International Journal of Multidisciplinary Engineering in Current Research - IJMEC
Volume 8, Issue 3, March-2023, http://ijmec.com/, ISSN: 2456-4265

CAR TRAFFIC SIGN RECOGNIZER USING CONVOLUTIONAL NEURAL NETWORK

Ashhar Ali Fahad [1], Faraz Syed Ahmed[2], Mohammed Abdul Aziz[3], Mr. Amer Noor Khan[4]
[1,2,3] B.E.Student, Department of IT, Lords Institute of Engineering and Technology, Hyderabad
[4]Associate Professor, Department of IT, Lords Institute of Engineering and Technology, Hyderabad
amernoorkhan@lords.ac.in

Abstract— The roadside traffic signs we see every day are crucial to our safety on the road. Drivers and passengers alike rely on them for vital data. As a result, students must learn to control their driving habits and always adhere to the rules of the road as they are now enforced, all without endangering the safety of other motorists or pedestrians. To help drivers be aware of and comply with traffic regulations, traffic sign classification systems are used to identify and categorize signs. The suggested method addresses several of the issues with previously utilized categorization systems, such as inaccurate predictions, expensive hardware, and frequent maintenance. The suggested method uses a convolutional neural network to create a traffic indicators categorization algorithm. In addition, it has a function that can identify a traffic signal from a webcam. This will allow the motorist to keep the traffic sign in plain sight without having to constantly look up from the screen.

Keywords: Car Traffic Automotive Signals, Machine Learning Models, Neural network.

I. INTRODUCTION

The two main steps of traffic sign acknowledgment (TSR) frameworks are recognition and arrangement, while some TSR frameworks also intend on including a third phase—detection and classification—to better handle video streams.When it comes to resolving issues with picture classification and identification, deep learning emerges as the most effective subset of machine learning. There is no denying deep learning's success in the realm of autonomous vehicles. Differential neural networks (DNNs) have been put to use in a wide variety of domains, such as scene semantic division [2], traffic signal identification [3], crosswalk arrangement [4], [5], traffic sign position [6], walker investigation [7], vehicle heading course evaluation [8], and many more. Identifying road signs is at the heart of the planned study. Customers on the street may learn a great deal by just looking at the traffic coming in. The creation of a trustworthy automated traffic sign detecting system is crucial. While Deep Learning has many benefits, it's not without certain risks.Training a deep neural network demands a big, evenly distributed data set, which in turn necessitates a huge computing investment. Given that certain traffic signs are quite unusual and seldom seen in everyday driving, accomplishing this goal will need a dataset that includes almost all information about traffic. Dentifying traffic signs may be done using one of three main methods: (i) "AdaBoost based detection," (ii) "Support Vector Machine," and (iii) "Neural Network" based detection. These photos are from authentic settings on German streets and are utilized in the German Traffic Sign Detection Benchmark (GTSDB) [9].

1.1 Input Design

The input layout connects the user to the data system. Data preparation entails the development of specifications and procedures for transforming raw transaction data into a format that can be processed by a computer. This can be done visually, by having the computer read data from a paper document, or manually, by having people key the information into the system. reducing the needed quantity of input, reducing the mistakes, preventing latency, eliminating superfluous stages, and making the process simple are all priorities in input design. The input was created to keep users' personal information private while still being secure and simple to use. The following were taken into account by Input Design: What data should be given as input?

- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

1.2 Objectives

Input Design, first and foremost, is the procedure through which a user-centric description of the input is translated into a digital system. Errors in data entry may be avoided with this layout, and the right information can be obtained by management from the computerized system.

To manage a big amount of data, we design intuitive displays for data input. The purpose of input design is to reduce the potential for human mistake during data entering. All data manipulations are conveniently accessible on the data entering screen. It also has a place to watch old records.

Third, it validates the information as it is input. With the aid of displays, information may be input. So that the user is not caught off guard, appropriate messages are sent at the right time. As a result, while designing inputs, simplicity and clarity are prioritized.

1.3 Output Design

A good output is one that provides useful information to the intended audience in an understandable way. Outputs are the means through which a system conveys its processed results to its users and other systems. Output design is where decisions about how data will be made available in a timely manner and in hard copy form are made. This is the most relevant and useful data for the user. The system's ability to aid user decision-making is enhanced by efficient and intelligent output design.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2. Select methods for presenting information.

3. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the future.
- Signal important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action.

II. LITERATURE SURVEY

2.1 Machine Vision Based Traffic Sign Detection Methods: Review, Analyses and Perspectives

Authors: C. Liu, S. Li, F. Chang and Y. Wang,

Many modern driver-assistance and autonomous driving systems rely on traffic sign recognition (TSR) to function properly. Traffic sign detection (TSD) is an essential initial step in TSR, but it is a difficult task because to factors such as the variety of traffic signs, their tiny sizes, the complexity of driving situations, and occlusions. Over the last several years, many TSD algorithms have been developed using machine vision and pattern recognition. An extensive analysis of previous works on TSD is provided in this publication. Specifically, we classify the evaluated detection techniques into the following five groups: color-based, shape-based, color-and-shape-based, machine-learning-based, and LIDAR-based. In order to better comprehend and summarize the mechanics behind the various approaches, the methods within each category are further broken down into subcategories. When evaluating the evaluated techniques, we reimplemented a subset of the methods that lacked comparisons using publicly available datasets. We give experimental analysis and comparisons between the reported performance and our reimplemented approaches. To further the advancement of the TSD, suggestions and proposals for further study are provided.

2.2 Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation

Authors: L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff and H. Adam

Spatial pyramid pooling module or encode-decoder structure are used in deep neural networks for semantic segmentation task. The former networks are able to encode multi-scale contextual information by probing the incoming features with filters or pooling operations at multiple rates and multiple effective fields-of-view, while the latter networks can capture sharper object boundaries by gradually recovering the spatial information. In this work, we propose to combine the advantages from both methods. Specifically, our proposed model, DeepLabv3+, extends DeepLabv3 by adding a simple yet effective decoder module to refine the segmentation results especially along object boundaries. We further explore the Xception model and apply the depthwise separable convolution to both Atrous Spatial Pyramid Pooling and decoder modules, resulting in a faster and stronger encoder-decoder network. We 4 demonstrate the effectiveness of the proposed model on PASCAL VOC 2012 and Cityscapes datasets, achieving the test set performance of 89% and 82.1% without any post-processing.

III. SYSTEM ANALYSIS

3.1 Existing System

Some traffic sign recognition (TSR) frameworks include a third step intended between detection and Classification for handling video sequences, whereas others just have the two basic stages of recognition and organization. When it

comes to resolving issues with picture classification and identification, deep learning emerges as the most effective subset of machine learning. There is no denying deep learning's success in the realm of autonomous vehicles. The use of DNNs is widespread, and some examples include: scene semantic division; traffic signal identification; crosswalk arrangement; traffic sign position; walker investigation; vehicle heading course evaluation; and many more.

3.1.1 Disadvantages of Existing System

- Results are not up to the mark
- Image based results are not particular and not prominent.

3.1.2 Algorithm Used:

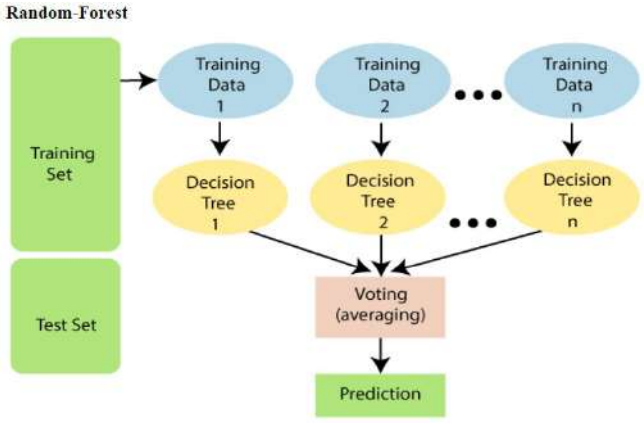

Fig 3.1 Random-forest Flow diagram

The supervised learning approach that Random Forest is a part of is one of the most widely used in machine learning. It is applicable to both ML Classification and ML Regression issues. It relies on ensemble learning, which involves merging several classifiers to address a difficult issue and boost the model's accuracy.

As its name indicates, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead than relying on just one decision tree, the random forest averages the forecasts made by each individual tree to arrive at an overall prediction.

Overfitting is avoided and accuracy is increased as the forest size increases.

3.2 Proposed System

The proposed project work focuses around traffic sign detection. A traffic in alone can convey a whole lot of information to the street clients. It is significantly important to develop a reliable automatic traffic sign detection system. Apart from the advantages of Deep learning there are few concerns related to it as .Deep learning requires expensive annotation, a large balanced data set which in result requires high computational cost in training. To achieve this task, a dataset that contains almost all the information about traffic is required as there are some traffic signs, which are rarely seen and uncommon under normal driving scenarios.

3.2.1 Advantages of Proposed System

• CNNs end up being unrivaled at Image order, Video Analysis, Natural Language Processing

• German Traffic Sign Detection Benchmark (GTSDB) is used as these images are taken from real scenarios of German street.

Algorithm: AdaBoost based detection,Support Vector Machine (SVM), and ,Neural Network (NN)

3.2.2 Algorithms Used:

SVM:

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n- dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane. SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine.

3.3 Process Model Used with Justification:

Fig 3.2 The software development life cycle

The Software Development Life Cycle is a standard which is used by software industry to develop good software.

Stages in SDLC:

- Requirement Gathering

- Analysis
- Designing
- Coding
- Testing
- Maintenance

3.3.1 Requirements Gathering stage

The requirements gathering process takes as its input the goals identified in the high-level requirements section of the project plan. Each goal will be refined into a set of one or more requirements. These requirements define the major functions of the intended application, define operational data areas and reference data areas, and define the initial data entities. Major functions include critical processes to be managed, as well as mission critical inputs, outputs and reports. A user class hierarchy is developed and associated with these major functions, data areas, and data entities. Each of these definitions is termed a Requirement. Requirements are identified by unique requirement identifiers and, at minimum, contain a requirement title and textual description.

These requirements are fully described in the primary deliverables for this stage: the Requirements Document and the Requirements Traceability Matrix (RTM). The requirements document contains complete descriptions of each requirement, including diagrams and references to external documents as necessary. Note that detailed listings of database tables and fields are not included in the requirements document.

The title of each requirement is also placed into the first version of the RTM, along with the title of each goal from the project plan. The purpose of the RTM is to show that the product components developed during each stage of the software development lifecycle are formally connected to the components developed in prior stages.

In the requirements stage, the RTM consists of a list of high-level requirements, or goals, by title, with a listing of associated requirements for each goal, listed by requirement title. In this hierarchical listing, the RTM shows that each requirement developed during this stage is formally linked to a specific product goal. In this format, each requirement can be traced to a specific product goal, hence the term requirements traceability.

The outputs of the requirements definition stage include the requirements document, the RTM, and an updated project plan.

Feasibility study is all about identification of problems in a project.

No. of staff required to handle a project is represented as Team Formation, in this case only modules are individual tasks will be assigned to employees who are working for that project.

Project Specifications are all about representing of various possible inputs submitting to the server and corresponding outputs along with reports maintained by administrator.

3.2 Analysis Stage

The planning stage establishes a bird's eye view of the intended software product, and uses this to establish the basic project structure, evaluate feasibility and risks associated with the project, and describe appropriate management and technical approaches.

3.3.3 Designing Stage

The design stage takes as its initial input the requirements identified in the approved requirements document. For each requirement, a set of one or more design elements will be produced as a result of interviews, workshops, and/or prototype efforts. Design elements describe the desired software features in detail, and generally include functional hierarchy diagrams, screen layout diagrams, tables of business rules, business process diagrams, pseudo code, and a complete entity-relationship diagram with a full data dictionary. These design elements are intended to describe the software in sufficient detail that skilled programmers may develop the software with minimal additional input.

When the design document is finalized and accepted, the RTM is updated to show that each design element is formally associated with a specific requirement. The outputs of the design stage are the design document, an updated RTM, and an updated project plan.

3.3.4 Development Stage

The development stage takes as its primary input the design elements described in the approved design document. For each design element, a set of one or more software artifacts will be produced. Software artifacts include but are not limited to menus, dialogs, data management forms, data reporting formats, and specialized procedures and functions. Appropriate test cases will be developed for each set of functionally related software artifacts, and an online help system will be developed to guide users in their interactions with the software.

The RTM will be updated to show that each developed artifact is linked to a specific design element, and that each developed artifact has one or more corresponding test case items. At this point, the RTM is in its final configuration. The outputs of the development stage include a fully functional set of software that satisfies the requirements and design elements previously documented, an online help system that describes the operation of the software, an implementation map that identifies the primary code entry points for all major system functions, a test plan that describes the test cases to be used to validate the correctness and completeness of the software, an updated RTM, and an updated project plan.

IV. SYSTEM DESIGN:

4.1 System Architecture:

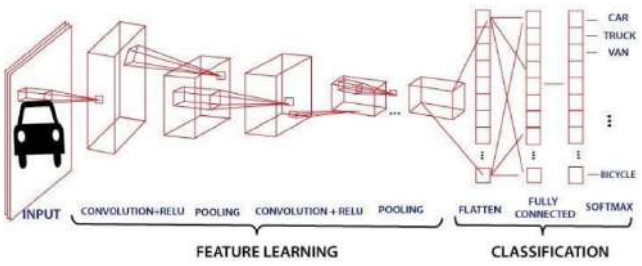

Fig 4.1 System Architecture

4.2 Data Flow Diagram

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modelling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

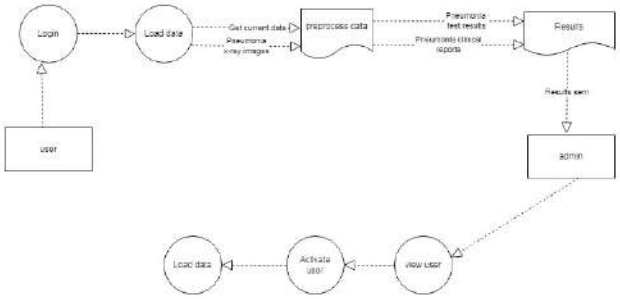

Fig 4.2 Data Flow Diagram

V. IMPLEMENTATION:

5.1 PYTHON

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. An interpreted language, Python has a design philosophy that emphasizes code readability (notably using whitespace indentation to delimit code blocks rather than curly brackets or keywords), and a syntax that allows programmers to express concepts in fewer lines of code than might be used in languages such as C++or Java. It provides constructs that enable clear programming on both small and large scales. Python interpreters are available for many operating systems. CPython, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of its variant implementations. CPython is managed by the non-profit Python Software Foundation. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

Interactive Mode Programming

Invoking the interpreter without passing a script file as a parameter brings up the following prompt −

$ python

Python 2.4.3 (#1, Nov 11 2010, 13:34:43)

[GCC 4.1.2 20080704 (Red Hat 4.1.2-48)] on linux2

Type "help", "copyright", "credits" or "license" for more information.

>>>

Type the following text at the Python prompt and press the Enter −

>>> print "Hello, Python!"

If you are running new version of Python, then you would need to use print statement with parenthesis as in print ("Hello, Python!");. However in Python version 2.4.3, this produces the following result −

Hello, Python!

Script Mode Programming

Invoking the interpreter with a script parameter begins execution of the script and continues until the script is finished. When the script is finished, the interpreter is no longer active.

Let us write a simple Python program in a script. Python files have extension .py. Type the following source code in a test.py file −

Live Demo

print "Hello, Python!"

We assume that you have Python interpreter set in PATH variable. Now, try to run this program as follows −

$ python test.py

This produces the following result −

Hello, Python!

Let us try another way to execute a Python script. Here is the modified test.py file −

Live Demo

#!/usr/bin/python

print "Hello, Python!"

We assume that you have Python interpreter available in /usr/bin directory. Now, try to run this program as follows −

$ chmod +x test.py # This is to make file executable

$./test.py

This produces the following result −

Hello, Python!

Python Identifiers :

A Python identifier is a name used to identify a variable, function, class, module or other object. An identifier starts with a letter A to Z or a to z or an underscore (_) followed by zero or more letters, underscores and digits (0 to 9).

VI. CONCLUSION & FUTURE ENHANCEMENTS

In this project we proposed a Traffic Sign recognition system using Convolutional Neural Networks(CNN) with State of art results having 96.19% accuracy on the German dataset(GTSDB) used. The CNN framework integrates multiple levels of convolution feature and multiple levels of contextual information. At the detection stage, the

region proposals are generated from the fused feature map with sufficient information. ADAM optimizer was used to decrease the computational and training cost which helped in achieving the given accuracy.

Future Enhancements:

In order to build a good and reliable classification model, it is very important to gather as much data as possible. Further research steps will include experimenting with various pre-processing and CNN configurations, data augmentation techniques, as well as using additional datasets with additional data labels. In the future we would like to increase the accuracy and efficiency of our project by visualizing remaining errors and using further optimization techniques.

REFERENCES

[1] C. Liu, S. Li, F. Chang and Y. Wang, "Machine Vision Based Traffic Sign Detection Methods: Review, Analyses and Perspectives," in IEEE Access, vol. 7, pp. 86578-86596, 2019, doi: 10.1109/ACCESS.2019.2924947.

[2] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff and H. Adam, "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation", European Conference on Computer Vision (ECCV), 2018.

[3] K. Behrendt, L. Novak and R. Botros, "A deep learning approach to traffic lights: Detection tracking and classification"International Conference on Robotics and Automation (ICRA), pp. 1370- 1377, 2017. [4] R. F. Berriel, F. S. Rossi, A. F. de Souza and T. Oliveira-Santos, "Automatic Large-Scale Data Acquisition via Crowdsourcing for Crosswalk Classification: A Deep Learning Approach", Computers & Graphics, vol. 68, pp. 32-42, 2017

[5] R. F. Berriel, A. T. Lopes, A. F. de Souza and T. OliveiraSantos, "Deep Learning-Based Large-Scale Automatic Satellite Crosswalk Classification", Geoscience and Remote Sensing Letters, vol. 14, no. 9, pp. 1513-1517, 2017.

[6] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li and S. Hu, "Traffic-Sign Detection and Classification in the Wild", Conference on Computer Vision and Pattern Recognition (CVPR), 2016

[7] R. Guidolini, L. G. Scart, L. F. Jesus, V. B. Cardoso, C. Badue and T. Oliveira-Santos, "Handling Pedestrians in Crosswalks Using Deep Neural Networks in the IARA Autonomous Car", International Joint Conference on Neural Networks (IJCNN), 2018.

[8] R. F. Berriel, L. T. Torres, V. B. Cardoso, R. Guidolini, C. Badue, A. F. D. Souza, et al., "Heading direction estimation using deep learning with automatic large-scale data acquisition", International Joint Conference on Neural Networks IJCNN, 2018.

[9] S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing and C. Igel, "Detection of Traffic Signs in Real-World Images: The German Traffic Sign Detection Benchmark", International Joint Conference on Neural Networks (IJCNN), 2013.

[10] Hatolkar, Yuga & Agrawal, Poorva & Patil,Seema. (2018). A Survey on Road Traffic Sign Recognition System using Convolution Neural Network. International Journal of Current Engineering and Technology. 8. 10.14741/ijcet/v.8.1.21.