

An Automatic Vehicle Identification Mechanism Using a Distributed Processing System

Jun Heon Kim, Earl Kim, Beom Su Lee, Kee Hyun Choi and Dong Ryeol Shin

Abstract—Currently, vehicle related studies are activated due to the increase of vehicle. Thus, the technology using computer vision is being studied to identify the vehicle's information. It enables automatically identify the vehicle number. The image processing rate, however, is limited to the speed of data processing, and its inefficiency of processing is increase gradually. Therefore, in this paper we propose a method to enhance the efficiency of the processing using HDFS and distributed processing system.

Keywords—computer vision, vehicle identifying, OpenCV, HDFS, distributed processing system

I. Introduction

Recently, due to the development of computer vision technology, related studies are being conducted in various fields. A system for identifying vehicle number is one of the computer vision applications, which has been studied for increasing traffics. This is being used in various areas such as parking systems, stolen vehicle tracking, and crackdown on speeding vehicles. Along with automatic vehicle recognition system, it is evolving into a system that provides efficiency and accuracy. [1] Currently, most system for identifying vehicle number use image processing technique which uses data collected from CCTV or black box to check specific vehicle number. This process is consist of two steps. First of all, it extracts string value from image of vehicle number plate and process them for recognition. [2]

Jun Heon Kim
Sungkyunkwan University
South Korea

Earl Kim
Sungkyunkwan University
South Korea

Beom Su Lee
Sungkyunkwan University
South Korea

Kee Hyun Choi
Sungkyunkwan University
South Korea

Dong Ryeol Shin
Sungkyunkwan University
South Korea

As the number of vehicles increased, the accuracy and speed is regarded as the most important factors in the system for identifying. When we try to track a specific vehicle, it takes lots of time to extract desired data from the number of plates in an image. Additionally, when we perform repetitive data analysis in one image, the consumption of time will be increased inevitably.

In this paper, we propose an efficient mechanism to analyze the image data recognized by using OpenCV(Open Computer Vision) through distributed processing system. In section 2, we describe OpenCV, one of the image processing techniques, and HDFS(Hadoop Distributed File System). And section 3, we describe the proposed system for identifying vehicle number using OpenCV and HDFS, and finally we conclude this paper with comments and remarks on future work in section 4.

II. Related Work

A. OpenCV

OpenCV is open-source C library for computer vision, which is developed by intel. It is being a BSD-License product, we use and modify it easy. As a library focused on real-time image processing, support interfaces such as C, C++, Python, Java. Besides, Support almost platform such as Windows, Linux, Mac OS, iOS, so it is easy to migrate other platform from original form. And it has more than 2500 optimized algorithms, which includes computer vision and machine learning algorithms. furthermore, it can process more than 30 frame per second, so many people use it for many fields like image processing, Human-Computer Interaction, Object Identification. [3]

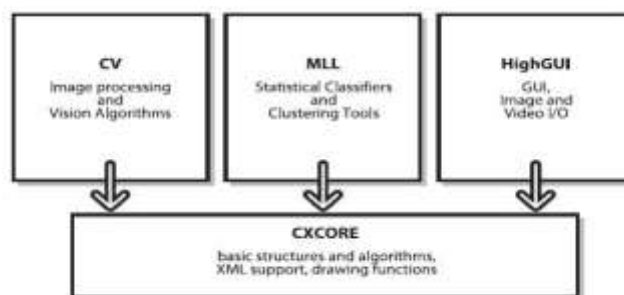


Figure 1. OpenCV architecture

Figure 1 shows openCV's architecture. CV(Computer Vision) conduct image processing, structural analysis of images, camera calibration, and so on. MLL(Machine Learning Library) conduct about clustering and classification. HighGUI print out image/video and user interface GUI. And CXCORE includes matrix operation, data structure, memory management, dynamic loading, and so on.

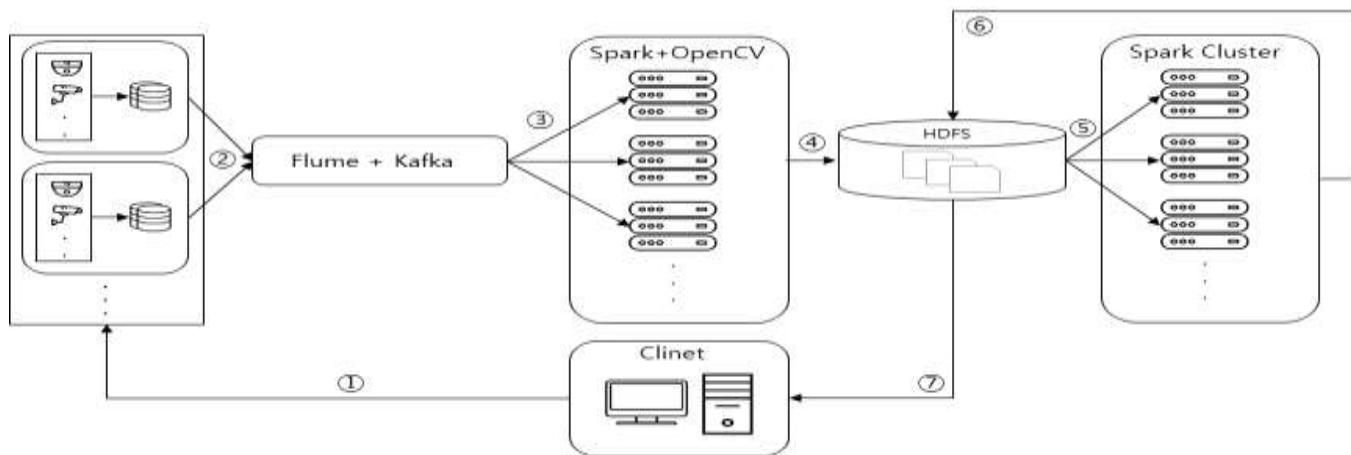


Figure 4. Block diagram about system

B. HDFS

HDFS(Hadoop Distributed File System) is data storing system for distributed data processing. Typically, the file size of HDFS is larger than gigabytes unit, so it can support huge file. It stores node by 64M unit block, and duplicate a block to other nodes in order to respond flexibly when failure occurs some nodes. The composition of the HDFS is divided into *namenode* and *datanode*, *namenode* manages the entire file system and *datanode* store files in blocks. [4]

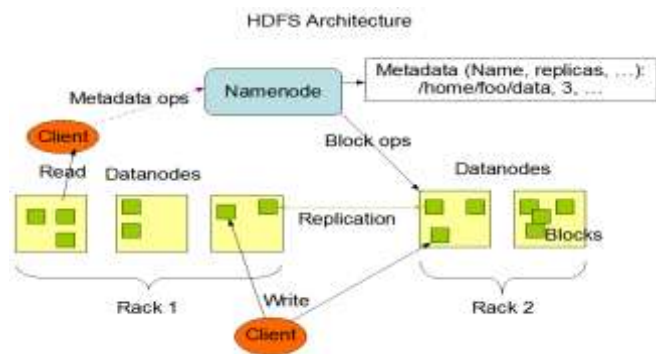


Figure 2. Hadoop HDFS architecture

Figure 2 shows architecture of Hadoop HDFS. Basically it has a master/slave architecture, an HDFS cluster has single *namenode* and a number of *datanode*. If client request read/write, preferentially reads the location of the data block from the *namenode*. And reads data from the data block, or writes the file to the data block. At this moment, *namenode* performs the tasks that return location of the *datanode* which is the closest to the client when reading or indicate the location of the next block if block is exceed when writing.

C. Spark

Apache Spark is cluster computing package for distributed data analysis. Spark provides high speed and versatility. It is 10 times faster than MapReduce on disk, and 100 times faster than in memory. And it provide multiple interface. It provide Java, Scala, Python, and so on. [5]



Figure 3. Apache Spark architecture

Figure 3 is architecture of Apache Spark. It is divided into 4 parts in library layer and SQL exposes the datasets and enable you to start SQL with visualization tools, Streaming process live data, MLlib conduct machine learning algorithms, and GraphX proceed graph processing. Each part in library layer are able to exchange data. And Spark Core, containing API, provide user to access library layer. [6]

III. System Design

For the system, you need server that collect video(3D) data and convert it into images(2D), and allows video data to be imaged using distribute processing system. And, you also need Hadoop HDFS for storing a number of image file, and second Spark Cluster server for image file analysis in HDFS.

Figure 4 is the block diagram about system. First, client transmit CCTV video file that client want to analyze to server for Image analysis. To process a lot of video data, server composes distributed processing environment. In here, use Spark and OpenCV, capture vehicle number from video file and change it to image file. All image files created on the server are saved in one HDFS, and the Spark Cluster Server use it to find specific vehicle number that client want to find. It provide higher processing speed beside existing image processing technique because size of image is smaller than size of video file.

IV. Conclusion

In this system, we propose a measure to improve the identification system of the vehicle number recognition using OpenCV, HDFS, distributed processing system. By converting video information into image file and save in HDFS, it can reduce the time for the vehicle number analysis because there is no need to view the video repeatedly when we repeat vehicle number tracking.

Acknowledgement

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(NRF2016R1A6A3A11930831)

References

- [1] Michalopoulos, Panos G. "Vehicle detection video through image processing: the autoscope system." *IEEE Transactions on vehicular technology* 40.1 (1991): 21-29.
- [2] Patel, Chirag, Dipti Shah, and Atul Patel. "Automatic number plate recognition system (anpr): A survey." *International Journal of Computer Applications* 69.9 (2013).
- [3] Bradski, Gary, and Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library.* " O'Reilly Media, Inc.", 2008.
- [4] Shvachko, Konstantin, et al. "The hadoop distributed file system." *Mass storage systems and technologies (MSST), 2010 IEEE 26th symposium on.* IEEE, 2010.
- [5] Shanahan, James G., and Laing Dai. "Large scale distributed data science using apache spark." *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* ACM, 2015.
- [6] Meng, Xiangrui, et al. "Mllib: Machine learning in apache spark." *Journal of Machine Learning Research* 17.34 (2016): 1-7.



Jun Heon Kim received the B.S degree in the Information & Communication Engineering from Sung-Kyul University, Korea, in 2016. Currently, he is working for M.S. in Department of Computer Science and Engineering at Sungkyunkwan University, Korea. His research interests include Machine learning and Big data.



Earl Kim received the B.S degree in the Electronic Engineering from Kyung-Hee University, Korea, in 2015. Currently, he is working for M.S. in Department of Computer Science and Engineering at Sungkyunkwan University, Korea. His research interests include Machine learning and Big data.



Beom su Lee received the B.S degree in the Cyber security from Kyungil University, Korea, in 2017. Currently, he is working for M.S. in Department of Computer Science and Engineering at Sungkyunkwan University, Korea. His research interests include Machine learning and Big data.



Kee Hyun Choi received the Ph.D. degree in the Electrical, Electronic & Computer Engineering from Sungkyunkwan University, Korea, in 2005. Currently, he is working for the Dept. of Software, College of Software, Sungkyunkwan University, Korea as a Research Professor (Research Fellow). Interest areas include WiFi MAC, P2P, AQM and IoT.



Dong Ryeol Shin received the B.S., M.S., and Ph.D. degrees in Electrical Engineering from the Sungkyunkwan University in 1980, the Korea Advanced Institute of Science and Technology (KAIST) in 1982, and the Georgia Institute of Technology in 1992, respectively. During 1992-1994, he had worked for Samsung Data Systems, Korea, where he was involved in the research of Intelligent Transportation Systems. Since 1994, he has been with the Department of Computer Science and Engineering at Sungkyunkwan University where he is currently a Professor in Network Research Group. His current research interests lie in the areas of mobile network, ubiquitous computing, cloud computing, and bioinformatics. And he is actively involved in the security of vehicular area networks, and the implementation and analysis of big data platform, applicable to 3D image processing of robotic arms.