디지털산업정보학회 논문지 제11권 제3호-2015년 9월
http://dx.doi.org/10.17662/ksdim.2015.11.3.001

MEAN Stack 기반의 컴퓨터 비전 플랫폼 설계*

홍 선 학** · 조 경 순*** · 윤 진 섭***

Computer Vision Platform Design with MEAN Stack Basis

Hong Seonhack · Cho Kyungsoon · Yun Jinseob

<Abstract>

In this paper, we implemented the computer vision platform design with MEAN Stack through Raspberry PI 2 model which is an open source platform. we experimented the face recognition, temperature and humidity sensor data logging with WiFi communication under Raspberry Pi 2 model. Especially we directly made the shape of platform with 3D printing design. In this paper, we used the face recognition algorithm with OpenCV software through haarcascade feature extraction machine learning algorithm, and extended the functionality of wireless communication function ability with Bluetooth technology for the purpose of making Android Mobile devices interface. And therefore we implemented the functions of the vision platform for identifying the face recognition characteristics of scanning with PI camera with gathering the temperature and humidity sensor data under IoT environment. and made the vision platform with 3D printing technology. Especially we used MongoDB for developing the performance of vision platform because the MongoDB is more akin to working with objects in a programming language than what we know of as a database. Afterwards, we would enhance the performance of vision platform for clouding functionalities.

Key Words : Vision Platform, Face Recognition, MongoDB, MEAN Stack, Raspberry PI

Ⅰ. 서론

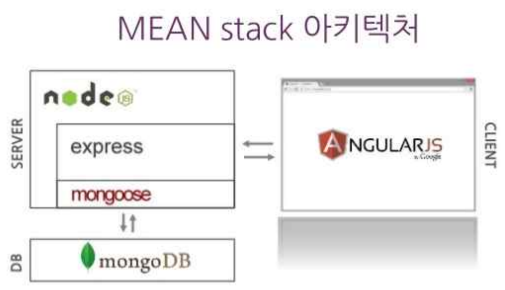
MEAN is an opinionated fullstack javascript framework for an easy starting point with MongoDB, Node. js, Express, and AngularJS based applications. It is designed to give us a quick and organized way to start developing MEAN based web Apps with useful modules like Mongoose and Passport pre-bundled and configured. In this paper, we mainly try to take care of the connection points between existing popular frameworks and solve common integration efforts for face recognition fields with OpenCV under MEAN stack.

MongoDB is the leading NoSQL database with empowering businesses to be more agile and

*본 연구는 2015학년도 서일대학 교내학술연구비지원으로 수행되었음
** 서일대학교 컴퓨터전자과 교수
*** 서일대학교 컴퓨터응용과 교수

scalable. Express is a minimal and flexible node. js web application framework, providing a robust set of features for building single and multi-page, and hybrid web applications. AngularJS lets us extend HTML vocabulary for your application. The resulting environment is extraordinarily expressive, readable, and quick to develop. And therefore Node. js is a platform built on Chrome's JavaScript runtime for easily building fast, scalable network applications.

In this paper, we used the Raspberry Pi 2 Model and PI camera module, with at least an 16 GB SD card, and the Raspbian operating system because it has the advantage of cost and performance respects against other platforms. <Figure 1> showed the overview of MEAN stack architecture system which had an server including node, express, and mongoose, an client named AngularJs and Mongo Database for developing the face recognition under the Linux operating system.



<Fig 1> Overview of MEAN stack system

Especially, we would make MEAN stack helpful for developing the easy made computer vision with OpenCV and realtime processing through Bluetooth wireless technology under the IoT environment [1-3].

Ⅱ. Basic Theory

2.1 OpenCV Modular Structure

In this paper, we used the OpenCV software for face detection and recognition algorithm under Raspberry operating system, Raspbian. It has an open-source BSD-licensed library that includes several hundreds of computer vision algorithms. OpenCV has a modular structure, which means that package includes several shared or static libraries. We downloaded the opencv-2.4.8. zip from git clone https://github.com/Itseez/opencv.git after updating, upgrading and installing dependencies.

2.2 Raspberry Pi Features

The Raspberry Pi 2 Model is a credit-card sized computer that plugs into our monitor and a keyboard. It is a capable little computer which can be used in electronic projects, and for many of the things that our desktop PC does, like spreadsheets, word-processing and games. It also plays high-definition video.

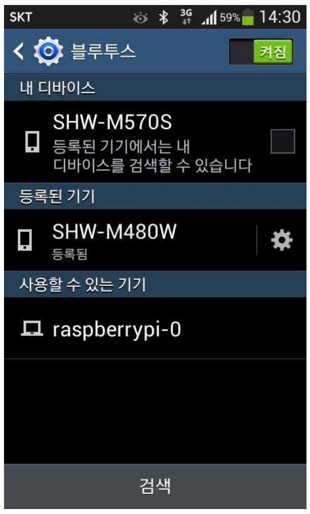
We used the Pi 2 model which has the 900MHz quad-core ARM Cortex-A7 processor. It runs the same software, and still has 1GB RAM with Camera interface. We used the most common Raspberry Pi distribution open source operating system "Raspbian" which is the "Debian" distribution was configured and because it has an ARMv7 processor, it can run the full range of ARM GNU/Linux distributions, including Snappy Ubuntu Core, as well as Microsoft Windows 10.

The Raspberry Pi camera module could be used to take high-definition video, as well as stills photographs. It's easy to use for beginners, but has plenty to offer advanced users if we're looking to expand our knowledge. There are lots of examples online of people using it for time-lapse, slow-motion and other video cleverness. we could also use the libraries that bundle with the camera to create effects.

As we're interested in the nitty-gritty, we would use the module that has a five megapixel fixed-focus camera which supports 1080p30,720p60 and VGA90 video modes, as well as stills capture. It attaches via a 15cm ribbon cable to the CSI port on the Raspberry Pi. The camera works with all models of Raspberry Pi 1 and 2. It can be accessed through the MMAL and V4L APIs, and there are numerous third-party libraries built for it, including the Picamera Python library.

We used an official and recommended universal Bluetooth dongle for Raspberry Pi. In our tests, we used the CSR 4.0 dongle adapter which has 3Mbps speed, 20 meters working range, wirelessly connect to Bluetooth devices, and plug and play functions.

<Figure 2> shows the Bluetooth connection windows. Click on the available device button and move to the registered device on the window to pair between mobile and Raspberry Pi. We could connect or disconnect using the buttons. Notice also the IP address of the Pi is shown at the bottom of the window[4-6].



<Fig 2> Bluetooth connection of Raspberry Pi

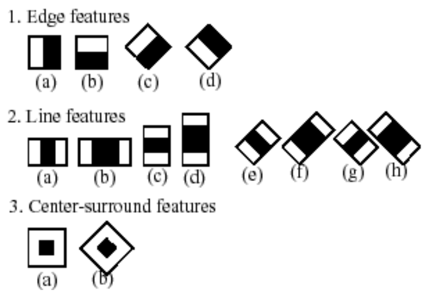
2.3 Cascade Classification

In this paper, we used the Haar Feature-based Cascade Classifier for Object Detection. The object detector described below has been initially proposed by Paul Viola and improved by Rainer Lienhart. Rainer Lienhart and Jochen Maydt.

First, a classifier (namely a cascade of boosted classifiers working with haar-like features) is trained with a few hundred sample views of a particular object (i.e., a face), called positive examples, that are scaled to the same size (say, 100x100), and negative examples - arbitrary images of the same size.

After a classifier is trained, it can be applied to a region of interest (of the same size as used during the training) in an input image. The classifier outputs a "1" if the region is likely to show the object of face, and "0" otherwise. To search for the face in the whole image one can move the search window across the image and check every location using the classifier. The classifier is designed so that it can be easily "resized" in order to be able to find the faces of interest at different sizes, which is more efficient than resizing the image itself. So, to find an face of an unknown size in the image the scan procedure should be done several times at different scales.

The word "cascade" in the classifier name means that the resultant classifier consists of several simpler classifiers (stages) that are applied subsequently to a region of interest until at some stage the candidate is rejected or all the stages are passed. The word "boosted" means that the classifiers at every stage of the cascade are complex themselves and they are built out of basic classifiers using one of four different boosting techniques (weighted voting). Currently Discrete Adaboost, Real Adaboost, Gentle Adaboost and Logitboost are supported. Paul Viola and Michael J. Jones. Rapid Object Detection using a Boosted Cascade of Simple Features. IEEE CVPR, 2001.[7] The basic classifiers are decision tree classifiers with at least 2 leaves. Haarcascade features are the input to the basic classifiers, and are calculated as described below. In this paper, we used the following Haarcascade features such as <Figure 3>.



<Fig 3> Overview of Haarcascade feature

Ⅲ. Features of Platform

In this paper, we used the Pi CAMERA BOARD plugs directly into the CSI connector on the Raspberry Pi. It's able to deliver a crystal clear 5MP resolution image, or 1080p HD video recording at 30fps with latest v1.3. Board features a 5MP (2592 × 1944 pixels) Omnivision 5647 sensor in a fixed focus module.

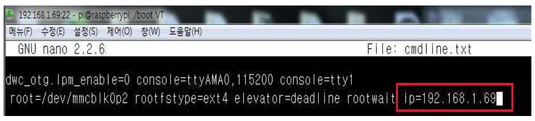
The module attaches to Raspberry Pi, by way of a 15 pin Ribbon Cable, to the dedicated 15 pin MIPI Camera Serial Interface (CSI), which was designed especially for interfacing to cameras. The CSI bus is capable of extremely high data rates, and it exclusively carries pixel data to the BCM2835 processor[7-9].

3.1 Physical Platform

The Raspberry Pi 2 model contains the interrupt controller, GPIO, UAB, PCM/I2S. DMA controller, I2C master, I2C/SPI slave, SPI1/2/3, PWM and UART0, UART1 peripherals. In addition to the ARM's MMU, BCM2835 includes a second coarse-grained MMU for mapping ARM physical addresses onto system bus addresses.

<Figure 4> showed the bluetooth setup of Raspberry PI. We need to add "ip=192.168. 1.69" at

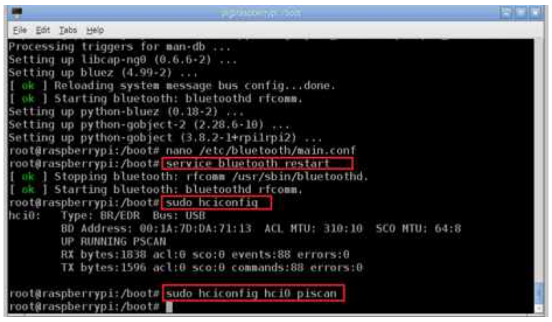
the end of boot/cmdline. txt which passed to the parameters to the kernel on boot. If you may have different IP address, you should alter the exact IP address to adjust the wireless communication abilities.


192.168.1.69:22 - pi@raspberrypi: /boot VT
메뉴(F) 수정(E) 설정(S) 제어(O) 창(W) 도움말(H)

```
  GNU nano 2.2.6                                    File: cmdline.txt

dwc_otg.lpm_enable=0 console=ttyAMA0,115200 console=tty1
 root=/dev/mmcblk0p2 rootfstype=ext4 elevator=deadline rootwait ip=192.168.1.69
```

<Fig 4> Bluetooth Setup on boot mode

<Figure 5> showed the procedure of Bluetooth service. At first we restart the Bluetooth service with < ~$sudo service bluetooth restart >. If everything has been successful we should be able to get information about the installed Bluetooth dongle with below command which is <~$sudo hciconfig >. To make the device discoverable we also need to type <~$sudo hciconfig hci0 pisacan >.



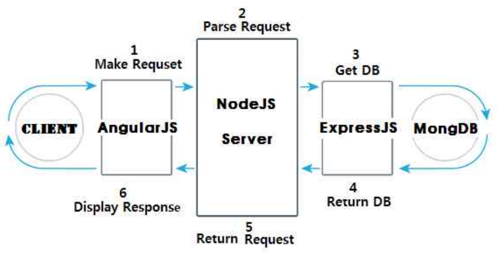
```
Processing triggers for man-db ...
Setting up libcap-ng0 (0.6.6-2) ...
Setting up bluez (4.99-2) ...
[ ok ] Reloading system message bus config...done.
[ ok ] Starting bluetooth: bluetoothd rfcomm.
Setting up python-bluez (0.18-2) ...
Setting up python-gobject-2 (2.28.6-10) ...
Setting up python-gobject (3.8.2-1+rpi1rpi2) ...
root@raspberrypi:/boot# nano /etc/bluetooth/main.conf
root@raspberrypi:/boot# service bluetooth restart
[ ok ] Stopping bluetooth: rfcomm /usr/sbin/bluetoothd.
[ ok ] Starting bluetooth: bluetoothd rfcomm.
root@raspberrypi:/boot# sudo hciconfig
hci0:   Type: BR/EDR  Bus: USB
        BD Address: 00:1A:7D:DA:71:13  ACL MTU: 310:10  SCO MTU: 64:8
        UP RUNNING PSCAN
        RX bytes:1838 acl:0 sco:0 events:88 errors:0
        TX bytes:1596 acl:0 sco:0 commands:88 errors:0

root@raspberrypi:/boot# sudo hciconfig hci0 piscan
root@raspberrypi:/boot# 
```

<Fig 5> Procedure of Bluetooth Setup

3.2 Logical Platform

We used the MongoDB which is a cross-platform document-oriented database. Classified as a NoSQL database, MongoDB eschews the traditional table-based relational database structure in favor of JSON-like documents with dynamic schemas (MongoDB calls the format BSON), making the integration of data in certain types of applications easier and faster. Released under a combination of the GNU Affero General Public License and the Apache License, MongoDB is free and open-source software[10-12].

<Figure 6> represent the full stack operation of MEAN stack with each section individually acting request and response of web service.



<Fig 6> Feature of MEAN stack

Express is a minimal and flexible Node. js web application framework that provides a robust set of features for web and mobile applications.

Express is a routing and middleware web framework with minimal functionality of its own: An Express application is essentially a series of middleware calls. Middleware is a function with access to the request object (req), the response object (res), and the next middleware in the application’s request- response cycle, commonly denoted by a variable named next.

AngularJS is a toolset for building the framework most suited to your application development. It is fully extensible and works well with other libraries.

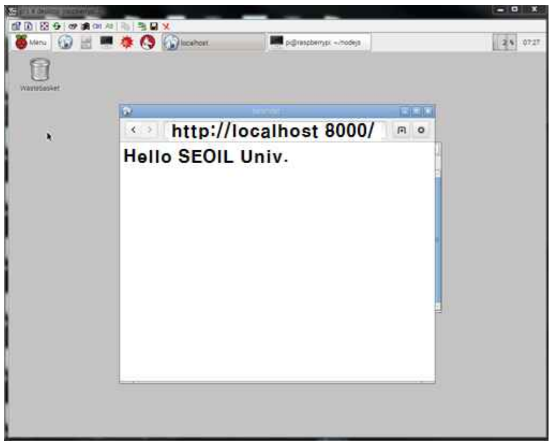
Node. js is a platform built on Chrome's JavaScript runtime for easily building fast, scalable network applications. Node. js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.

<Figure 7> shows the source code of app. js to make web server. In the terminal, let's node app. js execute under the GUI, connect to the http://localhost:8000, and then display the result on the browser window.

```
var http = require('http');

var server = http. createServer(function (req, res){
  res. writeHead(200, { 'Content-Type' : 'text/plain'});
  res. end('Hello SEOIL Univ.');
});
server. listen(8000);
```

<Fig 7> Source code of app.js

<Figure 8> shows the execution of NodeJS under Linux Ubuntu environments.

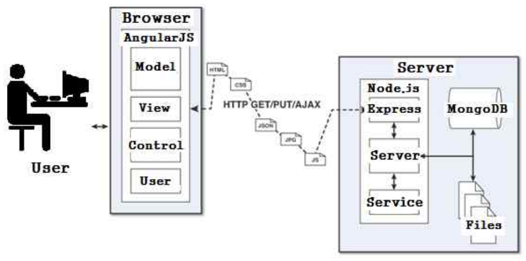


```
http://localhost 8000/
Hello SEOIL Univ.
```

<Fig 8> Execution of NodeJS

3.3 MongoDB Platform

In this paper, we used an open-source database MongoDB that provides high performance, high availability and automatic scaling. MongoDB obviates the need for an Object Relational Mapping(ORM) to facilitate development. A record in MongoDB is a document, which is a data structure composed of field and value pairs. MongoDB documents are similar to JSON objects. The values of fields may include other documents, arrays, and arrays of documents. <Figure 9> shows the architecture of MEAN stack structure[13-14].



<Fig 9> Architecture of MEAN Stack

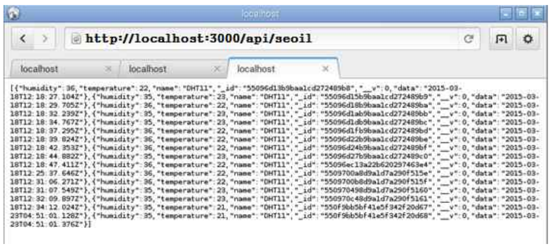
MongoDB stores documents in collections. Collections are analogous to tables in relational databases. Unlike a table, however, a collection does not require its documents to have the same schema. In MongoDB, documents stored in a collection must have a unique _id field that acts as a primary key.

Ⅳ. Platform Feasibility Design

In this section, we described the characteristics of computer vision platform. The basic part of platform is that an Raspberry Pi, OpenCV software and MEAN Stack with MongoDB. And with Analog sensor modules, temperature and humidity sensors for gathering environmental data and for displaying real time information via the bluetooth communication module[15-17].

4.1 Interface platform architecture

The proposed platform took a modular type for accumulating the individual modules in order to extend the abilities. <Figure 10> showed the data of interface platform with MongoDB execution.



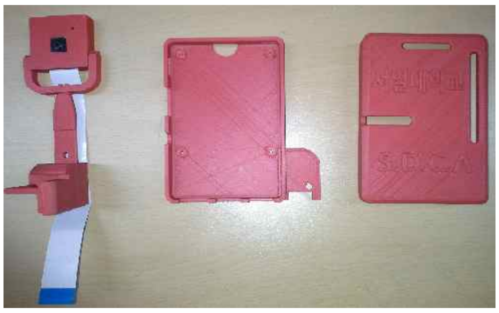
```
[{"humidity":36,"temperature":22,"name":"DHT11","_id":"55096d13b9baa1cd272489b8","__v":0,"data":"2015-03-
18T12:18:27.104Z"},{"humidity":35,"temperature":23,"name":"DHT11","_id":"55096d15b9baa1cd272489b9","__v":0,"data":"2015-03-
18T12:18:29.705Z"},{"humidity":36,"temperature":22,"name":"DHT11","_id":"55096d18b9baa1cd272489ba","__v":0,"data":"2015-03-
18T12:18:32.239Z"},{"humidity":35,"temperature":23,"name":"DHT11","_id":"55096d1ab9baa1cd272489bb","__v":0,"data":"2015-03-
18T12:18:34.767Z"},{"humidity":35,"temperature":23,"name":"DHT11","_id":"55096d1db9baa1cd272489bc","__v":0,"data":"2015-03-
18T12:18:37.295Z"},{"humidity":36,"temperature":22,"name":"DHT11","_id":"55096d1fb9baa1cd272489bd","__v":0,"data":"2015-03-
18T12:18:39.824Z"},{"humidity":36,"temperature":22,"name":"DHT11","_id":"55096d22b9baa1cd272489be","__v":0,"data":"2015-03-
18T12:18:42.353Z"},{"humidity":36,"temperature":22,"name":"DHT11","_id":"55096d24b9baa1cd272489bf","__v":0,"data":"2015-03-
18T12:18:44.882Z"},{"humidity":35,"temperature":23,"name":"DHT11","_id":"55096d27b9baa1cd272489c0","__v":0,"data":"2015-03-
18T12:18:47.411Z"},{"humidity":36,"temperature":22,"name":"DHT11","_id":"55096ec13a22b620297463e4","__v":0,"data":"2015-03-
18T12:25:37.646Z"},{"humidity":36,"temperature":22,"name":"DHT11","_id":"5509700a8d9a1d7a290f515e","__v":0,"data":"2015-03-
18T12:31:06.271Z"},{"humidity":36,"temperature":22,"name":"DHT11","_id":"5509700b8d9a1d7a290f515f","__v":0,"data":"2015-03-
18T12:31:07.549Z"},{"humidity":35,"temperature":23,"name":"DHT11","_id":"550970498d9a1d7a290f5160","__v":0,"data":"2015-03-
18T12:32:09.897Z"},{"humidity":35,"temperature":23,"name":"DHT11","_id":"550970c48d9a1d7a290f5161","__v":0,"data":"2015-03-
18T12:34:12.024Z"},{"humidity":35,"temperature":21,"name":"DHT11","_id":"550f9bb5bf41e5f342f20d67","__v":0,"data":"2015-03-
23T04:51:01.128Z"},{"humidity":35,"temperature":21,"name":"DHT11","_id":"550f9bb5bf41e5f342f20d68","__v":0,"data":"2015-03-
23T04:51:01.376Z"}]
```

<Fig 10> MongoDB Data of Interface Platform

We would designate the basic platform to be an Raspberry Pi 2 module, the B/T wireless communication to be an channel for data logging, the Pi-Camera module to be for gathering the scanning image in order to recognize the face matching, the temperature and humidity sensor for collecting the environmental data[18-19].

4.2 Design layout

In this paper, We made the vision platform with 3D printing technology like <Figure 11>. There were temperature and humidity sensors around the Pi-Camera for taking a picture and recoding an image with real time, and bluetooth communicating Android mobile device. <Figure 11> showed the 3D design of platform.



<Fig 11> 3D Design of Platform

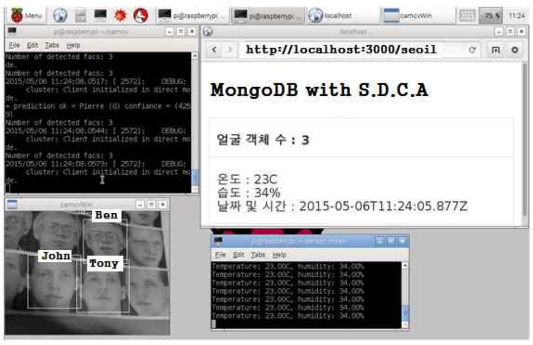
4.3 Experiments for Vision Platform

We experimented the computer vision platform with Bluetooth communication through MongoDB of MEAN Stack. The temperature and humidity sensor attached on platform measured the environmental data every specified period with MongoDB database for logging in centigrade unit[20-21].

<Figure 12> represented the face recognition and matching result of Raspberry Pi camera captured image with OpenCV haarcascade feature detecting algorithm.

Ⅴ. Conclusions

In this paper, we implemented the face recognition, temperature and humidity sensor data



```
Number of detected facs: 3
de.
Number of detected facs: 3
2015/05/06 11:24:08.0517: [ 2572]:    DEBUG:
    cluster: Client initialized in direct mo
de.
+ prediction ok = Pierre (0) confiance = (425
9)
Number of detected facs: 3
2015/05/06 11:24:08.0544: [ 2572]:    DEBUG:
    cluster: Client initialized in direct mo
de.
Number of detected facs: 3
2015/05/06 11:24:08.0573: [ 2572]:    DEBUG:
    cluster: Client initialized in direct mo
de.
```

http://localhost:3000/seoil

MongoDB with S.D.C.A

얼굴 객체 수 : 3

온도 : 23C
습도 : 34%
날짜 및 시간 : 2015-05-06T11:24:05.877Z

```
Temperature: 23.00C, humidity: 34.00%
Temperature: 23.00C, humidity: 34.00%
Temperature: 23.00C, humidity: 34.00%
Temperature: 23.00C, humidity: 34.00%
Temperature: 23.00C, humidity: 34.00%
Temperature: 23.00C, humidity: 34.00%
Temperature: 23.00C, humidity: 34.00%
```

<Fig 12> Face Recognition with MongoDB

logging with WiFi communication under Raspberry Pi 2 model platform. Especially we directly designed and made the shape of platform with 3D printing technology. Afterwards, we would enhance the performance of vision platform for clouding functionalities.

참고문헌

[1] 박재호, MEAN 스택을 사용한 모던 웹 개발 입문, 에이콘, 2015.
[2] 손병대, Node Web Development, 거침없이 배우는 라즈베리 파이, 에이콘, 2011.
[3] https://nodejs. org/documentation/Tutorials
[4] 배장열, 파이선으로 시작하는 라즈베리 파이 프로그래밍, 제이펍, 2013.
[5] Leo Breiman, "Random Forest, Machine Learning," vol. 45, 2001, pp.5~32.
[6] Paul Viola and Michael J. Jones, "Robust real-time face detection," International Journal of Computer Vision, Vol. 57, No. 2, 2004, pp. 137~154.
[7] http://research. microsoft. com/en-us/um/people/viola/Pubs/Detect/violaJones_CVPR2001.pdf.
[8] 오일석, 컴퓨터 비전-기본개념부터 최신 모바일 응용예까지, 한빛 아카데미, 2014.
[9] https://docs. mongodb. org/getting-started/shell/introduction/,2015.
[10] 홍선학, 이아리, "모바일 아두이노 임베디드 플랫폼 설계," 디지털산업정보학회논문지, 제9권, 제4호, 2013, pp. 33-41.
[11] 홍선학, 남궁일주, "안드로이드 리얼 타깃 포팅 응용 소프트웨어 개발," 디지털산업정보학회논문지, 제7권, 제3호, 2011.
[12] 홍선학, 조경순, "히어캠 임베디드 플랫폼 설계," 디지털산업정보학회논문지, 제10권, 제4호, 2014, pp. 79-87.
[13] 몽고DB 사이트, http://docs. mongodb. org/manual/, 2015.
[14] 마이클 맥티어, 까예하스, 조효성옮김, 안드로이드 애플리케이션 개발, 에이콘출판사, 2014.
[15] 홍선학, 모바일로 배우는 아두이노 따라하기, 성안당, 2013.
[16] Wolfgang Mauerer, Professional Linux Kernel architecture, Wiely Pub, 2007.
[17] Downey, Allen B, Python for software Design, Cambridge, 2009.
[18] 사이먼 몽크, 라즈베리 파이 쿡북, 한빛 미디어, 2015.
[19] 로버트 라가니에, OpenCV 2를 활용한 컴퓨터 비전 프로그래밍, 에이콘 출판사, 2012.
[20] Computer Bookshops, Opencv Computer Vision Application Programming Cookbook (2nd Edition), Packet Pub, 2013.
[21] 드류 콘웨이, 존 마일즈 하이트, 김진홍, 해커스타

일로 배우는 기계학습, 인사이트, 2014.

■ 저자소개 ■

홍 선 학
(Hong Seonhack)

1992년~현재
서일대학교 컴퓨터전자과 교수
1994년 광운대학교 대학원 박사 졸업

관심분야 : 제어 응용, 임베디드 리눅스
E-Mail : hongsh@seoil.ac.kr

조 경 순
(Cho Kyoungsoon)

1998년~현재
서일대학교 컴퓨터응용과 교수
2008년 광운대 대학원 박사 졸업

관심분야 : 제어 응용, 전기전자재료
E-mail : kscho@seoil.ac.kr

윤 진 섭
(Yun Jinseob)

1991년~현재
서일대학교 컴퓨터응용과 교수
2002년 동국대 대학원박사 졸업.

관심분야 : 정보기술, 반도체 설계
E-Mail : js22y@seoil.ac.kr

논문접수일: 2015년 8월 24일
수 정 일: 2015년 9월 1일
게재확정일: 2015년 9월 4일