Traffic Sign Detection

*Note: Sub-titles are not captured in Xplore and should not be used

Aniket Patel
Computer Science
Stevens Institute of Technology
Hoboken, NJ
apate211@stevens.edu

***Abstract*—Aim of the Project is to implement image classification using deep neural network.**

***Keywords—component, formatting, style, styling, insert** (key words)*

I. INTRODUCTION

Computer vision is a field of artificial intelligence (AI) that enables computers and systems to derive meaningful information from digital images, videos, and other visual inputs and take actions or make recommendations based on that information.Automobile manufacturers are increasingly reliant on computer vision and industrial robotics technologies to accomplish several levels of automation throughout the production process. The aim of the project is to detect the traffic sign using the webcam.

II. LIBRARIES REQUIRED

REQUIREMENTS

- PYTHON
- OPENCV
- TENSORFLOW
- SKLEARN
- PANDAS
- NUMPY
- MATPLOTLIB
- SEABORN

II. PREPROSESSING

First, normalize the image by mapping all the pixel vlues between 0 to 1. Apply some image processing on training, validation and test images like convert image into grayscale, histogram equalization and image normalization etc. I have performed data augmentation to generate more generic data. I have set different parameters for augmentation like width and height shift range is 10%, zoom range is 0.2 and rotation range is 10 degree. batch_size=20 means that 20 images are augmented at the same time.

III. DEFINING AND COMPILING THE MODEL

The model used for performing classification is deep neural network(CNN). A **Convolutional Neural Network (ConvNet/CNN)** is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics.

The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlap to cover the entire visual area.

For creating the model I have used a sequential model from keras library. Then I have to add convolutional layers in the network. In the first Conv2D layer I have used 16 filters and kernel size is (3,3). This kernel moves around the image and gets the image features. In MaxPooling2D I have used pool size (2,2). In the second Conv2D layer I have used 32 filters and kernel size is (3,3). In MaxPooling2D I have used pool size (2,2). In the third Conv2D layer I have used 64 filters and kernel size is (3,3). In MaxPooling2D I have used pool size (2,2).

After that I have added a Flatten layer with 512 nodes which is fully connected with the output layer. Basically the flatten layer is converting the data into a 1-D array for inputting it to the next layer. After that I have added a Dropout layer with 0.5 drop rate and then added an output layer. Output layer has only 43 nodes. For this model I have used Adam optimizer with 0.001 learning rate and categorical cross-entropy loss.

III. Training And Testing

We trained our model on 58510 images and then tested on 14629 images and then we also used early stopping form tenser flow library to prevent from overfitting.

IV. Deploying Model Using OpenCv

The main program reads video frame by frame. First of all apply preprocessing on the image and detect single or multiple signs in an frame after that crop the detected part where the traffic sign exists and convert into grayscale and resize them into (32x32) and then reshape into (32x32x1) and then pass image to model for making prediction and after prediction the name of the sign shown on the screen.

V. Conclusion

At last I would like to conclude that we were able to detect the traffic signals properly but before deploying it in the real world we need to train our model on more images to improve its accuracy and can also use different models for training and testing the data.

References

1. *https://www.kaggle.com/datasets/flo2607/traffic-signs-classification*
2. https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn
3. https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148
4. https://medium.com/the-andela-way/simple-operations-on-images-using-opencv-d37b26e6e3ab
5. Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
6. M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

IEEE conference templates contain guidance text for composing and formatting conference papers. Please ensure that all template text is removed from your conference paper prior to submission to the conference. Failure to remove template text from your paper **may result in your paper not being published.**