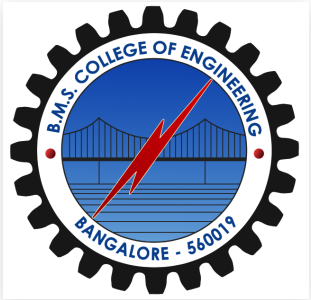
BMS COLLEGE OF ENGINEERING, BANGALORE – 560 019

(Autonomous institute, Affiliated to VTU)

Department of Information Science and Engineering



Multi Disciplinary Project - 20IS6PWMPR

TRAFFIC SIGNS DETECTION

2021 – 2022 – 6TH SEMESTER

Submitted by:

Mohan D - 1BM19IS092

Moksh Jayanth GR - 1BM19IS094

N Prabhu - 1BM19IS096

Under the guidance of

Prof. Shubha V Rao

BMS COLLEGE OF ENGINEERING, BANGALORE -19

(An autonomous institute, affiliated to VTU)

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

2021 – 2022 – EVEN SEMESTER

CERTIFICATE

Certified that **Mr. Mohan D** bearing USN **1BM19IS092** , **Mr.Moksh Jayanth GR** bearing USN **1BM19IS094** and **Mr.N Prabhu** bearing USN **1BM19IS096** of Sixth semester belonging to the Department of Information Science and Engineering had successfully completed AAT as a part of the course **Multi-Disciplinary Project [20IS6PWMPR]**

Faculty Incharge

Prof. Shubha V Rao

BMS COLLEGE OF ENGINEERING, BANGALORE -19

(An autonomous institute, affiliated to VTU)

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING
2021 – 2022 – EVEN SEMESTER

CONTENTS

Course: Multi Disciplinary Project
Course code: 20IS6PWMPR

ABSTRACT

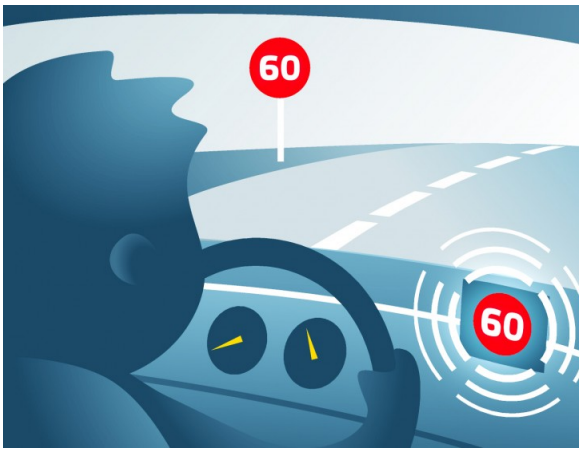
The web app's functionality is simple and minimal. It uses the concepts of Deep Learning to predict the Traffic signs and notify the same to the user by sending them a caution message.

This Traffic sign Detection web app also uses Concepts of Deep Learning And Computer Vision to Correctly Identify the proper signs. Training such models requires TeraBytes of labeled data.

Inspired by the following, we built a simple CNN Classification Model which detects and identifies the image of the given sign.

The model is capable of detecting over 40 classes of Traffic signs.

This technology could be applied in self-driving and automated cars, where the captured images are detected and the car could take the respective actions.

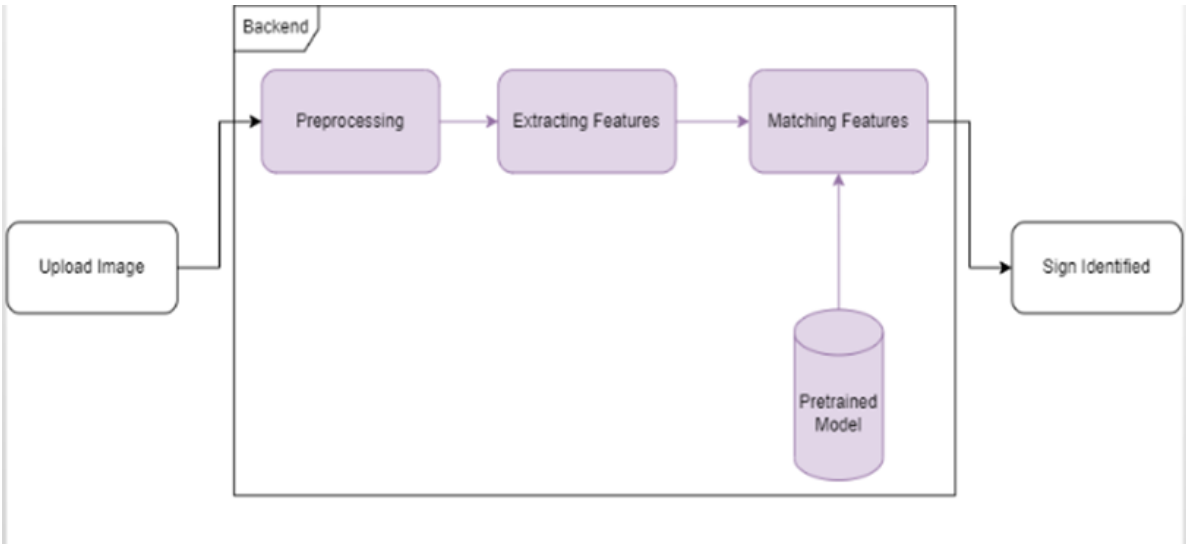


Technology Applications of the App

PROBLEM STATEMENT

To analyze a given image of a traffic sign and predict which class it belongs to. Employ different image recognition deep learning architecture for our test case/scenario and train traffic signs categorized into 40 different classes and to notify the user by sending a proper caution message to the registered mobile number.

HIGH LEVEL DESIGN OF THE WEB APPLICATION



INTRODUCTION

In this era of Artificial Intelligence, humans are becoming more dependent on technology. With the enhanced technology, multinational companies like Google, Tesla, Uber, Ford, Audi, Toyota, Mercedes-Benz, and many more are working on automating vehicles. They are trying to make more accurate autonomous or driverless vehicles. You all might know about self-driving cars, where the vehicle itself behaves like a driver and does not need any human guidance to run on the road. This is not wrong to think about the safety aspects—a chance of significant accidents from machines. But no machines are more accurate than humans. Researchers are running many algorithms to ensure 100% road safety and accuracy. One such algorithm is Traffic Sign Recognition.

Road traffic sign recognition

When you go on the road, you see various traffic signs like traffic signals, turn left or right, speed limits, no passing of heavy vehicles, no entry, children crossing, etc., that you need to follow for a safe drive. Likewise, autonomous vehicles also have to interpret these signs and make decisions to achieve accuracy. The methodology of recognizing which class a traffic sign belongs to is called Traffic signs classification.

In this project, we will build a model for the classification of traffic signs available in the image into many categories using a convolutional neural network(CNN) and Keras library.

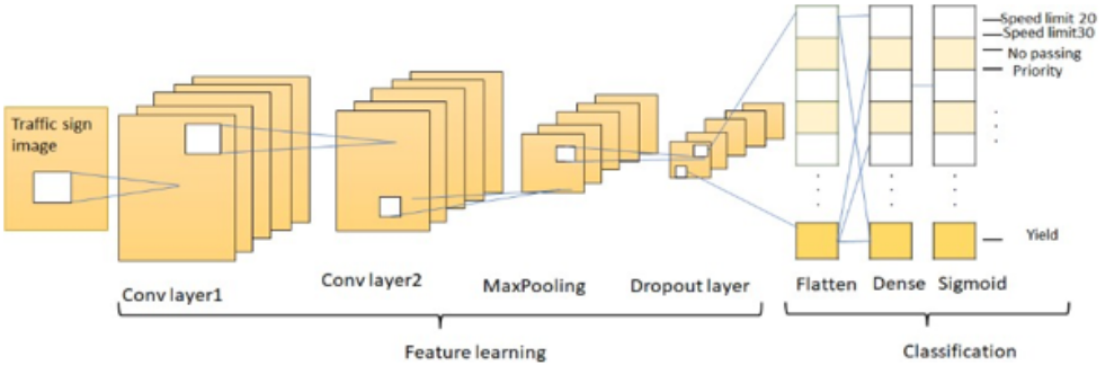


Fig.2: Model Architecture

About Dataset

The German Traffic Sign Benchmark is a multi-class, single-image classification challenge held at the International Joint Conference on Neural Networks (IJCNN) 2011. We cordially invite researchers from relevant fields to participate: The competition is designed to allow for participation without special domain knowledge. Our benchmark has the following properties:

- Single-image, multi-class classification problem
- More than 40 classes
- More than 50,000 images in total
- Large, lifelike database

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.

IMPLEMENTATION - MODEL CODE

Importing Dataset

```
import opendatasets as od

import pandas

od.download("https://www.kaggle.com/datasets/meowmeowmeowmeowmeow/gtsrb-german-traffic-sign")
```

Preprocessing the data

```
data = []

labels = []

classes = 40

curr_path = "/content/gtsrb-german-traffic-sign/Train"

#Retrieving the images and their labels

for i in range(classes):

   path = os.path.join(curr_path,str(i))

   images = os.listdir(path)

   for a in images:

          image = Image.open(path + '/'+ a)

          image = image.resize((30,30))

          image = np.array(image)

          data.append(image)

          labels.append(i)
```

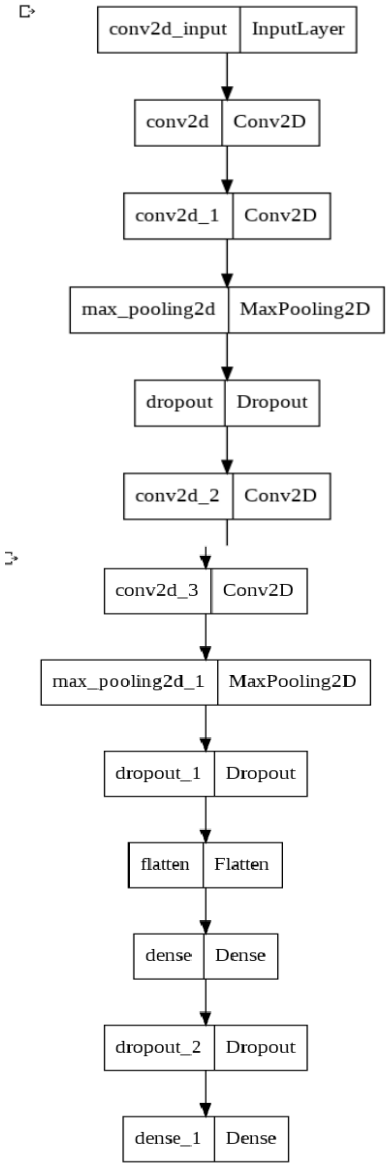
```
#Converting lists into numpy arrays

data = np.array(data)

labels = np.array(labels)

print(data.shape, labels.shape)
```

Building the model



```
model = Sequential()

model.add(Conv2D(filters=32, kernel_size=(5,5),
activation='relu', input_shape=X_train.shape[1:]))

model.add(Conv2D(filters=32, kernel_size=(5,5),
activation='relu'))

model.add(MaxPool2D(pool_size=(2, 2)))

model.add(Dropout(rate=0.25))

model.add(Conv2D(filters=64, kernel_size=(3, 3),
activation='relu'))

model.add(Conv2D(filters=64, kernel_size=(3, 3),
activation='relu'))

model.add(MaxPool2D(pool_size=(2, 2)))

model.add(Dropout(rate=0.25))

model.add(Flatten())

model.add(Dense(256, activation='relu'))

model.add(Dropout(rate=0.5))

model.add(Dense(40, activation='softmax'))
```

Compilation of the model

```
model.compile(loss='categorical_crossentropy', optimizer='adam',
metrics=['accuracy'])
```

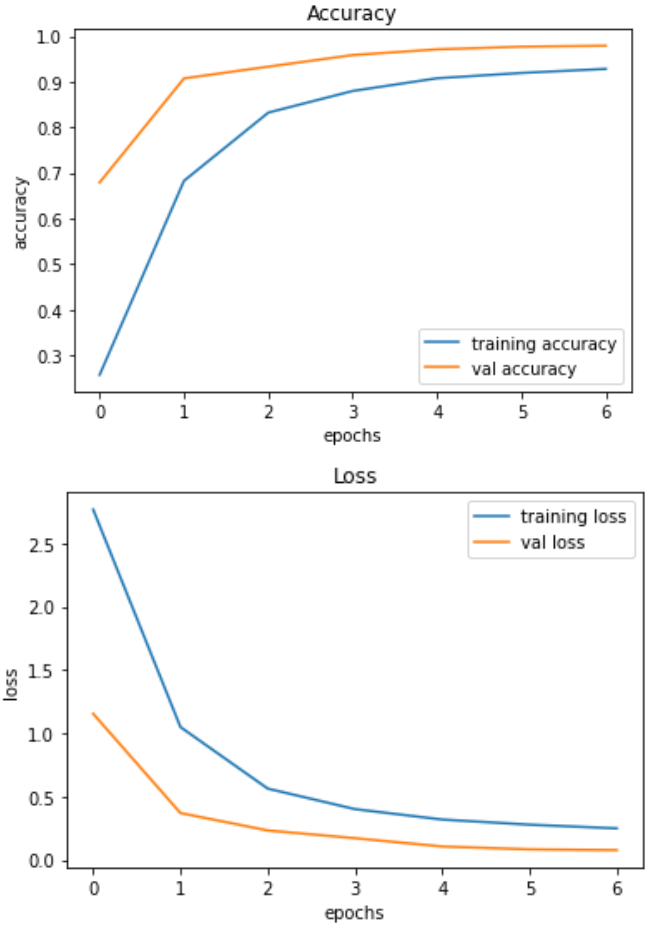
Training the model

```
epochs = 15
```

```
history = model.fit(X_train, y_train, batch_size=32,
epochs=epochs, validation_data=(X_test, y_test))

model.save("my_model.h5")
```

Model Performance



DEPLOYING OUR WEB APPLICATION

We Deployed the web application in 2 steps:

1. Extended our application functionality to provide text based SMS services to the user about the traffic sign boards.
2. Hosted the overall app using Gradio python library.

1. SMS FUNCTIONALITY OF THE WEB APPLICATION

- Twilio provides programmable communication tools for making and receiving phone calls, sending and receiving text messages, and performing other communication functions using its web service APIs.
- With just a few lines of code, the Python application can send SMS messages with Twilio Programmable Messaging.

STEPS TO CREATE A TWILIO PLATFORM

- Sign up for Twilio and get the SMS-enabled Twilio phone number
- Set up the development environment to send outbound messages
- Set up the first Twilio Messaging Service
- Send a SMS from that Messaging Service

CODE TO SEND SMS USING THE TWILIO API

```
!pip install twilio

from twilio.rest import Client

account_sid = "ACb103647d6f07ddc78a87b76a00b1e779"

   auth_token = "414988b2430c6f3d913878307f97f688"

   client = Client(account_sid, auth_token)
```

```
message = client.messages \
  .create(
      body= "\nSign Board Ahead!\n"+ targets[pred] + ". Drive Safe!",
      from_='+13853868323',
      to=PhoneNumber
  )
```

2. HOSTING THE APPLICATION

To Deploy the app we have used a python library known as **Gradio**.

Gradio is the fastest way to demo your machine learning model with a friendly web interface.

- **Gradio can be** installed with pip**.**
- **Gradio can be embedded in** Python notebooks **or presented as a** web page**.**
- **A Gradio interface can automatically generate a public link you can share with colleagues that lets them interact with the model.**

CODE SNIPPET FOR THE DEPLOYMENT USING GRADIO

```
!pip install -q gradio
import gradio as gr
def traffic_app(TrafficSignBoard, PhoneNumber):
   image = TrafficSignBoard
   image = np.expand_dims(image,axis=0)
   pred = model.predict(image)
   pred = np.argmax(pred)

   account_sid = "ACb103647d6f07ddc78a87b76a00b1e779"
   auth_token = "414988b2430c6f3d913878307f97f688"
   client = Client(account_sid, auth_token)
```

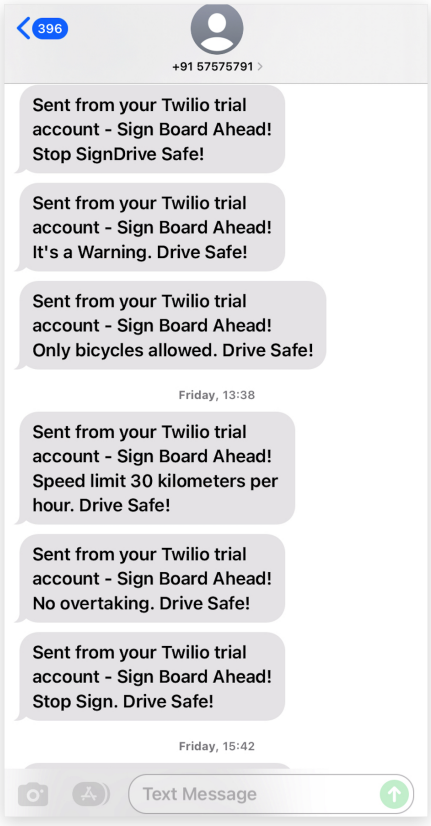
```
    message = client.messages \
      .create(
          body= "\nSign Board Ahead!\n"+ targets[pred] + ". Drive Safe!",
          from_='+13853868323',
          to=PhoneNumber
      )
    return targets[pred]


demo = gr.Interface(traffic_app, [gr.Image(shape=(30, 30)), "text"],
"text",
                    title="Traffic Sign Detection App",
                    description="Follow Traffic Rules. Drive Safe!")


demo.launch(share=True)
```

OUTPUT SNAPSHOTS

User Interface of the App



Instant SMS Delivery Service

CONCLUSION

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.

Available Traffic Sign Recognition uses a forward-facing camera to identify speed-limit and related traffic signs, such as those denoting school and construction zones. An additional safety and security feature, available on Wagoneer and standard on Grand Wagoneer, is Surround View Camera.

Our model gives a training accuracy of 94.8% and loss of 0.2093.

Model gives Validation accuracy of 98.41% and loss of 0.0582.

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.

FUTURE IMPLEMENTATION

- The Web Application takes the input manually from the user. But further the overall process i.e input, processing and output can be fully automated.
- SMS Reader or Voice messages can also be sent to the user to further adding to the app's functionality.
- The Functionality to automatically capture the image of the Sign board (i.e while the user is driving) within approx. of 208 meters between the board and the user and alert the user via voice message or an SMS immediately.
- The Functionality to mark the location of the sign board on google maps after its detection so that it may help the other users using the google maps.

RESEARCH PAPER AND OTHER REFERENCES

Research Papers -

- Research and Application of Traffic Sign Detection and Recognition Based on Deep Learning (2018) - https://ieeexplore.ieee.org/document/8410256
- A traffic sign detection algorithm based on deep convolutional neural network (2016) - https://ieeexplore.ieee.org/document/7888348

Other References -

- Kaggle For Sign Board Dataset - Kaggle: Your Machine Learning and Data Science Community
- Tensorflow - https://www.tensorflow.org
- Twilio - https://www.twilio.com
- Gradio As Deployment Library - Gradio