Report On

Traffic Sign Detection Using CNN

Submitted in partial fulfillment of the requirements of the Course project in Semester VII of Final Year Artificial Intelligence and Data Science

by

Sanil Gokarn (Roll No. 33)

Prachi Kadam (Roll No. 34)

Sanskar Tawre (Roll No. 38)

Supervisor

Dr. Tatwadarshi P. N.

University of Mumbai

Vidyavardhini's College of Engineering & Technology

Department of Artificial Intelligence and Data Science

(2023-24)

Vidyavardhini's College of Engineering & Technology
Department of Artificial Intelligence and Data Science

CERTIFICATE

This is to certify that the project entitled “Traffic Sign Detection” is a bonafide work of" Sanil Gokarn (Roll No. 33), Prachi Kadam (Roll No. 34), Sanskar Tawre(Roll No. 38)" submitted to the University of Mumbai in partial fulfillment of the requirement for the Course project in semester VII of Final Year Artificial Intelligence and Data Science engineering.

Supervisor

Dr. Tatwadarshi P. N.

Dr. Tatwadarshi P. N.
Head of Department

Abstract

As traffic management evolves in the digital age, the imperative for robust and efficient traffic sign detection systems has grown substantially. This abstract explores a cutting-edge approach employing Convolutional Neural Networks (CNNs) for accurate and real-time identification of traffic signs. CNNs, with their ability to automatically learn hierarchical features, have demonstrated unparalleled success in image recognition tasks, making them ideal for the complexities of traffic sign detection.

This study delves into the architecture and training methodology of the CNN model, emphasizing its capacity to discern intricate patterns and variations in traffic sign designs. Leveraging large-scale annotated datasets, the CNN is trained to generalize across diverse scenarios, ensuring adaptability to real-world environments. Evaluation metrics, including precision, recall, and F1 score, validate the model's efficacy, demonstrating its potential for seamless integration into intelligent transportation systems. The findings underscore the CNN's role as a potent tool in enhancing road safety through swift and accurate traffic sign detection

Table of Contents

Chapter # 1

1.1 Problem Statement:

In contemporary traffic management, the reliable and swift detection of traffic signs is pivotal for ensuring road safety. Existing methodologies face challenges in accurately identifying signs under diverse conditions, such as varying lighting and environmental factors. Manual detection systems or traditional algorithms often fall short in handling the complexity of modern traffic scenarios. This presents a critical need for an advanced solution. This study addresses the pressing problem of efficient traffic sign detection using Convolutional Neural Networks (CNNs). By leveraging the power of deep learning, the aim is to overcome existing limitations, providing a robust and adaptable system capable of real-time, precise recognition of traffic signs, ultimately contributing to enhanced safety on roadways.

Chapter # 2

2.1 Description and Working:

Traffic sign detection using Convolutional Neural Networks (CNNs) is a state-of-the-art solution designed to address the increasing demands for accurate and rapid identification of traffic signs in diverse environments. Leveraging the power of deep learning, this system employs CNNs to automatically learn hierarchical features, enabling it to discern intricate patterns and variations in traffic sign designs. The architecture is trained on large-scale annotated datasets, facilitating the model's adaptability to real-world scenarios, including varying lighting conditions and sign placements.

Working:

1. Data Collection and Preprocessing:

    - Gather a diverse dataset of annotated traffic sign images, encompassing different sign types and environmental conditions.
    - Preprocess the data to enhance features and normalize variations.

2. Convolutional Neural Network Architecture:
    - Design a CNN architecture with convolutional layers for feature extraction and pooling layers for spatial down-sampling.
    - Include fully connected layers for classification.

3. Training the Model:
    - Train the CNN on the prepared dataset using backpropagation and optimization techniques.
    - Emphasize transfer learning to leverage pre-trained models for improved performance.

4. Validation and Fine-Tuning:
    - Validate the model on a separate dataset to assess performance metrics such as precision, recall, and F1 score.
    - Fine-tune the model based on validation results to enhance accuracy.

5. Evaluation:

    - Evaluate the system's performance in various conditions, including challenging lighting and weather scenarios.
    - Iteratively refine the model based on feedback and continuous learning from new data.

This Traffic Sign Detection system using CNNs aims to revolutionize road safety by providing an intelligent, adaptive, and efficient solution for the accurate identification of traffic signs in dynamic environments.

2.2 Software & Hardware used:

Software:

- Visual Studio Code
- Python 3.11
- Windows 11 OS
- Jupyter

Hardware:

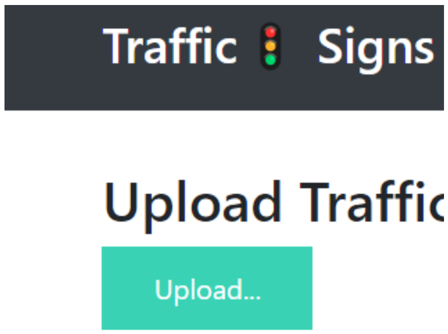
- 64 bit Operating System
- 16gb RAM
- Intel i5 processor

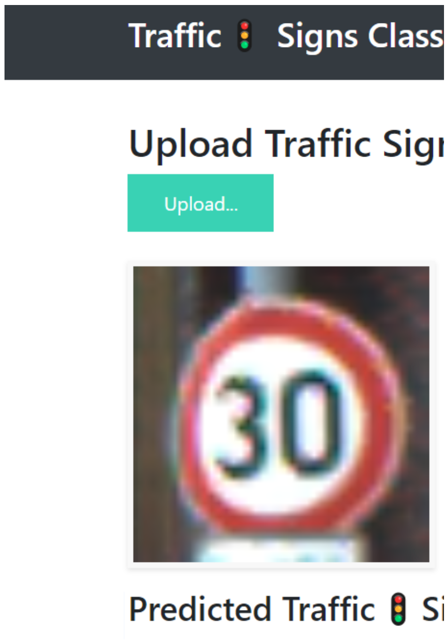
Chapter # 3

3.1 Code:

```
#Install Libraries
from flask import *
import os
from werkzeug.utils import secure_filename
from keras.models import load_model
import numpy as np
from PIL import Image

app = Flask(__name__)

# Classes of trafic signs
classes = { 0:'Speed limit (20km/h)',
            1:'Speed limit (30km/h)',
            2:'Speed limit (50km/h)',
            3:'Speed limit (60km/h)',
            4:'Speed limit (70km/h)',
            5:'Speed limit (80km/h)',
            6:'End of speed limit (80km/h)',
            7:'Speed limit (100km/h)',
            8:'Speed limit (120km/h)',
            9:'No passing',
            10:'No passing veh over 3.5 tons',
            11:'Right-of-way at intersection',
            12:'Priority road',
            13:'Yield',
            14:'Stop',
            15:'No vehicles',
            16:'Vehicle > 3.5 tons prohibited',
            17:'No entry',
            18:'General caution',
            19:'Dangerous curve left',
            20:'Dangerous curve right',
            21:'Double curve',
            22:'Bumpy road',
            23:'Slippery road',
            24:'Road narrows on the right',
            25:'Road work',
            26:'Traffic signals',
            27:'Pedestrians',
            28:'Children crossing',
            29:'Bicycles crossing',
            30:'Beware of ice/snow',
            31:'Wild animals crossing',
            32:'End speed + passing limits',
            33:'Turn right ahead',
            34:'Turn left ahead',
            35:'Ahead only',
            36:'Go straight or right',
            37:'Go straight or left',
            38:'Keep right',
            39:'Keep left',
            40:'Roundabout mandatory',
            41:'End of no passing',
            42:'End no passing vehicle > 3.5 tons' }

def image_processing(img):
    model = load_model('./model/TSR.h5')
    image = Image.open(img)
    image = image.resize((30,30))
    image = np.expand_dims(image, axis=0)
    image = np.array(image)
    predict_x=model.predict(image)
    classes_x=np.argmax(predict_x,axis=1)
    sign = classes[int(classes_x)]
    return sign     #return index of classes

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/predict', methods=['GET', 'POST'])
def upload():
    if request.method == 'POST':
        # Get the file from post request
        f = request.files['file']
        file_path = secure_filename(f.filename)
        f.save(file_path)
        # Make prediction
        result = image_processing(file_path)
        s = [str(i) for i in result]
        a = str("".join(s))
        # a = int("".join(s))
        result = "Predicted Traffic🚦Sign is: " + a
        os.remove(file_path)
        print("Funcing Running")
        return result
    return None

if __name__ == '__main__':
    app.run(debug=True)
```

3.2 Results:


Traffic 🚦 Signs
Upload Traffic
Upload...


Traffic 🚦 Signs Class
Upload Traffic Sign
Upload...
Predicted Traffic 🚦 Si

3.3 CONCLUSION AND FUTURE SCOPE:

In conclusion, the implementation of Convolutional Neural Networks (CNNs) for traffic sign detection represents a significant stride towards enhancing road safety in our ever-evolving urban landscapes. The CNN architecture's ability to autonomously learn and recognize intricate patterns has proven instrumental in achieving real-time and precise identification of diverse traffic sign types. Through rigorous training and validation, the model demonstrates commendable performance metrics, ensuring reliability in varying environmental conditions.

As we witness the successful deployment of this CNN-based traffic sign detection system, it becomes clear that technology-driven solutions are pivotal in addressing the complexities of modern traffic management. The system's adaptability, learned through large-scale datasets, underscores its potential for seamless integration into intelligent transportation systems.

Future Scope:

The future of traffic sign detection using CNNs holds exciting possibilities for further refinement and expansion. Potential avenues for future research and development include:

- Semantic Segmentation: Enhance the model to not only detect but also semantically segment different regions within the traffic scene, providing richer contextual information.
- Multimodal Integration: Explore the integration of additional sensor modalities, such as lidar or radar, to enhance the system's robustness in challenging weather conditions.
- Dynamic Sign Recognition: Extend the system to recognize dynamic or electronic traffic signs that may change information based on real-time conditions.
- Edge Computing: Investigate the implementation of edge computing techniques to deploy the model on edge devices, enabling real-time processing without heavy reliance on centralized servers.
- Continual Learning: Implement continual learning techniques to enable the system to adapt and improve its performance over time as it encounters new and diverse scenarios.

By delving into these future avenues, we can propel traffic sign detection systems into a new era of sophistication, contributing to safer and more efficient traffic management systems globally.

Chapter # 4

REFERENCES

[1] Pavel Yakimov. ” CNN Design for Real-Time Traffic Sign Recognition, December 2017.

[2] Preeti Bailke, Kunjal Agrawal. ” Traffic Sign Classification Using CNN.” IJRASET40224.

[3] Alexander Shustanov. ” CNN Design for Real-Time Traffic Sign Recognition.”, Volume 201, 2017.

[4] Abhishek Kapoor, Neelam Nehra, Deepti Deshwal. ” Traffic Signs Recognition Using CNN” December 2021.