ISSN:1226-7244 (Print)
ISSN:2288-243X (Online)
j.inst.Korean.electr.electron.eng.Vol.21,No.2,142~145,June 2017
논문번호 17-02-06
http://dx.doi.org/10.7471/ikeee.2017.21.2.142

하둡을 이용한 번호판 인식 시스템

A Licence Plate Recognition System using Hadoop

박 진 우*, 박 호 현*★

Jin-Woo Park*, Ho-Hyun Park*★

Abstract

Currently, a trend in image processing is high-quality and high-resolution. The size and amount of image data are increasing exponentially because of the development of information and communication technology. Thus, license plate recognition with a single processor cannot handle the increasing data. This paper proposes a number plate recognition system using a distributed processing framework, Hadoop. Using SequenceFile format in Hadoop, each mapper performs a license plate recognition with a number of image data in a data block Experimental results show that license plate recognition performance with 16 data nodes accomplishes speedup of maximum 14.7 times comparing with one data node. In large dataset, the recognition performance is robust even if the number of data nodes increases gradually.

요 약

현재 활용되는 영상 데이터가 고화질 고화소 추세이며, 정보통신기술의 발달로 인해 이미지 데이터의 사이즈와 양이 기하급수적으로 증가하고 있다. 이러한 영상데이터를 효율적으로 처리한다면 다양한 컨텐츠로 활용할 수 있지만 기존의 단일컴퓨터로 처리하기에는 늘어나는 데이터를 처리하기에는 한계가 있다. 본 논문은 분산 처리 프레임워크인 Hadoop을 이용하여 번호판 인식 시스템을 제안한다. SequenceFile 포맷을 이용하여 매퍼당 여러 개의 이미지 데이터를 가지고 있는 데이터 블록을 인풋으로 받아 번호판 인식을 수행한다. 실험결과 하둡의 데이터 노드 1개와 비교하여 데이터 노드 16개에서 최대 14.7배의 속도향상을 보였으며, 데이터 셋의 크기를 10배 증가하여도 데이터 노드가 점진적으로 늘어남에 따라 번호판 인식 속도의 강인함을 확인하였다.

Key words : Licence plate recognition, ANPR, Hadoop, Big data, CCTV

* Dept. of Electrical and Electronics Engineering, Chung-Ang University
★ Corresponding author
e-mail: hohyun@cau.ac.kr tel: 02-820-5345
Manuscript received Jun. 19, 2017; revised Jun. 28, 2017 ; accepted Jun. 30, 2017

Ⅰ. 서론

현재의 CCTV와 블랙박스 분석 및 처리 기술은 빅데이터 환경을 고려하지 못하고 있다. CCTV는 저장매체와 카메라 모두가 아날로그 방식에서 디지털 방식으로 전환되어, 고화질, 고화소 추세에 있으며 '스마트 CCTV'와 같은 지능형 관제 서비스 같은 다양한 응용 방안이 제시되고 있다. 반면, 블랙박스는 CCTV에 비해 설치비용이 1/60 정도로 개인에게 널리 보급되었으나[1], 개인

정보 보호 차원에서 활용 방안이 제한적이다. 만약 전국의 CCTV와 블랙박스 영상 데이터를 한곳으로 수집할 수 있다면 번호판 인식을 통해 차량 조회, 위치, 경로 등을 추적해 낼 수 있으며, 더욱 정확한 도로의 교통 예측 시스템 같은 다양한 활용방안들이 생겨날 것이다. 하지만 기하급수적으로 늘어나는 CCTV와 블랙박스의 영상데이터를 효율적으로 처리할 방안을 먼저 생각해야 한다. 이러한 빅데이터를 처리하기 위한 분산처리시스템 프레임워크인 하둡[2]은 구글 파일 시스템[3]과 맵리듀스[4]를 기반으로 구현되었으며, 본 논문에서는 빅데이터 영상데이터를 처리하기 위해 하둡을 이용하고, 번호판 이미지 프로세싱 라이브러리인 JavaCV[5]를 이용하여 번호판 인식시스템을 수행하고자 한다.

II. 번호판 인식 알고리즘

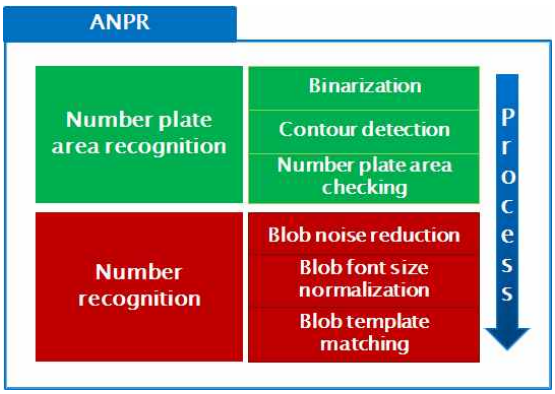


Fig. 1. ANPR algorithm
그림 1. 번호판인식(ANPR) 알고리즘

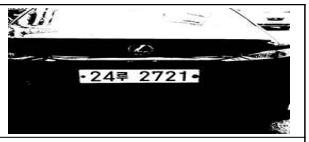
가. 번호판 영역 추출 단계
(1) 영상 이진화
gray-scale 이미지로 변환 후 이진화 처리.
(2)윤곽선 에지 검출
sobel 에지 필터[6]를 적용하여 에지검출 후 Contour Tracing[7]을 이용하여 에지의 윤곽선을 추출.
(3) 번호판 영역 추출
추출된 윤곽선 좌표 내에 Labeling[8]을 수행하여 추출된 Blob이 7개 이상인 영역을 후보 번호판 영역으로 추출.

번호판 영역추출을 위해 각 단계를 거치며, 가능성 있는 번호판 영역을 추출한다. 실제 번호판[9]의 크기가 520mm x 110mm 로서 약 5:1 의 비율이지만 이미지로 나타나는 번호판은 찍힌 각도에 따라 변화하기 때문에 이를 고려하여 3:1 - 6:1 비율 사이의 직사각형을 가능성 있는 번호판 영역이라 정하였다. 번호판영역 후보를 구하고, Labeling을 하여 Blob이 7개 이상인 후보 번호판 영역은 문자인식 단계로 넘어가게 된다.

나. 문자 인식 단계
(1) 잡영(Noise) 제거
후보번호판 영역에서 문자영역이 될 수 없는 크기의 Blob 만 추출하여 잡영(Noise)을 제거.
(2) 내부 Blob의 크기 정규화
문자 인식을 위한 단계로 매칭할 문자의 Font의 사이즈와 같게 Blob들을 정규화 처리.
(3) 문자 Template Matching
문자인식을 위해 미리 준비된 이진화된 폰트와 Blob의 AND 연산을 통해 매칭율 기반으로 인식.

번호판 영역을 추출 후 문자 인식 단계에서는 우선 불필요한 잡영(Noise) 즉 번호판 내의 문자가 아니라고 판단되는 Labeling 된 Blob 들을 제거한다.
잡영 Blob들을 제거하기 위해 후보 번호판 영역 넓이 10%이하, 높이 20%이상의 blob들만 추출한다. Blob들이 번호판 각도에 의해 크기가 각각 다를 수 있기 때문에 문자인식을 위해 Blob들의 크기를 매칭 할 Font의 크기와 같게 정규화 한다.
Labeling으로 추출된 Blob이 번호판 문자 Font에 따라 자음과 모음이 분리되어 Blob이 추출된 경우, 즉 검출된 Blob이 8개 일 경우 문자영역인 3번째와 4번째 Blob의 min/max 좌표를 이용하여 Blob을 합병한다. 문자인식의 최종단계로서 Blob의 3번째를 제외한 각각의 1 - 7 번째의 Blob은 미리 준비된 이진화 처리된 Font와 매칭하여 AND 연산 후 매칭률이 가장 높은 문자의 인덱스를 인식한 결과로 출력한다.


original image


Binarization

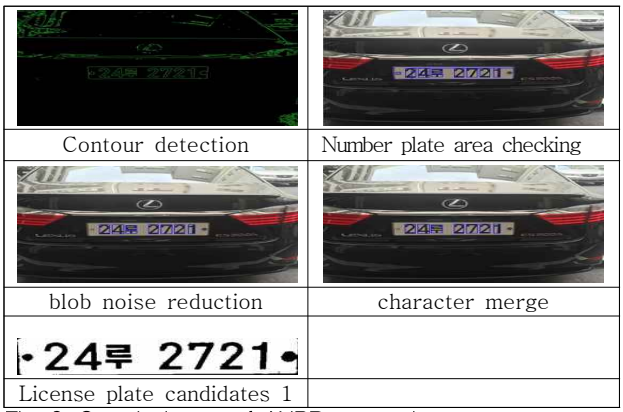


Fig. 2. Sample image of ANPR processing
그림 2. 번호판 인식 처리과정 샘플 이미지

III. 제안하는 하둡+번호판인식(ANPR) 시스템

제안한 번호판 인식을 하둡을 이용하여 분산 처리로 수행하는 흐름도는 다음과 같다.

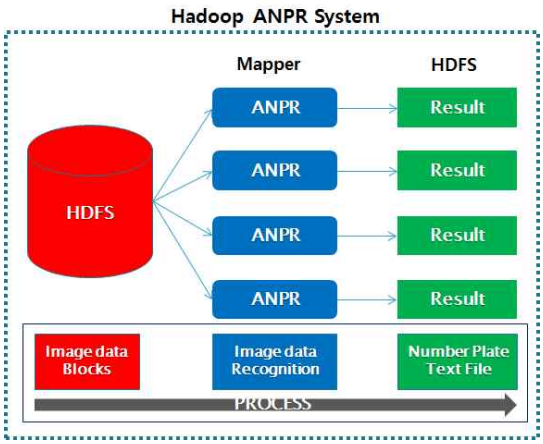


Fig. 3. A integrated Hadoop ANPR System
그림 3. 통합된 ANPR 시스템

맵리듀스에서 이미지파일을 이용하기 위해 본 논문에서는 SequenceFile를 이용한다. MapReduce는 본래 텍스트분석을 위해 설계되어있기 때문에 일반적인 방법으로는 이미지파일과 같은 바이너리 파일은 읽어오지 못한다. 따라서 이미지를 SequenceFile format으로 압축하여 사용한다. SequenceFile을 사용할 때 키는 파일이름으로, 그리고 값은 파일 내용으로 지정하였다. 따라서 맵의 입력으로 Key = 파일명, Value = 이미지가 된다.

하둡에서 맵에서 처리되는 데이터는 HDFS에서 데이터 Block (default 64MB) 당 Mapper 하나로 처리된다. 따라서 하둡의 Map함수 내에 SequenceFile 포맷을 이용하여 압축된 데이터 Block Size 만큼의 이미지들을 번호판 인식기를 통해 인식할 수 있다. 아웃풋은 Map 함수 내에서 어떠한 아웃풋으로 출력할 지 설정 가능하며, 이 시스템에서는 모든 이미지를 인식하고 인식된 결과를 Reduce를 통해 집계할 필요가 없으므로, Reduce를 사용하지 않고 Mapper에서 바로 인식결과를 출력하였다. 따라서 Mapper당 하나의 출력파일을 생성한다.

IV 실험 결과 및 분석

본 논문에서 제안한 빅데이터 번호판인식 시스템 성능을 평가하기 위해 다음과 같이 클러스터를 구축하였다. 마스터 노드 1대와 슬레이브노드 16대로 클러스터를 구성하였고, 각 시스템 사양은 마스터노드 CPU: Intel E3-1230v2 3.3GHz, 슬레이브 노드 CPU : Intel i5-3570 3.4GHz, 공통사항으로 RAM: 4GB, HDD: 1TB 7200rpm 구성되어 있으며 클러스터는 100Mbps의 네트워크로 연결되어 있다. 각 시스템에는 Ubuntu 13.10 32bit OS, Java 1.8.0_50 32bit, 그리고 Hadoop 1.2.1 버전으로 설치하였다. 실험을 위해 하둡을 다음과 같이 설정하였다. HDFS의 Replica를 1로 설정하였고, 각 노드에서 실행되는 Mapper의 최대 개수를 2개로 설정하였다.

번호판 인식률은 이미지 데이터의 촬영된 각도와 번호판 크기 등에 영향이 있지만 본 논문에서는 처리속도 확인에 중점을 두므로 번호판 인식률 100%의 이미지 데이터만을 사용하였다. 실험에 사용된 이미지 데이터 셋은 2가지로, 실제로 촬영하여 얻은 1920x1210 해상도의 jpg 포맷의 이미지 데이터 641개를 얻은 후 복제하여 6410(2.5GB), 64100(25.06GB)개를 생성하여 사용하였다. 또한 실험을 위해 번호판 인식 시스템에서 이미지 당 하나의 번호판 인식을 처리할 수 있게 설정 하였다.

Table 1. Processing time of Haddop+NPR system comparing dataset A and B

표 1. 제안하는 번호판인식 시스템의 수행시간 결과

Number of Datanode	Dataset A Processing Time(min)	Dataset B Processing Time(min)
1	96	958
2	49	489
4	25	241
8	13	124
16	7	65

실험결과에 의하면 데이터셋 A에서 1개의 데이터 노드의 번호판인식의 수행시간은 약 96분이며 약 0.9 초당 하나의 이미지를 인식할 수 있었다. 16개의 데이터노드 에서 번호판인식 수행시간은 약 7분으로서 0.065초당 하나의 번호판 이미지를 인식할 수 있었다. 따라서 노드가 16개 까지 늘어남에 따라 거의 선형적으로 성능이 증가함을 보였고 최대 데이터노드 1개 대비 약 81분의 수행시간 감소가 있었으며, 약 13.7배의 Speedup 을 관측했다.

데이터 셋 B를 데이터 셋 A의 10배로 늘려서 수행시간을 측정해본 결과 1개의 데이터노드 번호판 인식 수행시간이 약 958분으로 약 16시간이 걸렸으며, 데이터 노드를 늘려감에 따라 최대 16개의 데이터 노드에서 수행시간이 약 65분으로 측정되었다. 최대 1대의 데이터 노드 대비 수행시간이 923분이 감소하였으며, 14.7배의 Speedup을 보였고 데이터셋 A와 Speedup이 데이터증가와 비례하게 거의 변함이 없었다. 이에 따라 하둡을 이용한 번호판 인식에서 고화질 이미지 데이터 양이 증가함에도 성능 저하 없이 데이터를 처리할 수 있다는 것을 확인 할 수 있었다.

Ⅵ 결론

본 논문에서는 빅데이터 이미지 데이터를 처리하기 위하여 번호판인식을 분산처리 프레임 워크인 하둡을 이용하여 처리 하였다. 실험에서는 하둡을 이용한 멀티노드의 비교에서 노드를 점진적으로 늘려감에 따라 16개의 데이터노드까지 비교적 선형적으로 성능이 올라갔다. 실험 결과에 의하면 번호판 인식 알고리즘을 이용하여 인식된 최종결과를 얻을 때까지 1개의 데이터노드와 비교하여, 16개의 데이터노드 에서 최대 약 14.7배의 Speedup이 관측되었다.

현재 고해상도 영상과 급격히 쌓이는 영상 데이터 사이즈가 증가함에 따라 기존의 단일 컴퓨팅으로만 처리하였던 번호판 인식 알고리즘을 하둡을 이용하여 분산처리환경에 적용하여 처리함으로서, 빅데이터 환경에 맞게 데이터 셋의 사이즈가 커져도 데이터 노드가 증가함에 따라 강건한 성능 증감을 확인할 수 있었다.

References

[1] J. H. Seol, “Dissemination and Application of Traffic Accident Video Recorder,” Monthly Magazine on Transportation Policy, Vol. 184, pp13-18, 2013

[2] http://hadoop.apache.org.

[3] J. Dean and S. Ghemawat, “MapReduce: simplified data processing on large clusters,” *Communications of ACM*, vol. 51, no. 1, pp. 107-113, 2004. DOI:10.1145/1327452.1327492

[4] S. Ghemawat, H. Gobioff and S. Leung, “The Google file system” *ACM*, vol. 37, no. 5, pp. 29-43, 2003. DOI:10.1145/945445.945450

[5] https://github.com/bytedeco/javacv

[6] Ondrej Martinsky, Algorithmic and Mathematical Principles of Automatic Number Plate Recognition Systems, Brno University of Technology, 2007

[7] Contour Tracing Algorithms, http://www.imageprocessingplace.com/downloads_V3/root_downloads/tutorials/contour_tracing_Abeer_George_Ghuneim/alg.html

[8] Connected-component labeling, https://en.wikipedia.org/wiki/Connected-component_labeling

[9] National Law Information Center, Notification on standards such as licence plate registration for automobiles, 2013.