

라즈베리 파이를 이용한 무선 자동차번호판 영역 추출 모듈 개발

김동경[†], 우중호^{**}

Development of Wireless License Plate Region Extraction Module Based on Raspberry Pi

Dong-Kyung Kim[†], Chong-Ho Woo^{**}

ABSTRACT

A wireless license plate region extracting module is proposed for LPR system controlling multiple gates. This module is cheaply implemented using Raspberry Pi which is open source and high performance. First, as the upper 1/3 of the captured image is discarded as it has no useful information on license plate. Using the OpenCV libraries the edge image is got by Canny algorithm after applying Gaussian filtering to gray image, and the labeling is conducted for 4 consecutive numbers in license plate. These numbers are located using various decision equations, and expanding the numbers region the final license plate region can be extracted. The result image is transferred to Server using wifi direct. Using the proposed module it becomes easy to set up and maintain the LPR system. The experimental results showed that the successful extracting rate was 98.4% using 500 car images with 640 × 480 resolution.

Key words: LPR System, OpenCV, License Plate Extraction, Raspberry Pi, Wifi Direct

1. 서 론

차량 번호판 인식(LPR: Licence Plate Recognition) 시스템에 대한 연구는 1980년대부터 활발히 진행되어왔고, 현재 시장에서는 여러 형태의 제품들이 판매되고 있다. 그러나 아파트와 유료 주차장 등에서 주차관리를 위한 수요는 많지만 아직 널리 사용되지 못하는 이유는 여전히 고가이고 운용의 편의성이 부족한 것으로 판단된다. 차량 번호판 인식 시스템은 크게 번호판 영역 추출과 문자 인식과정으로 나눌 수 있다[1,2].

번호판 영역 추출은 전체 영상에서 해당 번호판을

추출해야 함으로 상대적으로 처리할 계산이 많아 전체 시스템의 성능에 큰 영향을 준다. 차량 번호판 영역 추출 방법에는 이진영상에 허프(Hough) 변환을 적용하여 수직 및 수직방향의 에지를 탐색하고, 차량 번호판 영역의 가로와 세로 비율 정보를 활용하여 추출하는 방법[3,4], 명암도 변화와 기하적인 구조를 활용하는 방법[5], 컬러 영상 정보를 활용하는 방법[6,7] 등이 있다. 또한 차량의 번호판에서 특징이 잘 보존된 4개의 큰 숫자를 우선적으로 찾아 이를 기준으로 번호판 영역을 추출하는 방법들이 제안되었다[8,9]. 기존 시스템에서는 밤과 낮, 날씨, 장소의 조건에 따라 문제들을 극복하기 위한 알고리즘들을 적용

* Corresponding Author : Chong-Ho Woo, Address: (608-737) Youngso-ro 45, Nam-gu, Pusan, Korea, TEL : +82-51-629-6250, FAX : +82-51-629-6264, E-mail : chwoo@pknu.ac.kr

Receipt date : Aug. 7, 2015, Revision date : Aug. 12, 2015
Approval date : Aug. 13, 2015

[†] Dept. of Computer, Industrial Graduate School, Pukyong National University
(E-mail : sky1342@nate.com)

^{**} Dept. of Computer, Pukyong National University

* This research was supported by a Research Grant of Pukyong National University(2014)

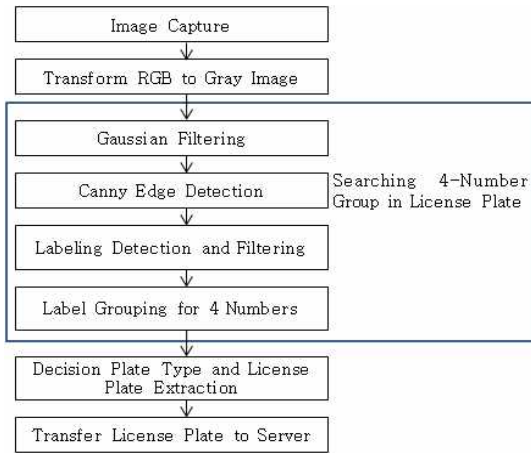


Fig. 1. Procedure for image processing in the license plate extraction module.

하였다. 본 연구에서는 주차 관리에 사용할 LPR 시스템에 사용할 번호판 영역을 추출하는 모듈로 한정하므로 앞의 조건을 고려하지 않았다.

한편 라즈베리 파이(Raspberry Pi)는 2012년 2월에 출시된 소형 교육용 컴퓨터 보드로 PC에서 사용되는 다양한 주변장치들을 직접 연결해서 적은 비용으로 리눅스 기반의 컴퓨터를 구성할 수 있다. 또한 인터넷 연동이 가능하므로 편리한 개발 환경을 제공한다. 본 연구에서는 OpenCV[10]에서 제공되는 각종 라이브러리를 사용해서 각 단계에서 요구되는 영상처리 작업을 수행했다. 또한 USB에 와이파이 동글(wifi dongle)을 사용해서 LPR 서버와 와이파이 다이렉트 방식으로 모듈에서 추출한 번호판 영역을 전송했다.

본 논문에서는 입력된 영상에서 차량의 번호판을 추출해서 서버로 전송하는 모듈을 제안한다. 현행의 모든 차량은 4자리의 숫자를 포함하고 있다. 이를 근거로 4개의 연속된 숫자들을 탐색한 후, 좌측과 상측

으로 영역을 확장해서 1줄 번호판 혹은 2줄 번호판을 조사해서 판단하고, 그에 따라 전체 차량 번호판을 추출하였다. 추출한 번호판 영상은 와이파이로 서버에 전송한다. LPR 시스템은 하나의 서버(PC)와 필요한 개수의 모듈들이 무선으로 연결되어 구성된다. 제안하는 모듈을 사용하면, 카메라 수의 증가에 따른 서버의 부하를 줄일 수 있다. 또한 규모에 맞는 시스템의 설치와 운용이 용이할 것이다.

본 논문의 구성은 다음과 같다. 2장에서 번호판 영역을 찾는 각 단계에서 필요한 영상처리를 다루었고, 3장에서 모듈을 구성하는 시스템 측면에서 간단하게 살펴보았다. 그리고 4장에서는 제안한 번호판 추출 모듈을 평가하고 5장에서 결론을 맺는다.

2. 차량 번호판의 영역 추출을 위한 영상처리

현재 국내 모든 차량의 번호판에는 4개의 연속된 숫자가 포함되어 있다. 본 논문에서는 이를 근거로 번호판 내부의 4개 번호를 우선적으로 검출하여, 번호판 영역을 추출한다. 설계 및 구현할 번호판 영역 검출 모듈은 Fig. 1과 같은 과정들을 수행한다.

2.1 영상입력의 크기 조절 및 그레이 영상

카메라에서 입력되는 영상은 Fig. 2와 같이 640×480 크기의 RGB 영상이다. 주차장 관리에 사용되는 카메라는 고정된 위치에서 출입하는 차량 영상을 촬영한다. 차량의 번호판은 영상의 아래 부분에 위치하므로 촬영한 영상에서 위의 1/3 부분에서는 얻을 정보가 없으므로 처리대상에서 제외한다. 따라서 아래의 640×320 부분만을 처리함으로써 모듈에서의 전체 계산을 크게 줄일 수 있다. 또한 축소된 컬러 영상에서 컬러 정보는 번호판 영역의 추출에 사용하지 않으므로 계산의 효율성을 위해서 그레이 영상으로



Fig. 2. (a) Reduced gray image, (b) Obtained edges image after applying Gaussian filter.

변환한다.

2.2 필터링과 윤곽선 검출

영상에서 차량 번호판을 찾기 위해서 에지 정보를 이용한다[8]. 이를 위해서 본 논문에서는 캐니 에지 추출 알고리즘을 이용하였다. 검출된 에지 정보에서 레이블링과 그룹화 및 적절한 조건 검사들을 통해서 차량 번호판 영역을 탐색한다.

에지를 추출하기 전에 우선 번호판 내의 숫자를 찾는 과정에서 불필요한 잡음을 제거한다. 사용할 적합한 필터를 선택하기 위해서, 각각의 필터를 적용한 후의 레이블 개수 및 수행속도를 실험적으로 구한 결과는 Table 1과 같다. 가우시안 필터가 차량번호판 영역 추출모듈의 영상처리에 가장 적합하였다.

축소한 그레이 영상의 한 예인 Fig. 2의 (a) 영상에 가우시안 필터를 적용해서 잡음을 제거한 후, 캐니 알고리즘을 적용하면 Fig. 2의 (b)와 같은 에지 영상이 구해진다.

2.3 레이블링 및 번호판의 숫자 후보 선택

레이블링은 영상의 연속하는 화소들을 나타낼 때 사용되는 알고리즘으로 인접 화소 사이의 영역을 분리할 때 사용한다. Fig. 3은 구해진 에지 영상에서 각각의 연결된 에지들을 사각형으로 나타낸 레이블

Table 1. Comparison of filters used for noise removing

| Filter Applied | Number of Labels | Speed(sec) |
|------------------|------------------|------------|
| No Filter | 568 | 0.00 |
| Box Filter | 240 | 0.06 |
| Bilateral Filter | 369 | 9.72 |
| Median Filter | 268 | 0.05 |
| Gaussian Filter | 217 | 0.01 |



Fig. 3. Labeling result.

링 결과이다. 각 레이블은 이차원 평면의 좌표공간에서 두 점 $P_1(x_1, y_1)$ 와 $P_2(x_2, y_2)$ 로 표시된다. 두 점의 좌표값을 사용해서 레이블 간의 위치 관계를 이차원 공간에서 구할 수 있다.

구해진 레이블들은 그 크기가 아주 다양하다. 그러나 카메라를 고정시키고 일정한 거리에서 촬영한 차량의 이미지에서 번호판에 포함될 4개 숫자의 후보에 해당하는 레이블의 크기는 실험적으로 구한 상수값 α 와 β 를 설정할 수 있고, 식 (1)의 가로/세로비는 W/H 로 하고 그 값이 < 0.9 인 것으로 판단한다.

$$\alpha < H < \beta, W / H < 0.9 \tag{1}$$

여기서 H와 W는 숫자 후보를 나타내는 각각 레이블의 높이와 폭을 의미한다. 이 조건을 적용하여 번호판에 포함되는 4개의 숫자 후보에 해당하는 레이블을 나타내면 Fig. 4와 같다.

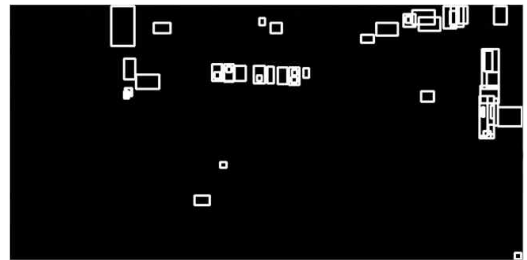


Fig. 4. Labels selected as the candidates of 4 numbers in license plate.

2.4 그룹화

그룹화는 인접한 위치에 있는 숫자들에 대한 위치 조건을 Fig. 5와 같이 설정해서 그룹화(grouping)하는 과정이다. 우선 이 4개의 숫자들을 찾고 이를 근거로 번호판 영역을 추정한다.

번호판 내의 숫자 후보인 레이블의 그룹화 조건은 레이블 간의 간격(Gap), 겹침(Overlay), 포함(Inclusion), 떨어짐(Difference)을 나타낸다. 본 논문에서

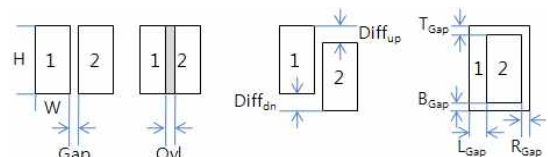


Fig. 5. Positional conditions for grouping the 4 numbers in license plate.

는 다음의 4가지 조건식을 실험적으로 설정했다.

$$\text{Gap} < \max(H * 0.2, 10) \quad (2)$$

$$\text{Ovl} < W * 0.15 \quad (3)$$

$$\text{Diff}_{\text{up}} < \max(H*0.3, 10), \text{Diff}_{\text{dn}} < \max(H*0.3, 10) \quad (4)$$

$$!(R_{\text{Gap}} \geq 0 \ \& \ L_{\text{Gap}} \geq 0 \ \& \ T_{\text{Gap}} \geq 0 \ \& \ B_{\text{Gap}} \geq 0) \quad (5)$$

식 (2)는 그룹 내 이웃 레이블의 x축 거리가 $\max(H * 0.2, 10)$ 미만으로 한정한다는 의미이다. 식 (3)은 그룹 내의 이웃 레이블이 X축으로 겹쳐지는 부분의 길이가 $W * 0.15$ 미만으로 제한한다. 식 (4)의 조건은 그룹 내에서 이웃 레이블의 Y축 좌표의 상단과 Y축 좌표의 하단의 차이가 각각 $\max(H * 0.31, 10)$ 미만이고, $\max(H * 0.3, 10)$ 미만이어야 함을 나타낸다. 마지막으로 식 (5)의 조건은 그룹 내에서 레이블이 다른 레이블을 포함할 때 그 레이블을 탈락시키기 위한 조건이다.

번호판 추출에서 우선 4개의 숫자를 탐색하며, 원영상의 숫자 부분이 손상되어 4개의 숫자 중 1개의 숫자가 레이블 되지 못할 경우를 고려한다. 또한 숫자 및 문자와 이들이 분리되어 레이블의 수가 증가될 수 있다. 그래서 그룹 내에 존재하는 레이블의 수는 $3 \leq n \leq 12$ 로 한정했다.

연속한 4개의 숫자 후보들을 선택하는 그룹화 과정은 다음과 같다. ① 레이블들을 X축을 기준으로 오름차순으로 정렬하여 번호를 부여한다. ② 첫 번째 그룹을 생성하여 첫 번째 레이블을 포함시킨다. ③ 나머지 각각의 레이블은 이미 형성된 그룹 내의 레이블들과 비교한다. ④ 앞의 식 2-6의 위치조건을 모두 만족하면 해당 그룹에 포함시키고, 만족하지 못하면 레이블은 새로운 그룹을 생성하거나 버려진다. ⑤ 모든 레이블들에 대해서 과정 ③과 ④의 단계를 반복한다.

앞의 그룹화 과정의 처리결과의 예시는 Fig. 6과

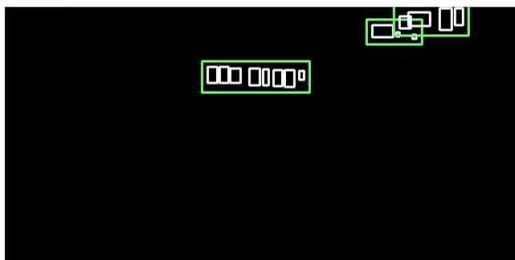


Fig. 6. Grouping result for 4 numbers in license plate.

같다. 앞의 조건식들을 만족하지 못하는 레이블들이 대상에서 제외됨을 확인할 수 있다.

2.5 차량번호판영역 후보 선택 방법

번호판 후보 영역 후보가 여러 개 나타나면 이중에서 최종 하나만을 선택해야 한다. 이 경우 각 그룹에 속하는 레이블들이 숫자인지를 판단한다. 본 논문에서는 편의상 인터넷에 공개된 Tesseract-OCR[7]을 이용하여 숫자여부를 판단해서 적어도 1개 이상의 숫자를 포함하는 그룹을 선택했다. 그러나 차량영상의 전면에 번호판이 아닌 숫자 혹은 비슷한 패턴이 존재하면 2개 이상의 그룹이 선택될 수도 있지만, 그러나 그러한 가능성은 극히 낮을 것이다. Fig. 7에서와 같이 2개 이상의 그룹이 나타나면, 영상의 중심점에서 가까운 그룹을 번호판의 숫자 영역이 위치할 가능성이 가장 높은 영역으로 판단하여 이것을 최종 후보로 선택한다. 본 연구에서는 카메라가 주차장과 같이 고정된 위치에 설치되는 것으로 가정하므로 이 방식으로 선택할 수 있다.

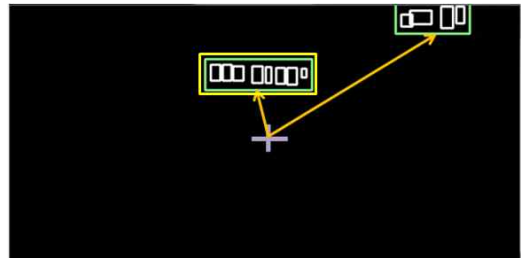


Fig. 7. The final candidate group for 4 numbers in license plate.

2.6 차량 번호판 추출

현재 사용되고 있는 모든 차량의 번호판은 문자 및 숫자들로 1줄 혹은 2줄의 형태를 갖는다. 최종적으로 선택된 그룹은 번호판 안의 4개 숫자들로 구성된 그룹이다. 따라서 이 그룹을 근거로 차량번호판 영역을 추출할 때, 우선 대상 차량의 번호판이 1줄 혹은 2줄 번호판의 조건을 갖는지 판단해야 한다. Fig. 8의 (a)와 같이 좌측으로 확장하여 1줄 번호판의 여부를 판단한다. 여기서 1줄 번호판의 조건이 만족되지 않으면 Fig. 8 (b)의 조건을 판단해서 2줄 번호판 여부를 결정한다.

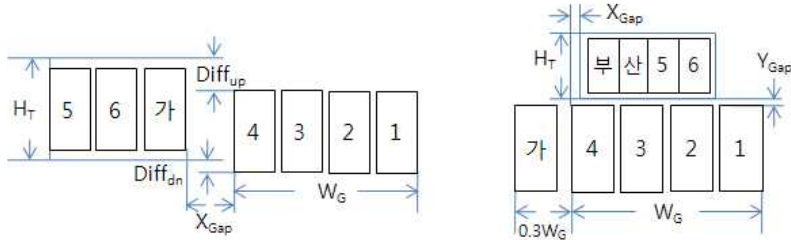


Fig. 8. License plate conditions (a) One line license plate, (b) Two lines license plate,

번호판의 조건식 (6)과 (7)를 만족하면 1줄 번호판으로 판단해서 왼쪽으로 확장시켜 전체 번호판 영역을 설정한다. 한편 식 (6)을 만족하지 않고 식 (8)를 만족하면 2줄 번호판으로 판단해서 전체 번호판 영역을 설정한다.

$$Diff_{up} \leq \max(15, H_T * 0.4) \ \& \ Diff_{dn} \leq \max(15, H_T * 0.4) \tag{6}$$

$$(W_G / \text{내부 사각형 수}) \leq X_{Gap} \leq 3 * (W_G / \text{내부 사각형 수}) \tag{7}$$

$$Y_{Gap} \leq H_T * 0.2 \tag{8}$$

2.7 추출된 차량 번호판의 전송

원 영상에서 추출된 차량번호판영역의 영상은 모듈의 와이파이 동글을 통하여 와이파이 다이렉트 통신으로 서버로 전송한다. 본 논문에서 제안하는 번호판 추출 모듈을 사용한 LPR 시스템은 1대의 PC 서버와 다수의 모듈로 구성되므로 모듈의 노드 식별번호를 번호판영역 영상과 함께 전송한다. 서버는 이 영상을 처리해서 번호판을 인식할 수 있고, 그 결과를 전송된 번호판과 함께 데이터베이스로 관리해서 미등록 차량의 검색, 주차 요금 산정 등 필요한 용도로 활용할 수 있다.

3. 번호판 추출 모듈의 구성

번호판 추출 모듈은 라즈베리파이 기반의 Linux 계열의 Raspbian OS에서 동작되게 했다. 각종 영상 처리에는 OpenCV 2.4에서 제공되는 라이브러리 함수를 활용하였다[5]. 카메라와 와이파이 동글을 USB에 연결해서, 차량번호판 영역 추출 모듈을 간단하게 구성하였다.

3.1 Linux계열의 Raspbian OS의 설치

라즈베리 파이에서 사용가능한 운영체제는 여러 가지가 있지만 본 모듈에서는 Raspbian OS를 설치하였다[11]. 여기에는 기본으로 GCC 컴파일러를 포함하므로 별도의 통합개발환경이 필요 없다. PC에서 SD Card로 다운받은 Raspbian OS 이미지를 설치해서 사용한다. SD Card를 라즈베리파이에 삽입하고 모니터, 키보드, 마우스를 연결한 후, 전원을 켜면 부팅된 후 초기설정 화면이 나타난다.

라즈베리파이의 기본 아이디는 pi이며, 패스워드는 raspberry이다. 초기설정을 완료하고 난 후 “startx” 명령어를 입력하면 Raspbian OS의 GUI 화면으로 이동한다.

3.2 OpenCV 2.4의 설치

번호판 영역 추출 모듈에서 수행되는 각종 영상처리를 OpenCV2.4 라이브러리 함수를 사용하면 편리하다. Raspbian에 OpenCV를 설치하기 위해서는 여러 가지 단계의 절차를 거쳐야 한다. 약 8개의 명령을 실행해서 OpenCV를 실행하기 위한 요소들이 설치해야 한다. 마지막으로 OpenCV 설치파일을 다운로드 받아 압축을 풀고, “cmake”명령어를 통하여 실행파일을 만든 후 설치한다.

3.3 와이파이 다이렉트

모듈과 서버간의 무선통신을 위해서 USB에 연결되는 와이파이 동글을 사용했다. 와이파이 다이렉트는 기존의 유선망을 통하지 않고 송수신기 간에 비교적 먼 거리의 고속 전송이 가능하고 비용이 저렴한 장점이 있다. 본 논문에서는 핫스팟(hot spot)기능을 사용하여 서버와 차량번호판 추출 모듈 간에 다이렉트로 무선통신을 수행하였다.

서버에서는 현재 몇 대의 모듈이 연결하였는지 확인할 수 있다. “netsh wlan show hostednetwork” 명령어를 실행하면 연결 상태를 확인할 수 있다. 와이파이 다이렉트 통신은 1 : N 통신이 가능하다. 따라서 LPR 시스템의 규모에 따라 모듈을 추가 설치 및 제거 등 관리가 쉬우므로 본 논문에서 제안하는 차량번호판 인식시스템에 적합하다.

4. 결과 및 고찰

4.1 번호판 추출 모듈 및 서버

차량 번호판 추출 모듈은 라즈베리 파이에 카메라와 와이파이 동글이 USB에 연결된 하드웨어 기반을 갖는다. 영상처리 및 전송 소프트웨어는 C++로 작성했다. 그레이 영상 변환, 가우시안 필터링, 에지 검출을 위한 캐니 알고리즘, 레이블링 등의 영상 처리는 OpenCV 2.4에서 제공되는 라이브러리함수를 사용했다.

영상처리의 각 단계에서 처리 시간은 Table 2와 같았다. OCR인식 소요시간이 나머지 전 과정보다 더 소요된 1.20초로 나타났다. 이것은 번호판 내의 4개의 숫자를 탐색하는 과정에서 선택한 레이블이 숫자인지 여부를 판단하기 위해서 Tesseract-OCR [12]를 사용했기 때문이다. 여기서는 숫자뿐만 아니라 문자 등을 모두 비교하므로 소요시간이 증가되었다. 그러나 전용의 신경망이나 퍼펙트 매트릭스(perfect matrix) 인식 등을 사용하면 처리시간은 크게 개선될 것으로 판단한다.

Table 2, Image processing time in each stage for license plate extraction

| Image Processing | Time(sec) |
|------------------------------------|-----------|
| Image Load | 0.15 |
| RGB to Gray Image | 0.03 |
| Gaussian Filter | 0.01 |
| Canny Edge Detection | 0.04 |
| Labeling and Filtering | 0.01 |
| Grouping Labels for 4 Numbers | 0.06 |
| OCR Recognition | 1.20 |
| Plate Type Decision and Extraction | 0.01 |
| Transfer Plate to Server | 0.20 |
| Total Processing Time | 1.51 |

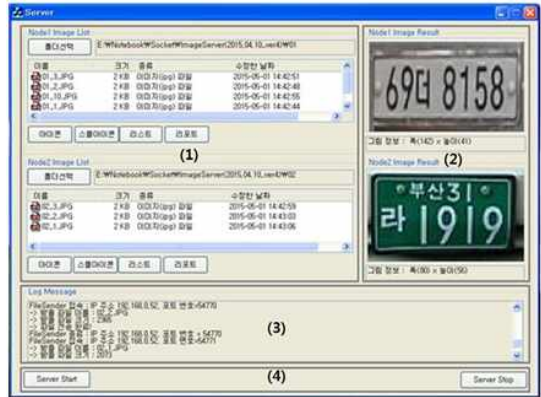


Fig. 9. License plates transferred from the proposed modules using wifi direct in server.

실험에 사용된 서버 노트북은 인텔 펜티엄 듀얼코어 i3-4025U, Windows 7, 4GB RAM이다. 추출된 번호판 영상 및 관련 정보를 와이파이로 수신하고 관리를 하기 위한 프로그램은 MFC Visual C++로 작성했다. 2개의 번호판 추출 모듈에서 무선으로 서버에 전송되는 상태를 Fig. 9와 같이 표시해서 정상적인 동작을 확인하였다.

본 연구의 결과를 검증하기 위해서 640 × 480 크기의 다양한 영상 500개를 사용하였다. 이 중에서 단지 8개 영상에서 자동차 번호판 영역 추출이 실패했지만, 98.4%의 높은 성공률을 보였다. 번호판 영역 추출이 실패한 경우는 밝은 조명과 반사로 인하여 에지의 추출이 부실한 경우이다. 이러한 문제점들은 실제의 주차 관리 시스템을 설치할 때 카메라의 위치와 조명을 적합하게 설정하면 해결할 수 있을 것으로 판단된다.

5. 결론

본 논문에서는 라즈베리파이를 이용하여 차량 번호판 영역을 추출하여 서버로 전송하는 모듈을 설계 및 구현하였다. 국내의 모든 차량 번호판에는 4개의 숫자를 포함하므로, 먼저 이 4개의 숫자들의 영역을 찾는다. 처리 과정은 그레이 영상에서 필터링을 수행한 후 에지 영상을 구한 후, 레이블링을 수행한다. 결과의 레이블들을 단계적으로 번호판의 숫자 후보 조건식을 설정해서 조건에 위배되면 탈락시키는 방식으로 최종 4개의 숫자를 포함하는 영역을 구했다. 구해진 숫자 영역을 근거로 좌측과 상측의 레이블들

을 조사해서 번호판 형태를 결정하고 전체 번호판 영역을 구했다. 결과의 번호판 영역은 와이파이 디렉트로 서버에 전송했다.

실험에서 640×480 크기의 다양한 500개의 영상을 테스트한 결과 98.4%의 인식률을 보였다. 실패한 경우는 밝은 조명과 불빛의 반사로 인한 문제였다. 실제의 주차관리시스템을 현장에 설치할 때 카메라의 위치와 조명을 적절하게 설정하면 이 문제는 해결될 것으로 판단된다.

본 논문에서 설계 및 구현한 모듈을 차량의 출입 게이트가 다수인 주차관리시스템에 사용하면 분산 처리로 인한 시스템의 처리속도를 향상시킬 수 있으며, 모듈의 추가 및 제거가 용이하므로 시스템의 설치 및 운용의 편의를 제공할 수 있다.

REFERENCE

- [1] J. Kim, Y. Han, and H. Hahn, "Number Plate Detection using Topology of Characters and Outer Contour," *Proceeding of the Summer Conference of The Institute of Electronics and Information Engineers*, Vol. 31, No. 1, pp. 1037-1038, 2008.
- [2] Y. Park, "A License Plate Extraction using Wavelet Transform," *Journal of Korean Institute of Information Technology*, Vol. 6, No. 2, pp. 91-99, 2008.
- [3] H. Oh and J. Choi, "A Recognition Algorithm of Car License Plate using Edge Projection and Directivity Vector," *Journal of Korean Institute of Information Technology*, Vol. 7, No. 1, pp. 83-92, 2009.
- [4] S. Park, J. Kim, S. Cho, S. Jung, and K. Lee, "A Recognition Algorithm of Car License Plate using Canny Edge Detection and CLNF Algorithm," *Proceeding of the Spring Conference of Korean Institute of Intelligent Systems*, Vol. 21, No. 1, pp. 39-41, 2011.
- [5] S. Kim, H. Cho, J. Min, and J. Choi, "A License Plate Extraction Algorithm using Brightness Vector," *Journal of Korean Institute of Information Scientists and Engineers*, Vol. 25, No. 4, pp. 676-684, 1998.
- [6] M. Kim, "A License Plate Detection Method Using Multiple-Color Model and Character Layout Information in Complex Background," *Journal of Korea Multimedia Society*, Vol. 11, No. 11, pp.1515-1524, 2008,
- [7] U. Chang, Y. Song, and Y. Kim, "The Extraction of a Car License Plate using the Color Information and Linear Regression Method," *Proceeding of the Fall Conference of the Korea Contents Association*, Vol. 1, No. 2, pp. 218-222, 2003.
- [8] D. Lee and I. Oh, "Car License Plate Extraction based on Numeral Recognition," *Journal of Korea Intelligence Transportation System Society*, Vol. 7, No. 1, pp. 59-67, 2008.
- [9] J. Kim, "Distortion Invariant Vehicle License Plate Extraction and Recognition Algorithm," *The Journal of the Korea Contents Association*, Vol. 11, No. 3, pp. 1-8, 2011.
- [10] S. Hwang, *Learning OpenCV(Korean translated)*, Hanbit Media, Seoul, 2010.
- [11] Raspbian OS Installation for Raspberry Pi, <https://www.raspberrypi.org/downloads/> (accessed June, 24, 2015).
- [12] Anthony Kay, "Optical Character Recognition by Open Source OCR Tool Tesseract: A Case Study," *International Journal of Computer Applications*, Vol. 55, No. 10, pp. 50-56, 2012.



김 동 경

2010년 2월 동명대학교 컴퓨터공
학과 공학사
2015년 8월 부경대학교 컴퓨터공
학과 공학석사
현재 (주)오토닉스 제어계측연구
소 주임연구원

관심분야: 임베디드시스템, 영상처리, IoT(Internet of Things)



우 종 호

1978년 2월 경북대학교 컴퓨터공
학과 학사
1981년 2월 경북대학교 전자공학
과 컴퓨터공학전공 석사
1990년 2월 경북대학교 전자공학
과 컴퓨터공학전공 박사

1981년 3월~현재 부경대학교 컴퓨터공학과 교수
관심분야: Embedded System, Image Processing, IoT,
Machine Learning System