How to Cite:
Sakthivel, K., Raghul, B., & Raghul, E. (2022). Traffic sign recognition system using CNN and Keras. *International Journal of Health Sciences*, *6*(S4), 4986–4994.
https://doi.org/10.53730/ijhs.v6nS4.9226

Traffic sign recognition system using CNN and Keras

Dr. K. Sakthivel
Professor in Computer Science and Engineering, K.S.Rangasamy College of Technology, Tiruchengode-637 215, Namakkal District, Tamil Nadu, India
Corresponding author email: sakthivelk@ksrct.ac.in

Raghul B
Students of Computer Science and Engineering, K.S.Rangasamy College of Technology, Tiruchengode-637215, Namakkal District, Tamil Nadu, India
Email: raghulbathirasamy688088@gmail.com

Raghul E
Students of Computer Science and Engineering, K.S.Rangasamy College of Technology, Tiruchengode-637215, Namakkal District, Tamil Nadu, India
Email: raghuleswaran22@gmail.com

***Abstract*---**In this paper, we propose the best approach for a Traffic sign recognition system with a high accuracy rate and less computing time. This process is done with help of CNN and Keras. In fully automatic driving cars, it is difficult to recognize the traffic signs with less computing time and a high accuracy rate. So, to solve this problem, first, we are exploring the sample traffic sign dataset, next images are sorted and their labels are set into a list and those lists are converted into NumPy arrays for feeding to the model. Secondly, the CNN model is built to classify the images into their respective categories, this is the best approach for image classification. After building the model, the model is trained, validated, and tested using the test dataset. Finally, the graphical user interface is built for traffic sign recognition using Tkinter.

***Keywords*---**traffic sign recognition system, convolutional neural network, Keras, image processing, tensorflow.

Introduction

The recognition of traffic signs is an important study area in computer vision and an important component of Advanced Driver Assistance Systems (ADAS). It can be divided into two types of technologies: traffic-sign detection and recognition.

Manuscript submitted: 27 March 2022, Manuscript revised: 18 May 2022, Accepted for publication: 9 June 2022

The accuracy of detection will directly influence the ultimate results of identification. Traffic signs convey important signals about vehicle safety and display current traffic conditions, define road rights, prohibit and permit certain behaviors and driving routes, and display dangerous messages, among other things. They can also assist drivers in determining the state of the road and thus the best driving routes.

Humans are becoming more reliant on technology in this age of artificial intelligence. Multinational corporations such as Google, Tesla, Uber, Ford, Audi, Toyota, Mercedes-Benz, and others are attempting to automate vehicles using improved technology. They're attempting to develop more precise autonomous or driverless automobiles. You may have heard about self-driving cars, in which the vehicle acts as a driver and does not require human intervention to operate on the road. It is reasonable to consider the safety aspects—the possibility of serious machine mishaps.

However, no machine is more precise than humans. Many algorithms are being conducted by researchers to assure complete road safety and accuracy. When driving on the road, you will encounter numerous traffic signs such as traffic lights, turn left or right, speed restrictions, no passing of heavy vehicles, no entering, children crossing, and so on, which you must obey in order to drive safely. In order to reach accuracy, autonomous vehicles must also analyze these indicators and make decisions. Traffic signs classification is the process of determining which class a traffic sign belongs to.

We will use a convolutional neural network (CNN) and the Keras library to create a model for classifying traffic signs in an image into multiple categories in this Deep Learning project. More than 50,000 images of various traffic signs make up the image dataset (speed limit, crossing, traffic signals, etc.) The dataset contains around 43 different picture categorization classifications. The dataset classes range in size, with some having relatively few photographs and others having a large number. Because the dataset is only 314.36 MB in size, it takes very little time and space to download. It has two folders: train and test, with the train folder including classes and each category containing different graphics.

A convolutional neural network (CNN) is a form of artificial neural network that is specifically intended to process pixel input and is used in image recognition and processing. CNN's are image processing, artificial intelligence (AI) systems that employ deep learning to do both generative and descriptive tasks, frequently using machine vision that includes image and video recognition, recommender systems, and natural language processing (NLP). A neural network is a hardware and/or software system that mimics the behavior of neurons in the brain. Traditional neural networks aren't designed for image processing and must be fed images in smaller chunks. CNN's "neurons" are structured more like those in the frontal lobe, the area in humans and other animals responsible for processing visual inputs. Traditional neural networks' piecemeal image processing difficulty is avoided by arranging the layers of neurons in such a way that they span the whole visual field. Keras is a Google-developed high-level deep learning API for implementing neural networks. It is built in Python and is used to make neural network implementation simple. Multiple backend neural network computations are also supported.

Keras is simple to understand and use since it gives a high-level python frontend with the flexibility of different backends for computation. Keras is slower than other deep learning frameworks because of this, yet it is incredibly user-friendly. TensorFlow is the only one of these five frameworks that have accepted Keras as its official high-level API. Keras is a deep learning framework that is built on top of TensorFlow and has integrated modules for all neural network operations. Simultaneously, tensor computations, computation graphs, sessions, and other custom computations may be made using the TensorFlow Core API, giving you complete flexibility and control over your application and allowing you to quickly implement your ideas. Keras is TensorFlow's high-level API, a user-friendly, highly productive interface for tackling machine learning issues with a focus on current deep learning. It provides the necessary abstractions and building pieces for rapidly designing and shipping machine learning solutions.

Literature Review

[1] Many scholars have recently conducted studies on the topic of traffic sign recognition. The authors apply the notion of convolutional neural networks to recognize and classify the photos, according to the research article. The image is first pre-processed to highlight the most significant details. The Hough Transform is then used to detect and locate the areas. [2] The suggested system detects and recognizes traffic sign images in real time. A freshly developed database of 24 different traffic signs collected from random roadsides in Saudi Arabia is also a contribution to this work. The photographs were taken from various perspectives and included various characteristics and circumstances. A total of 2718 photos were collected to create the Saudi Arabian Traffic and Road Signs collection (SA-TRS-2018). To get the best recognition rates, the CNN architecture was employed in various settings. The proposed CNN architecture attained a precision of 100 percent in experiments, which is higher than that of similar previous studies.

[3] This study provides an intelligent transportation system design based on current requirements and technology to address the bottleneck issues that plague intelligent transportation research and to investigate the ITS and research focus's future prospects as new technologies become accessible. [4] This research presents a framework for detecting and categorizing various sorts of traffic signs in photos. Road sign detection and classification and recognition are the two key elements of the technique. Color-based segmentation is used in the first stage to determine whether or not a traffic sign is present. The sign will be highlighted, normalized in size, and categorized if it is present. For classification, a neural network is used. Stop, No Entry, Give Way, and Speed Limit signs are among the four types of traffic indicators used for evaluation. For training purposes, 300 sets of photos are employed, 75 sets for each kind. Testing is done with 200 photos. The detection rate is above 90%, and the recognition accuracy is over 88 percent, according to the test findings.

[5] In this study, a deep learning-based system for recognizing road traffic signs is created, which has a lot of potential in the development of ADAS and autonomous

vehicles. The system architecture is intended to extract key elements from photographs of traffic signs in order to categorize them. To conduct the recognition, the presented method employs a modified LeNet-5 network to extract a deep representation of traffic signs. It is made up of a Convolutional Neural Network (CNN) with all convolutional layers' output connected to a Multilayer Perceptron (MLP). The training uses the German Traffic Sign Dataset and produces good results in terms of traffic sign recognition.

Existing System

Traffic signs carry much information necessary for successful driving - they describe up-to-date traffic situations, define right-of-way, prohibit or permit certain actions or directions, warn about risk factors, etc. Road signs also help drivers with routing the vehicle by identifying the road-sign by using computer vision. The road conditions in the actual scene are very complicated so it was really very hard for the researchers to make the system efficient. The existing system has been detected and categorized using standard computer vision methods, but it was taking a long time.

Proposed System

In this, a traffic sign detection and identification method on account of the image processing is proposed, which is combined with a convolutional neural network (CNN) and Keras to sort traffic signs. CNN may be utilized to perform a variety of computer vision tasks due to its high recognition rate. TensorFlow is used to implement CNN and Keras. In this, we are able to identify the symbol with more accuracy rate and with lesser time.

Methodology

Each of the 43 folders in our dataset folder represents a different class. The folder's size ranges from 0 to 42. We iterate over all of the classes using the OS module, appending images and their labels to the data and labels list. To open picture content into an array, the PIL library is utilized. Finally, we organized all of the images and labels into lists (data and labels). To feed the model, we must turn the list into NumPy arrays. The data has the shape (39209, 30, 30, 3), indicating that there are 39,209 images of 3030 pixels and that the last three indicate that the data comprises colored images (RGB value). The train_test_split() method in the sklearn package is used to split training and testing data. To transform the labels in y_train and t_test into one-hot encoding, we utilize the to_categorical method from the Keras.utils package.

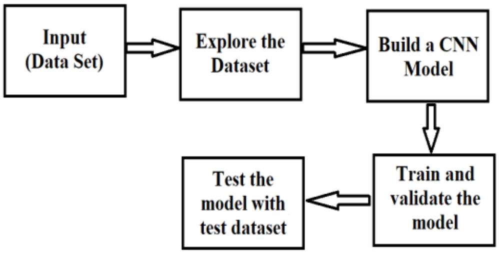

Figure 1 – Dataset block diagram

We'll use a CNN model to classify the photos into their appropriate groups (Convolutional Neural Network). For picture categorization, CNN is the best option.

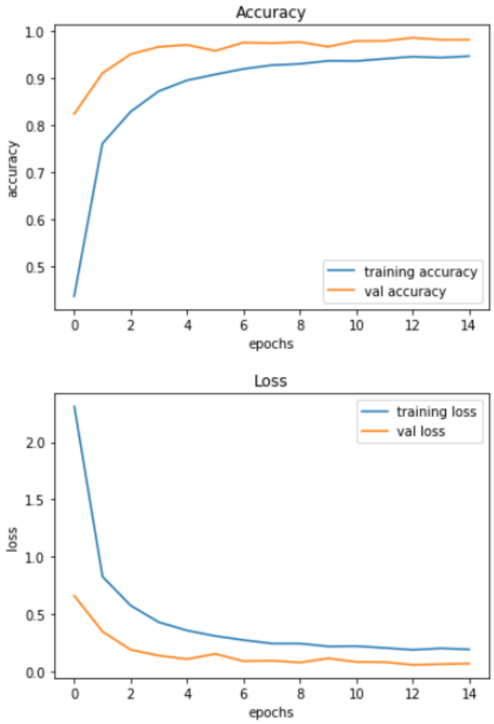

Figure 2 – Accuracy and Loss graph

Because we have several classes to categorize, we compile the model with Adam optimizer, which performs well, and the loss is "categorical_crossentropy." After constructing the model architecture, we use model.fit to train the model (). I tried batch sizes of 32 and 64. With 64 batches, our model fared better. The accuracy was stable after 15 epochs. On the training dataset, our model had a 95% accuracy rate. We visualize the graph for accuracy and loss using matplotlib.

The details connected to the image path and their appropriate class labels are contained in a test.csv file in our dataset. Using pandas, we extract the image path and labels. Then, in order to forecast the model, we must scale our photographs to 3030 pixels and create a NumPy array with all of the image data. We used the accuracy score from sklearn.metrics to see how our model predicted the real labels. In this model, we were able to attain a 95% accuracy rate. Now we are going to build a graphical user interface for our traffic signs classifier with Tkinter. Tkinter is a GUI toolkit in the standard python library.

We started by loading the trained model 'traffic classifier.h5' using Keras. Then we create the user interface for uploading the image, with a classify button that launches the classify() code function. The classify() function transforms an image into a shape dimension (1, 30, 30, 3). This is because we must provide the same dimension that we used to develop the model to forecast the traffic sign. Then we predict the class, and the model.predict_classes(image) returns us a number between (0-42) which represents the class it belongs to. We look up information about the class in the dictionary. Here's the code for the gui.py file. Importing necessary packages, retrieving the images and their tables, converting lists into Numpy arrays, splitting training and testing dataset, converting the labels, building the model, complication of the model, plotting graphs for accuracy, testing accuracy on the test dataset, accuracy with the test data.

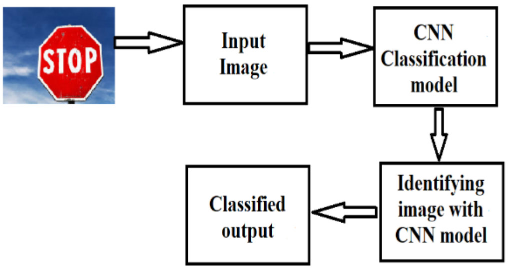

Figure 3 – Proposed block diagram

All the necessary details which are essential for the python to run the program are imported. The images which are already available in the dataset are retrieved with their tables. The list which are numerical values of the image is being converted into NumPy arrays. Dataset is split into training and testing. The labels are converted into one hot encoding. Building of the model is done with the use of the data set which is already available and we got in the dataset splitting process. The complication of the models is done. For checking the accuracy graphs are plotted with the help of the result we got from the solution. The accuracy of the result with the help of the trained dataset which we have already split into testing and training. The final accuracy of the result is given with the help of the Sklearn metric.

Result and Discussion

With the help of the given algorithms and others python libraries, we have successfully achieved the result of identifying the traffic signs in different environments. So, with the help of the given dataset from Kaggle, we achieved training sets based on the dataset. With the use of the dataset .h5 classifier file is created. With that classifier file, we are able to get the results quickly. The system is capable of classifying the input images with a 95% accuracy rate.

Figure 4 – Sample Output

The images are captured from the sensors in the cars. That image is given as input to the system. With help of a convolutional neural network (CNN) and Keras, images are classified. Dataset is used for training our algorithm, there are different sets of data present in the dataset. There are data of different climatic conditions and different tampered sign boards. During those difficult situations our algorithm with help of CNN and Keras, we are able to achieve a high accuracy rate for classifying the traffic signs.

Conclusion

In this paper, a traffic sign recognition method on account of deep learning with help of a convolutional neural network (CNN) and Keras is proposed, which mainly aims at different traffic signs. By using image pre-processing, the dataset from Kaggle, traffic sign detection, recognition, and classification, this method can effectively detect and identify traffic signs. With help of these results, we can identify the traffic sign. It helps the user in two ways, while the user is in manual mode it displays the result on the dashboard screen and while the car driver is set to automatic it helps the car to drive safely by identifying the traffic signs. The test result displays that the accuracy of this method is very high.

References

1. Traffic Sign Detection and Recognition Based on Convolutional Neural Network, Yingsun; pingshuge; dequan liu, 2019, IEEE
2. Research and application of traffic sign detection and recognition based on deep learning, Canyong wang,2018 IEEE
3. Traffic sign detection and classification using color feature and neural network, Md. Abdul alim sheikh; alok kole; Tanmoy maity 2018 IEE
4. Autonomous Traffic Sign(ASTR) Detection and Recognition using Deep CNN, Danyah A.Alghmgham , ghazanfar latif , jaafar alghazo , loay alzubaidi ,2019 SCIENCEDIRECT
5. Understanding of a convolutional Neural network, saad albawi; tareq abed mohammed; saad al-zawi 2018 IEEE
6. Intelligent transportation system(its): Concept, challenge and opportunity, yangxin lin; ping wang; meng ma,2017 IEEE.
7. Data classification with deep learning using Tensorflow, Galip Aydın; Fatih Ertam, 2017, IEEE.
8. TFCheck: A TensorFlow Library for Detecting Training Issues in Neural Network Programs, Foutse Khomh; Houssem Ben Braiek, 2019, IEEE.
9. Accuracy Testing of Data Classification using Tensor Flow a Python Framework in ANN Designing, Neeraj Chauhan; Rakesh Kr. Dwivedi Ashutosh Kr. Bhatt; Rajendra Belwal, 2019, IEEE.
10. Performance of deep learning computation with TensorFlow software library in GPU-capable multi-core computing platforms, Young Jong Mo; JoongheonKim; Jong-KookKim; Aziz Mohaisen; Woojoo Lee, 2017, IEEE.
11. A Survey of Deep Learning: Platforms, Applications and Emerging Research Trends, Wei Yu, 2018, IEEE.
12. Object recognition in images using 2convolutional neural network, Duth P Sudharshan; Swathi Raj; 2018, IEEE.
13. Machine Learning and Deep Learning frameworks and libraries for large-scale data mining: a survey, Giang Nguyen, Stefan Dlugolinsky, Martin Bobák, Viet Tran, Álvaro López García, Ignacio Heredia, Peter Malík & Ladislav Hluchý; 2019, Springer.
14. Image Recognition with Deep Learning, Md Tohidul Islam; B.M. Nafiz Karim Siddique; Taskeed Jabid, Sagidur Rahman, 2018 IEEE.
15. Region based image retrieval using k-means and hierarchical clustering algorithms, R Abinaya, K Sakthivel, I Nivetha, RA Kumar.

16. Rinartha, K., & Suryasa, W. (2017). Comparative study for better result on query suggestion of article searching with MySQL pattern matching and Jaccard similarity. In *2017 5th International Conference on Cyber and IT Service Management (CITSM)* (pp. 1-4). IEEE.
17. Rinartha, K., Suryasa, W., & Kartika, L. G. S. (2018). Comparative Analysis of String Similarity on Dynamic Query Suggestions. In *2018 Electrical Power, Electronics, Communications, Controls and Informatics Seminar (EECCIS)* (pp. 399-404). IEEE.
18. Susilo, C. B., Jayanto, I., & Kusumawaty, I. (2021). Understanding digital technology trends in healthcare and preventive strategy. *International Journal of Health & Medical Sciences*, *4*(3), 347-354. https://doi.org/10.31295/ijhms.v4n3.1769