

Traffic Sign Recognition using YOLO Architecture for Autonomous Driverless Vehicles

R.Sathish Kumar, S.S.Siva Sakthi Balan, P.siva Subramanian, S.Abirami

*SSM Institute of Engineering And Technology, Department of Electronics and Communication Engineering
Assistant Professor, Department of ECE.*

Submitted: 01-07-2022

Revised: 04-07-2022

Accepted: 08-07-2022

ABSTRACT- Traffic sign detection is a topic in computer science and language technology with the goal of interpreting traffic signs via mathematical algorithms. Traffic sign recognition has high industrial potential in Driver Assistant System and Intelligent Autonomous Vehicles. There are two tasks in a typical traffic sign recognition system: finding the locations and sizes of traffic signs in natural scene images (traffic sign detection) and classifying the detected traffic signs into their specific subclasses (traffic sign classification). Its real-time performance is highly desirable in addition to its recognition performance. This paper aims to deal with real-time traffic sign recognition, i.e., localizing what type of traffic sign appears in which area of an input image at a fast processing time. To achieve this goal, we first propose an extremely fast detection module. Our detection module is based on traffic sign proposal extraction and classification built upon a Deep learning architecture called You Only Look Once (YOLO) Algorithms. Then we use mean subtraction algorithm to eliminate the background and to segment the foreground. Then, we use neural network algorithm to further classify and recognize the different traffic signs.

I. INTRODUCTION

Traffic signs may be divided into different categories according to function, and in each category they may be further divided into subclasses with similar generic shape and appearance but different details. This suggests traffic-sign recognition should be carried out as a two-phase task: detection followed by classification. The detection step uses shared information to suggest bounding boxes that may contain traffic-signs in a specific category,

while the classification step uses differences to determine which specific kind of sign is present (if any). (We note that the words 'detection' and 'classification' have different meanings in the general object recognition community where, as exemplified by the ImageNet competition, classification means giving an image a label rather than an object, and detection means finding the bounding box of an object in a specific category.)

Traffic sign recognition has high industrial potential in Driver Assistant System and Intelligent Autonomous Vehicles. There are two tasks in a typical traffic sign recognition system: finding the locations and sizes of traffic signs in natural scene images (traffic sign detection) and classifying the detected traffic signs into their specific subclasses (traffic sign classification). Traffic signs are designed with regular shapes and conspicuous colors to attract human drivers' attention so as to be easily captured by human drivers. However, there are many difficulties for identifying traffic signs by computer algorithms due to illumination changes, color deterioration, motion blur, cluttered background and partial occlusion, etc.

Traffic sign detection is a topic in computer science and language technology with the goal of interpreting traffic signs via mathematical algorithms. Traffic sign recognition has high industrial potential in Driver

Assistant System and Intelligent Autonomous Vehicles. There are two tasks in a typical traffic sign recognition system: finding the locations and sizes of traffic signs in natural scene images (traffic sign detection) and classifying the detected traffic signs into their specific subclasses (traffic sign classification). Its real-time performance is highly desirable in addition to its recognition performance.

ce. This paper aims to deal with real-time traffic sign recognition, i.e., localizing what type of traffic sign appears in which area of an input image at a fast processing time. To achieve this goal, we first propose an extremely fast detection module. Our detection module is based on traffic sign proposal extraction and classification built upon a YOLO algorithm. Based on the darknet framework, the training process is being performed. Google Colab is a free open source framework which is used to training the whole network

II. LITERATURE SURVEY

[1] Jayaprakash.A and Dr. C. Kezi Selva Vijila, "Detection and Recognition of Traffic Sign using FCM with SVM, Journal of Advances in chemistry, vol.13, no.6, pp.6285-6294, Feb 2017. [2] Xiaoguang HU, A. Nagasaka, and T. Miyatake, "Traffic sign recognition using Scale Invariant Feature Transform and SVM". Joints symposium of ISPRS conference, 2010 [3] Tam T. Le, Son T. Tran, "Real Time Traffic Sign Detection Using Color and Shape-Based Features" Asian conference on Intelligent information and Database systems (ACIIDS), 2010. [4] Vishal R. Deshmukh, G.K Batnaik and M.E. Patil, "Real Time Traffic Sign Recognition system based on Color image segmentation," International Journal of Computer Applications, vol.83, no.3, Dec 2013. [5] Ayoub ELAHYANI, Mohamed ELANSARI "Traffic Sign Detection and Recognition using Features Combination and Random Forests," International Journal of Advanced Computer Science and Applications (IJACSA), vol.7, no.1, Dec 2016.

III. SYSTEM DESIGN

A use case diagram in the Unified Modelling Language (UML) is a type of behavioural diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted

BLOCK DIAGRAM

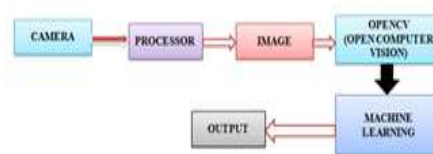


Fig. 1. Block Diagram

GENERAL PURPOSE PROCESSOR

There are five types of general-purpose processors they are, Microcontroller, Microprocessor, Embedded Processor, DSP and Media Processor.

PROCESSOR

The processor is a chip or a logical circuit that responds and processes the basic instructions to drive a particular computer. The main functions of the processor are fetching, decoding, executing, and write back the operations of an instruction. The processor is also called the brain of any system which incorporates computers, laptops, smartphones, embedded systems, etc. The ALU (Arithmetic Logic Unit) and CU (Control Unit) are the two parts of the processors. The Arithmetic Logic Unit performs all mathematical operations such as additions, multiplications, subtractions, divisions, etc and the control unit works like traffic police, it manages the command or the operation of the instructions. The processor communicates with the other components also they are input/output devices and memory/storage devices.

MICROPROCESSOR

The general-purpose processors are represented by the microprocessor in embedded systems. There are different varieties of microprocessors available in the market from different companies. The microprocessor is also a general-purpose processor that consists of a control unit, ALU, a bunch of registers also called scratchpad registers, control registers and status registers. There may be an on-chip memory and some interfaces for communicating with the external world like interrupt lines, other lines for the memory and ports for communicating with the external world. The ports often called the programmable ports that mean, we can program these ports either to be acting as an input or as an output. The general-purpose processors are shown in the below table.

Basic Components of Processor

➤ ALU stands for arithmetic logic unit, which help

out to execute all arithmetic and logic operations.

- FPU (Floating Point Unit) is also called the “Math coprocessor” that helps to manipulate mathematical calculations.
- Registers store all instructions and data, and it fires operands to ALU and save the output of all operations.
- Cache memory helps to save more time in travelling data from main memory.

Primary CPU Processor Operations are

- **Fetch** – In which, to obtain all instructions from main memory unit (RAM).
- **Decode** – In this operation, to convert all instructions into understandable ways then other components of CPU are ready to continue further operations, and this entire operations are performed by decoder.
- **Execute** – Here, to perform all operations and every component of CPU which are needed to activate to carry out instructions.
- **Write-Back** – After executing all operations, then its result is moved to write back.

TYPES OF PROCESSOR

Here, we will discuss about different types of CPU (Processors), which are used in computers. If you know how many types of CPU (Processors) are there, then short answer is 5 types of processor.

SINGLE CORE PROCESSOR

Single Core CPUs were used in the traditional type of computers. Those CPUs were able to perform one operation at once, so they were not comfortable to multi-tasking system. These CPUs got degraded the entire performance of computer system while running multiple programs at same time duration.

In Single Core CPU, FIFO (First Come First Serve) model is used, it means that couple of operations goes to CPU for processing according to priority base, and left operations get wait until first operation completed.

DUAL CORE PROCESSOR

Dual Core processor

contains two processors, and they are linked with each other like as single IC (Integrated circuit). Every processor consists its own local cache and controller, so they are able to perform different difficult operations in quickly than single core CPU.

There are some examples which are used as

dual core processors such as Intel Core Duo, the AMD X2, and the dual-core PowerPC G5.

MULTICORE PROCESSOR

Multicore processor is designed with using of various processing units’ means “Cores” on one chip, and every core of processor is able to perform the all tasks. For example, if you are doing multiple activities at a same time like as using WhatsApp and playing games

then one core handles WhatsApp activities and other core manage to another work such as game.

QUAD CORE PROCESSOR

Quad core processor is high power CPU, in which four different processor cores are recombined into one processor. Every processor is capable to execute and process all instructions on level without taking support to other left processor cores. Quad core processors are able to execute massive instructions at a time without getting waiting pools. Quad core CPU help to enhance the processing power of computer system, but it perform depends on their using computing components.

OCTA CORE PROCESSOR

Octa core processor is designed with using of multiprocessor architecture, and its design produces the higher processing speed. Octa core processor has best ability to perform multi-tasking and to boost up efficiency of your CPU. These **types of processors** are mostly used in your smart phones.

IV. IMPLEMENTATION METHOD REFERENCE IMPLEMENTATION

CPython is the reference implementation of Python. It is written in C, meeting the C89 standard with several selected C99 features. It compiles Python programs into an intermediate bytecode which is then executed by its virtual machine. CPython is distributed with a large standard library written in a mixture of C and native Python. It is available for many platforms, including Windows and most modern Unix-like systems. Platform portability was one of its earliest priorities.

OTHER IMPLEMENTATIONS

PyPy is a fast, compliant interpreter of Python 2.7 and 3.5. Its just-in-time compiler brings a significant speed improvement over CPython but several libraries written in C cannot be used with it. Stackless Python is a significant fork of CPython that implements microthreads; it does not use the C memory stack, thus allowing massively concurrent programs. PyPy also has a stackless

version. MicroPython and CircuitPython are Python 3 variants optimized for microcontrollers. This includes LEGO Mindstorms EV3. Rust Python is a Python 3 interpreter written in Rust.

UNSUPPORTED IMPLEMENTATIONS

Other just-in-time Python compilers have been developed, but are now unsupported:

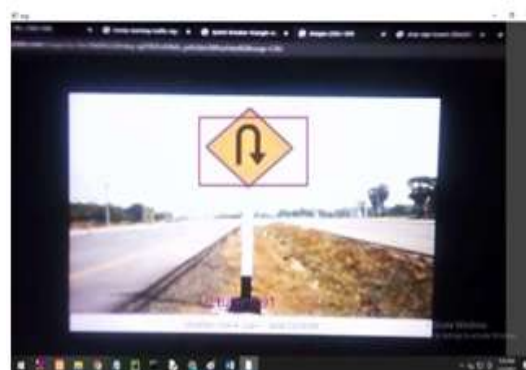
Google began a project named Unladen Swallow in 2009, with the aim of speeding up the Python interpreter five-fold by using the LLVM, and of improving its multithreading ability to scale to thousands of cores, while ordinary implementations suffer from the global interpreter lock.

Psyco is a just-in-time specialising compiler that integrates with CPython and transforms bytecode to machine code at runtime. The emitted code is specialized for certain data types and is faster than standard Python code.

In 2005, Nokia released a Python interpreter for the Series 60 mobile phones named PyS60. It includes many of the modules from the CPython implementations and some additional modules to integrate with the Symbian operating system. The project has been kept up-to-date to run on all variants of the S60 platform, and several third-party modules are available. The Nokia N900 also supports Python with GTK widget libraries, enabling programs to be written and run on the target device.

V. RESULTS

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.



Test Results: All the test cases mentioned above passed successfully. No defects encountered.

VI. CONCLUSION

This work proposed an approach of real-time traffic sign recognition system consisting of detection and classification modules. In this paper, we present an automatic traffic sign recognition system by using different traffic sign images.

We proposed YOLO algorithm for identifying different traffic sign and based on the extracted color and shape features, we can classify the different traffic sign images using neural network algorithm. The proposed approach presents a very low degree of complexity, which makes it suitable for real-time applications. At first, the different traffic sign images captured from the camera are given as the input.

In the future work experiments with more images were carried out and tested by considering various phenomenon's such as exclusion of low resolution images and including the calculation of some more statistical parameters.

REFERENCE

- [1]. Saini, S., Nikhil, S., Konda, K.R., Bharadwaj, H.S. and Ganeshan, N., 2017, June. An efficient vision-

- based traffic light detection and state recognition for autonomous vehicles. In 2017 IEEE Intelligent Vehicles Symposium (IV) (pp. 606-611). IEEE.
- [2]. Kulkarni, R., Dhavalikar, S. and Bangar, S., 2018, August. Traffic light detection and recognition for self-driving cars using deep learning. In 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA) (pp. 1-4). IEEE.
- [3]. Gao, F. and Wang, C., 2020. Hybrid strategy for traffic light detection by combining classical and self-learning detectors. *IET Intelligent Transport Systems*, 14(7), pp. 735-741.
- [4]. Ozelik, Z., Tastimur, C., Karakose, M. and Akin, E., 2017, October. A vision-based traffic light detection and recognition approach for intelligent vehicles. In 2017 International Conference on Computer Science and Engineering (UBMK) (pp. 424-429). IEEE.
- [5]. Lee, S.H., Kim, J.H., Lim, Y.J. and Lim, J., 2018, January. Traffic light detection and recognition based on Haar-like features. In 2018 International Conference on Electronics, Information, and Communication (ICEIC) (pp. 1-4). IEEE.