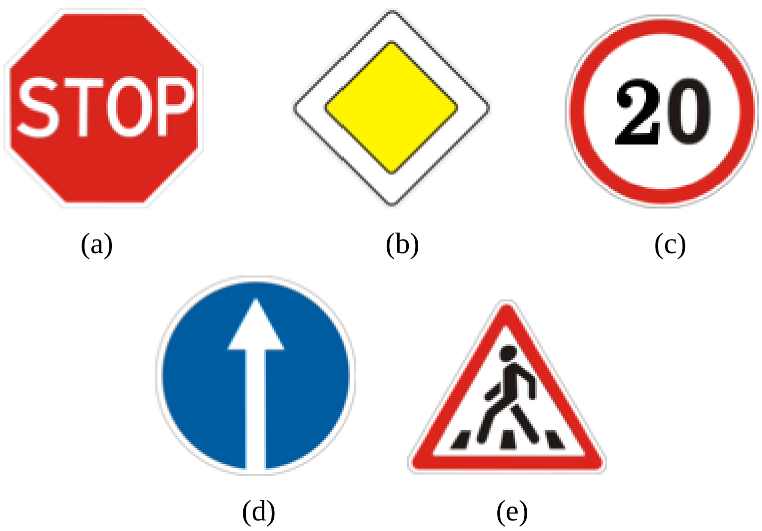


Figure 5.2 : Developed Dataset

5.2 Training The Model

- The model Summary is shown in figure 5.3
- The image input is resized to 30 X 30.
- The Cnn model used is based on LeNet 5. It has two convolutional layers of size 5 x 5 with 32 filters
- Followed by them are maxpool and dropout layers
- The next two layers are convolutional layers of size 3 x 3 with 64 filters.
- Followed by them are maxpool and dropout layers.
- The next layer is a flatten layer followed by a dense layer.
- The last Layer is softmax which performs the classification.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 26, 26, 32)        2432

 conv2d_1 (Conv2D)           (None, 22, 22, 32)        25632

 max_pooling2d (MaxPooling2D  (None, 11, 11, 32)       0
 )

 dropout (Dropout)           (None, 11, 11, 32)        0

 conv2d_2 (Conv2D)           (None, 9, 9, 64)          18496

 conv2d_3 (Conv2D)           (None, 7, 7, 64)          36928

 max_pooling2d_1 (MaxPooling  (None, 3, 3, 64)         0
 2D)

 dropout_1 (Dropout)         (None, 3, 3, 64)          0

 flatten (Flatten)           (None, 576)               0

 dense (Dense)               (None, 256)               147712

 dropout_2 (Dropout)         (None, 256)               0

 dense_1 (Dense)             (None, 43)                11051

=================================================================
Total params: 242,251
Trainable params: 242,251
Non-trainable params: 0
_________________________________________________________________
```

Figure 5.3 : Model Summary

- After Compiling the model, it is trained with 20 epochs with a batch size of 32.
- Epoch is the number of times the neural network is trained with the whole dataset.
- Usually with every epoch increasing, loss should be going lower and accuracy should be going higher. But with val_loss and val_acc, many cases can be possible like below:
    - val_loss starts increasing, val_acc starts decreasing. This means model is cramming values not learning .
    - val_loss starts increasing, val_acc also increases.This could be case of overfitting or diverse probability values in cases where softmax is being used in output layer .
    - val_loss starts decreasing, val_acc starts increasing. This is also fine as that means model built is learning and working fine.
- The validation loss and accuracy variations with each epoch is shown in figure 5.4

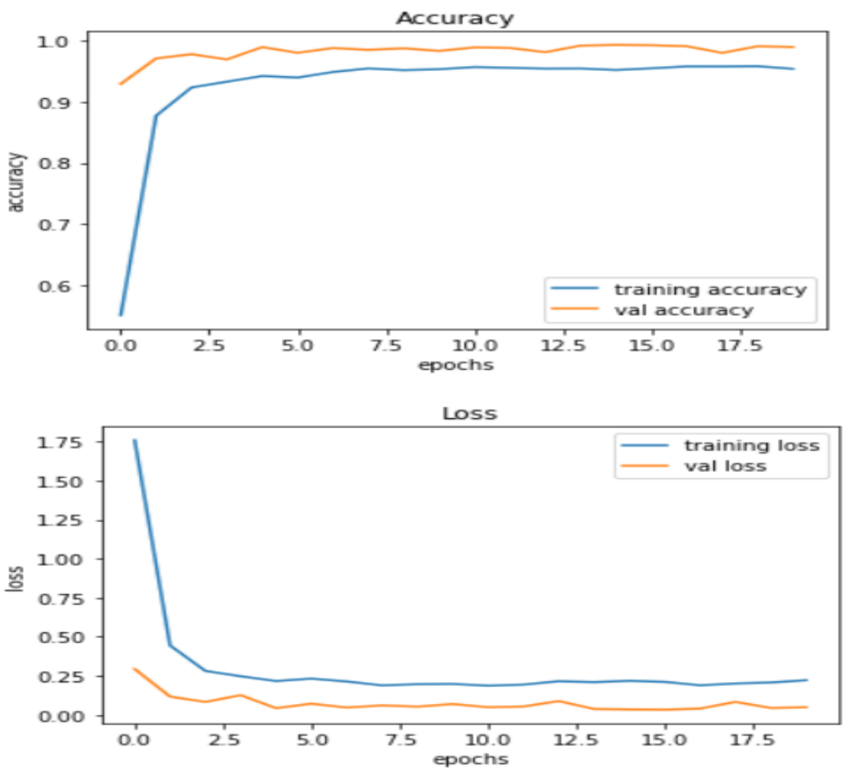


Figure 5.4 : Training Loss and Accuracy

5.3 Testing The Model

- After training the model with train data. Testing is done with test data to check for accuracy.
- The accuracy_score method is used to calculate the accuracy of either the faction or count of correct prediction in Python Scikit learn.
- Mathematically it represents the ratio of the sum of true positives and true negatives out of all the predictions.
- A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known.
- From the TP,TN,FP,FN values Precision , recall and accuracy are calculated as discussed in section 3.3
- A **true positive** is an outcome where the model correctly predicts the positive class.
- Similarly, a **true negative** is an outcome where the model correctly predicts the negative class.
- A **false positive** is an outcome where the model incorrectly predicts the positive class.
- And a **false negative** is an outcome where the model incorrectly predicts the negative class.
- TPR or True Positive Rate is also known as Recall.
- TNR or True Negative Rate is also known as Precision.
- The Diagonal values in the confusion matrix indicate the true positive values.
- In the seaborn confusion matrix as shown in figure 5.5 and figure 5.6 darker colours indicate greater value in the confusion matrix

5.3.1 GTSRB Dataset

Accuracy Score : 0.9566904196357878

Confusion Matrix -

```
[[ 60   0   0 ...   0   0   0]
 [  0 707   2 ...   1   0   0]
 [  0   3 737 ...   1   0   0]
 ...
 [  0   0   0 ...  82   0   1]
 [  0   0   0 ...   0  54   0]
 [  0   0   0 ...   0   0  87]]
```

Performance Metrics from Confusion Matrix –

TPR = 0.9566904196357878

TNR = 0.9989688195151378

FPR = 0.0010311804848621951

FNR = 0.043309580364212195

Accuracy = 0.9979856009132925

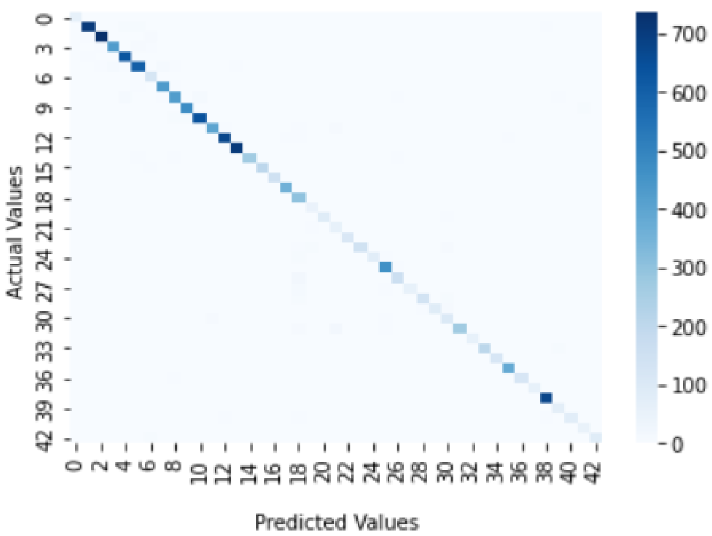


Figure 5.5 : Seaborn Confusion Matrix

5.3.2 Developed Dataset

Accuracy Score : 0.9333333333333333

Confusion Matrix -

```
[[2 0 0 0 0]
 [0 3 0 0 0]
 [0 0 3 0 0]
 [0 0 0 3 0]
 [1 0 0 0 3]]
```

Performance Metrics from Confusion Matrix –

TPR = 0.9333333333333333

TNR = 0.9833333333333333

FPR = 0.016666666666666666

FNR = 0.06666666666666667

Accuracy = 0.9733333333333334

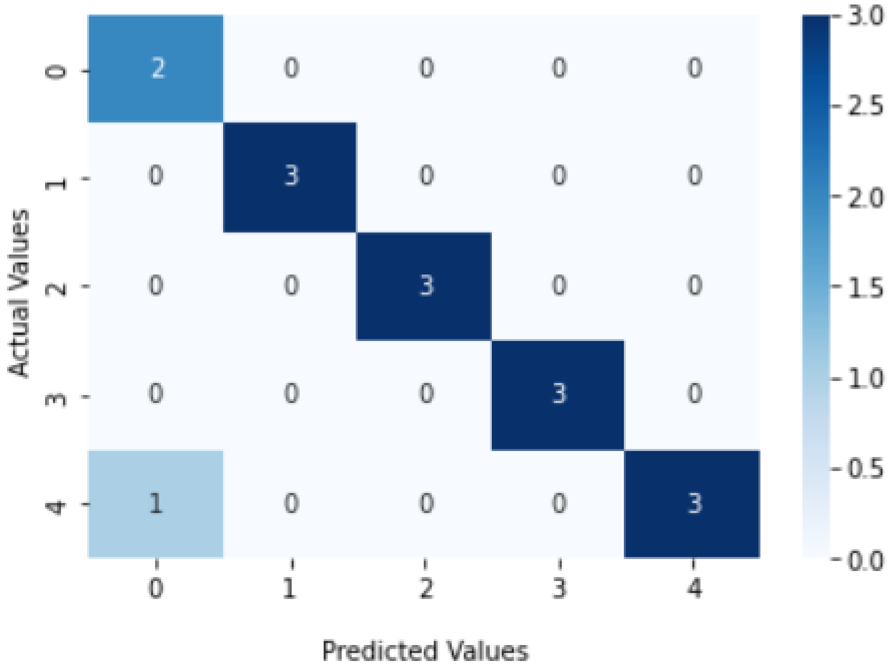


Figure 5.6 : Seaborn Confusion Matrix

5.4 Visualisation of Layers in LeNet

- For the visualisation of various layers output in the cnn model considered the input image is as shown in figure 5.7
- Outputs of each filter of each convolution layer are plotted
- As the number of layers increase, the size of image decreases, increasing the processing speed
- Also the depth of the image increases with the number of layers.
- Here the first convolution layer (figure 5.8) attempts to preserve everything from an image.
- As we go deeper, the convolutional layers focus on the traffic sign board completely (see figure 5.9).

Figure 5.7 : Input Image for LeNet

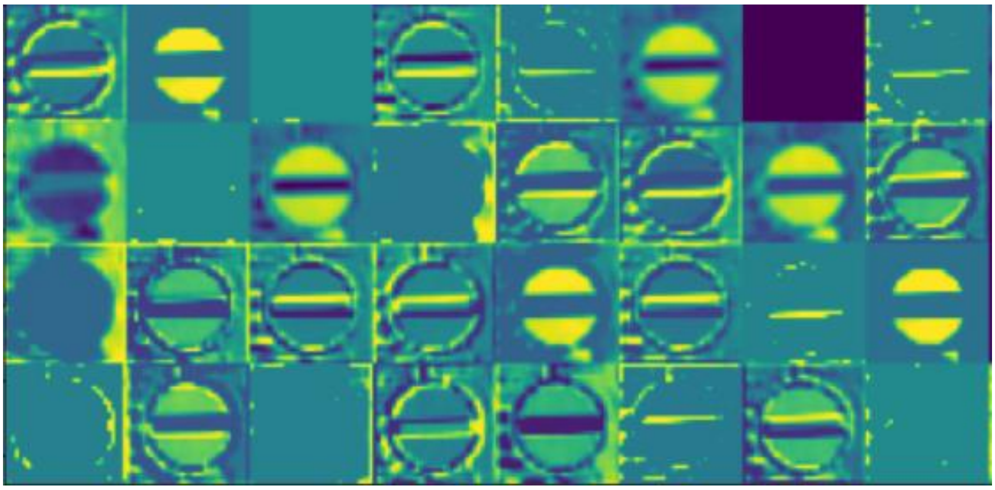


Figure 5.8 : Convolutional layer1

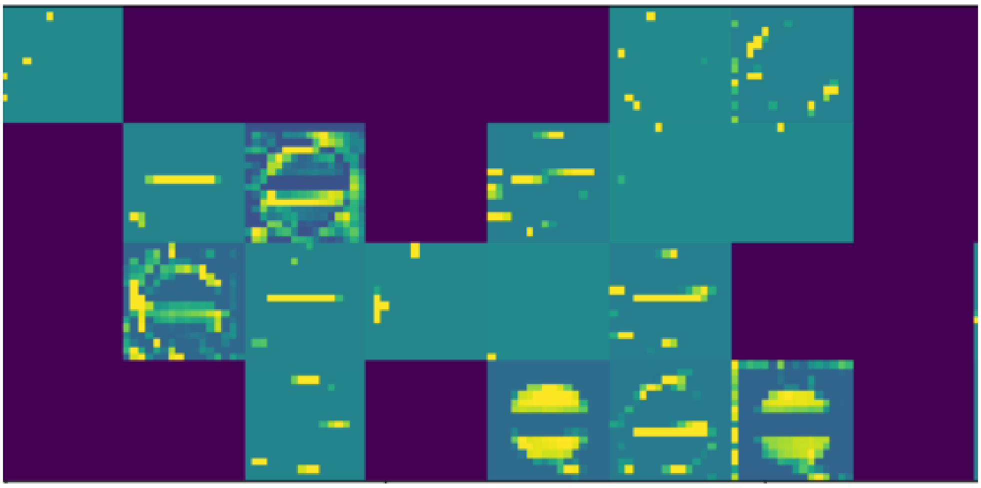


Figure 5.9 : Convolutional layer2 output

- Going more deeper it is beyond our understanding to the naked eye about what the layers try to preserve (see figure 5.10).
- Going towards the layers to the last we see that the traffic sign only is tried to be preserved (see figure 5.11).
- It is seen that some of the filter outputs are blank in the layers. That is because there isn’t any such feature in the image, which that particular filter wants to preserve
- The sparsity of the activations is increasing with the depth of the layer: in the first layer, all filters are activated by the input image, but in the following layers more and more filters are blank. This means that the pattern encoded by the filter isn't found in the input image.
- The features extracted by a layer get increasingly abstract with the depth of the layer.
- The activations of layers higher-up carry less and less information about the specific input being seen, and more and more information about the target (in our case, recognise a traffic sign).

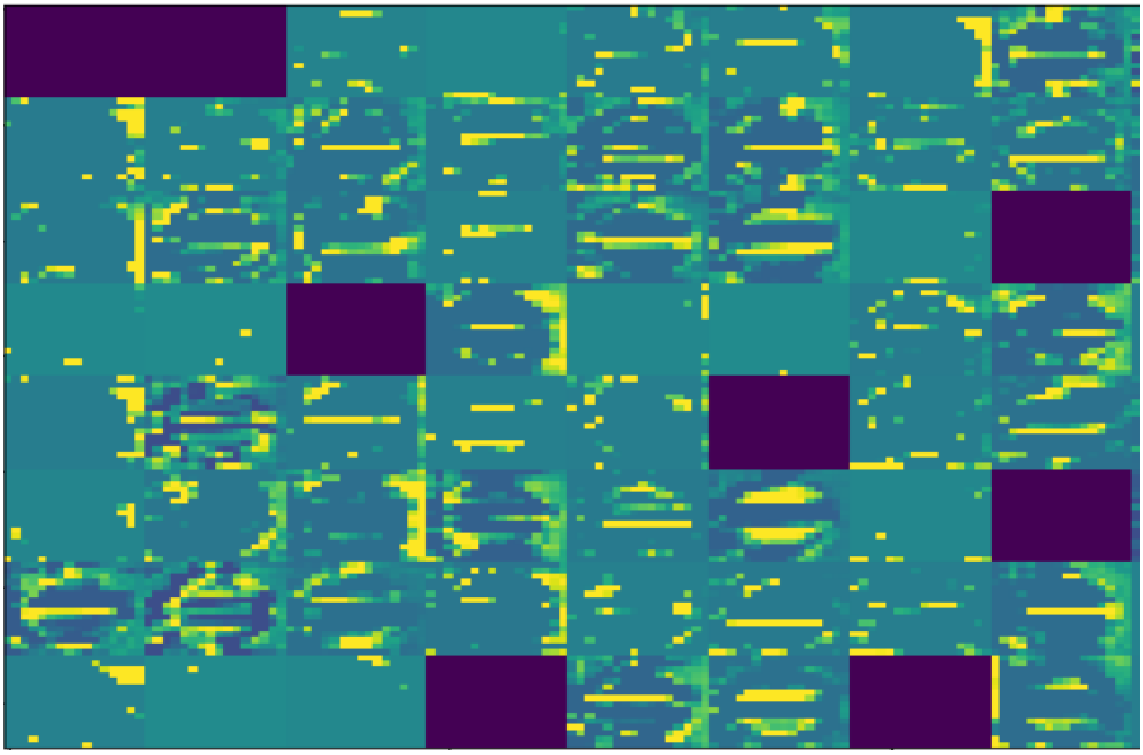


Figure 5.10 : Convolutional layer3 output

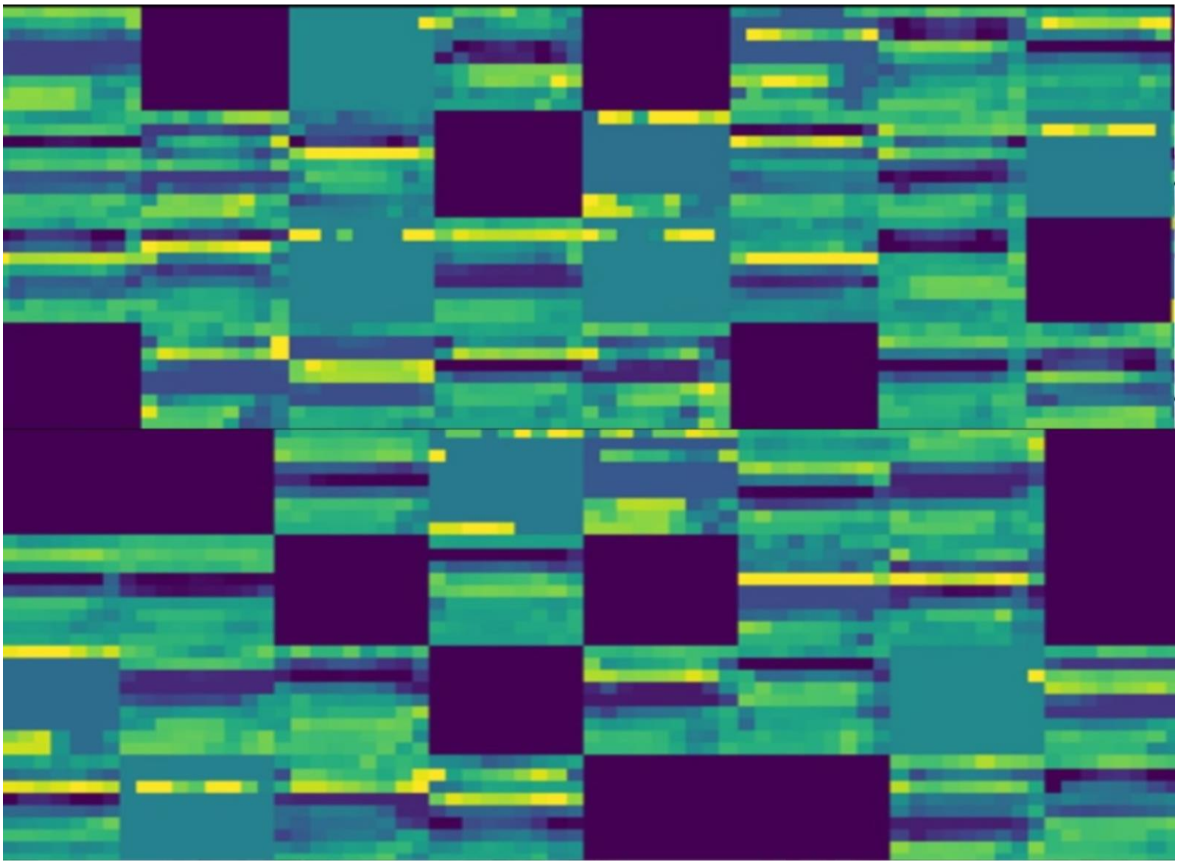
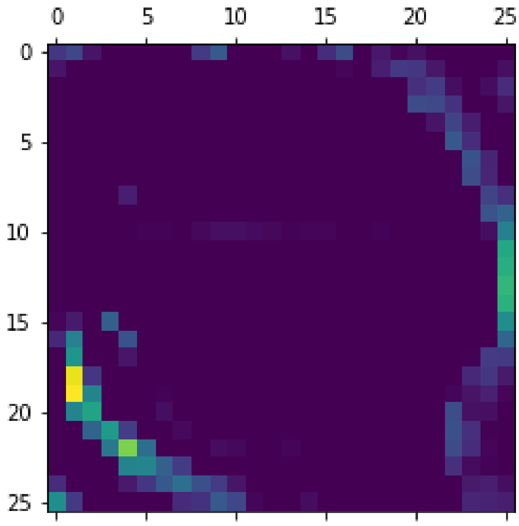
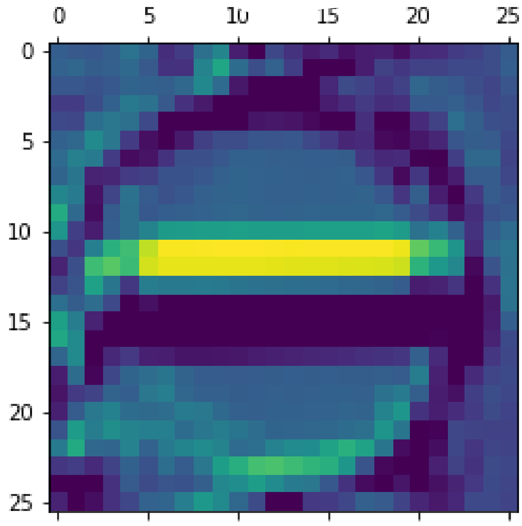


Figure 5.11 : Convolutional layer4 output

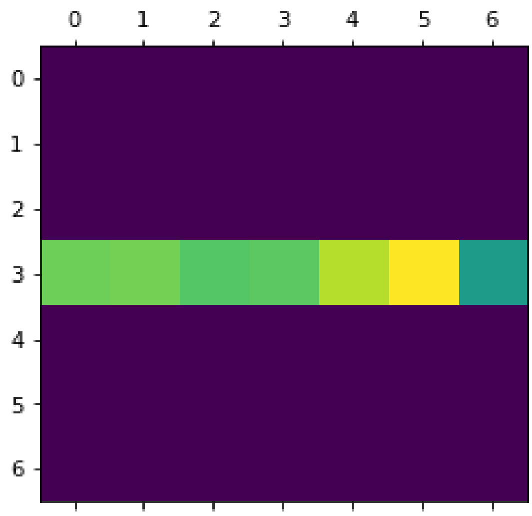
- Specific filter outputs are as shown below in figure 5.12
- The sixth filter of first layer tries to preserve the edges of sign board as shown in figure 5.12(a).
- The fourteenth filter of first layer tries to preserve the sign board as a whole as shown in figure 5.12(a).
- Whereas going into deeper layers as last convolutional layer tries to preserve the sign of sign board as shown in figure 5.12(a).



(a) 6th filter output of 1st conv layer



(b) 14th filter output of 1st conv layer



(c) 49th filter output of 4th conv layer

Figure 5.12 : Specific Layers output

5.5 Detection

- The input image is pre-processed as discussed in Chapter 4. Once the image is processed the sign board position is precisely located as in figure 5.13 by using the coordinates obtained
- An other instance of traffic sign board detection is as shown in the figure 5.14.

Figure 5.13 : Sign board detection 1

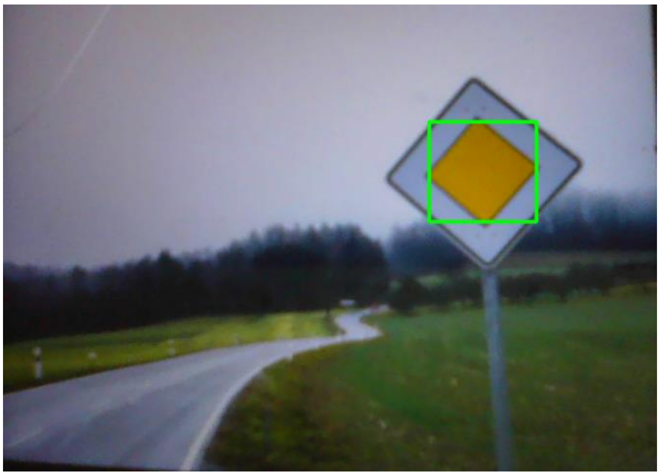


Figure 5.14 : Sign board detection 2

5.6 GUI Output

- The GUI is created using the tkinter library as discussed in chapter 4.
- Various buttons are added onto the window and callback function is used to load the model and predict th traffic sign
- The GUI initially consists of only an "upload image " button on it, where an image can be uploaded as shown in figure 5.15
- Once an image is uploaded, the gui displays a second button 'Classify image' as shown in figure 5.16
- By clicking the classify image button, the input image is classified into one of the 43 traffic sign classes and output is displayed above the image. Also a voice alert is initiated through the speaker as shown in figure 5.17
- Figure 5.18 shows an other output for a sign uploaded on to GUI

Figure 5.15 : GUI

Figure 5.16 : GUI after uploading an image

Figure 5.17 : Classification by GUI

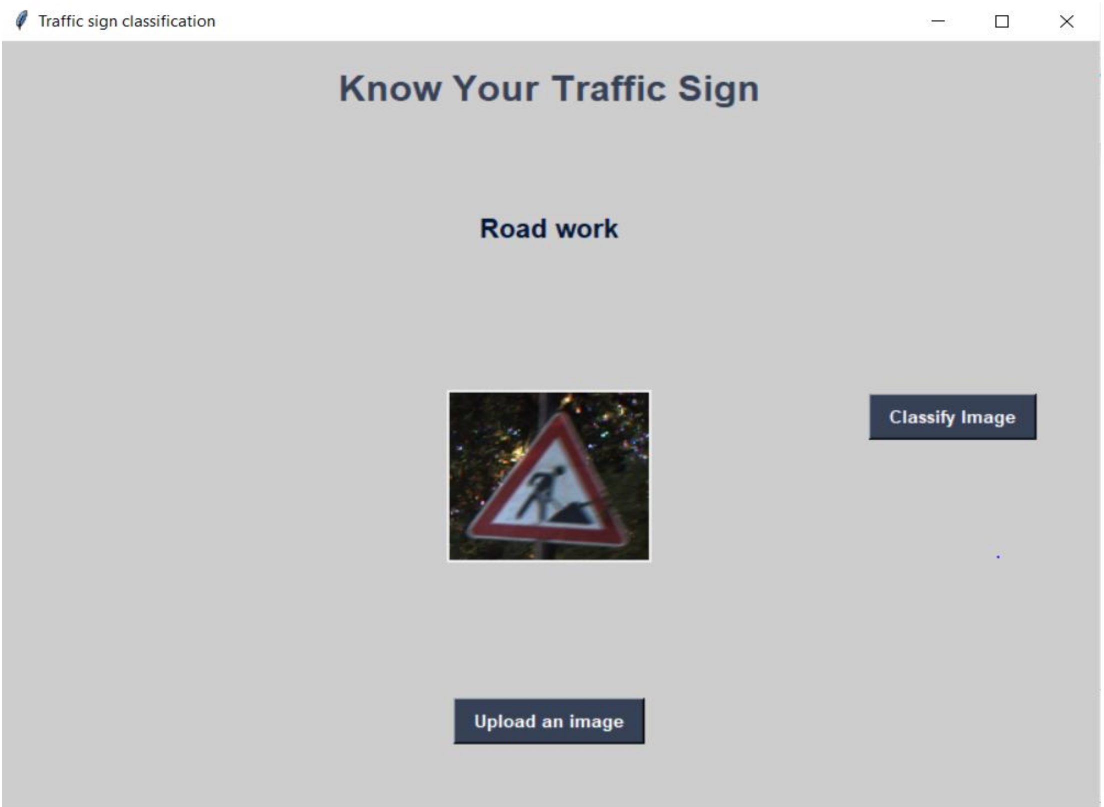


Figure 5.18 : Classification by GUI

5.7 Live Capture

- The image is first captured as the 0^{th} frame through videocapture function in python.
- The captured image is pre-processed as discussed in Chapter 4. Once the image is processed the sign board is classisfied into one of the 1 to 43 labels which are mapped to sign board names.
- A sign board is recognised only when the probability is greater than ninety percent.
- The sign board name and its probability of true prediction is shown on the output image as shown in figure 5.19
- The class number as stored in the dataset is also shown in the output.
- An other instance of traffic sign board recognition is as shown in the figure 5.20.

Figure 5.19 : Sign board recognition 1 (frame 0)

Figure 5.20 : Sign board recognition 2 (frame 0)

5.8 Video Capture

- Each frame is captured through videocapture function in python.
- The frames per second rate is by default 60fps.
- The video input considered is of nineteen seconds duration with 30fps.
- The captured frame is pre-processed as discussed in Chapter 4. Once the image is processed the sign board is classisfied into one of the 1 to 43 labels which are mapped to sign board names.
- The sign board name and its probability of true prediction is shown on the output image as shown in figure 5.21
- The next frame is captured only after the sign board is outputted through speaker.
- Similarly frames as shown in figure 5.22 and figure 5.23 are also processed and traffic sign is recognised.

Figure 5.21 : Video Capture (Frame 0)

Figure 5.22 : Video Capture (Frame 150)

Figure 5.23 Video Capture (Frame 300)

Chapter 6

ADVANTAGES AND DISADVANTAGES

This Chapter deals with the various advantages and disadvantages of the traffic sign recognition system. The various advantages of TSR system are-

Reduction in number of Accidents

Nowadays it is difficult to find new models that do not include some form of driver assistance system. Manufacturers are investing in the development of safety equipment with the aim of reducing the number of accidents due to driver distraction as well as the severity of such accidents.

Automated Driving

The automated driving is coming to the market gradually so that drivers get used to this way of travelling by car and learn to trust it. It is based on driving assistance systems that increasingly facilitate semi-automated driving, such as those that help to change or stay in a lane, to brake in the event of an obstacle or pedestrian, or the adaptive cruise control that helps the driver maintain a certain speed when driving on the highway.

Driver Safety

The aim of this technology is to help improve driver safety, especially when the driver is tired or unable to pay proper attention to the signs.

The various disadvantages of TSR system are-

- The existence of a number of similar objects (either in colour or in shape) in the scene.
- The presence of obstacles in the scene which can partially or totally occlude the sign.
- The amount of information in the scene is vast and time is needed to analyse the scene and extract the desired information.
- The main disadvantage of visual recognition of traffic signs is associated with difficult conditions of image acquisition and hence problems with noise, blurring, scale and orientation changes should be solved

Chapter 7

FUTURE SCOPE

Traffic sign detection and recognition plays an important role in expert systems, such as traffic assistance driving systems and automatic driving systems. It instantly assists drivers or automatic driving systems in detecting and recognizing traffic signs effectively.

At Present, the Traffic Sign Recognition systems are available only in custom costly cars. In near future this system can be deployed into ‘Low Budget Auto Pilot Vehicles’. This system can be used in ADAS to only assist the driver about the traffic signs ahead or can be used directly to take appropriate action automatically

Chapter 8

CONCLUSION

As a conclusion, this proposal successfully uses various pre-processing techniques to process the input and recognise the traffic sign appropriately and then send a voice alert through speaker. It uses both standard GTSRB Dataset and a custom Dataset for the training of our convolutional neural network (LENET 5)

The Results of the last layer of cnn is used to classify the input into one of the traffic sign classes. It proposes a system that uses Python software for Traffic Sign Recognition system. This proposed system solves the potential problem of the unaffordable ADAS system. The various pre processing techniques used are the strength of this approach, which increase the efficiency.

Chapter 9

REFERENCES

- Krish Sukhani Radha Shankarmani Jay Shah Krushna Shah, “Traffic Sign Board Recognition and Voice Alert System using Convolutional Neural Network”, IEEE Conference, 2021.
- https://www.v7labs.com/blog/train-validation-test-set
- https://benchmark.ini.rub.de/gtsrb_dataset.html
- Y. Sun, P. Ge and D. Liu, "Traffic Sign Detection and Recognition Based on Convolutional Neural Network," *2019 Chinese Automation Congress (CAC)*, 2019, pp. 2851-2854, doi: 10.1109/CAC48633.2019.8997240.

- https://unsplash.com/s/photos/traffic-sign-20
- https://www.tensorflow.org/
- https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53
- https://opencv.org/
- https://www.python.org/downloads/
- benchmark.ini.rub.de