

# 낙관 인식을 위한 애플리케이션의 설계 및 구현<sup>☆</sup>

## Design and Implementation of Luo-kuan Recognition Application

김 한 슬<sup>1</sup>                      서 귀 빈<sup>2</sup>                      강 민 구<sup>3</sup>                      류 기 수<sup>4</sup>                      홍 민\*  
Han-Syel Kim                      Kwi-Bin Seo                      Mingoo Kang                      Gee Soo Ryu                      Min Hong

### 요 약

대부분의 동양화 작품에는 작가의 정보를 압축시켜 하나의 그림으로 표현한 낙관이 존재하고 이러한 낙관은 작품의 제목이나 작가의 이름 등 다양한 정보를 포함하고 있다. 따라서 동양화를 수집하거나 즐기는 사람들에게 낙관은 동양화에 대한 중요한 정보를 제공하는 단서 역할을 한다. 하지만 낙관에 있는 글자들은 대부분 어려운 한자나 간자 혹은 다양한 모양으로 변형되어 있어 일반인들이 쉽게 해석하기 어려운 문제점이 있다. 본 논문에서는 낙관의 정보를 손쉽게 확인할 수 있도록 안드로이드 기반의 낙관 검색 애플리케이션을 개발하였다. 해당 애플리케이션은 촬영한 낙관 이미지를 분석하여 서버에 전송해 서버 내의 데이터베이스에서 촬영한 낙관 사진과 가장 유사한 낙관 후보에 대한 정보를 검색하는 알고리즘을 적용하였다. 또한 제안하는 알고리즘의 성능 분석을 위해서 촬영된 낙관 사진과 170개의 낙관 데이터 후보 중에서 정확하게 낙관을 찾아내는지에 대한 여부와 제공되는 낙관의 순위를 바탕으로 알고리즘의 정확도를 비교 및 분석하였다. 정확도 분석 실험 결과 본 애플리케이션의 검색 알고리즘의 정확도는 약 90%로 확인되었으며 추후 알고리즘의 최적화와 멀티쓰레딩 알고리즘의 보완을 통해 빅 데이터 환경에서 자동으로 이미지를 분석 및 검색하는 플랫폼으로의 발전이 가능할 것으로 기대한다.

☞ 주제어 : 낙관, 애플리케이션, 영상처리, 데이터베이스, 서버

### ABSTRACT

In oriental paintings, there is Luo-kuan that expressed in a single picture by compressing the artist's information. Such Luo-kuan includes various information such as the title of the work or the name of the artist. Therefore, information about Luo-kuan is considered important to those who collect or enjoy oriental paintings. However, most of the letters in the Luo-kuan are difficult kanji, kanzai, or various shapes, so it is difficult for the ordinary people to interpret. In this paper, we developed an Luo-kuan search application to easily check the information of the Luo-kuan. The application uses a search algorithm that analyzes the captured Luo-kuan image and sends it to the server to output information about the Luo-kuan candidates that are most similar to the Luo-kuan images taken from the database in the server. We also compared and analyzed the accuracy of the algorithm based on 170 Luo-kuan data in order to find out the ranking of the Luo-kuan that matched the Luo-kuan among the candidates. Accuracy Analysis Experimental Results The accuracy of the search algorithm of this application is confirmed to be about 90%, and it is anticipated that it will be possible to develop a platform to automatically analyze and search images in a big data environment by supplementing the optimizing algorithm and multi-threading algorithm.

☞ keyword : Luo-kuan(落款), Mobile Application, Image processing

## 1. 서 론

일반적으로 서양화 작품에는 작품의 작가를 알리기 위해 본인의 서명을 사용하여 왔다. 그러나 동양화 작품에는 작가의 정보를 압축시켜 그림으로 표현한 낙관이 널리 사용되어 왔다. 낙관이란 그림이나 글씨를 하나의 도장 형태로 표현하여 찍은 것으로 작품에 찍혀있는 낙관을 통해서 작품의 작가, 제작일, 제작 이유 등의 다양한 정보를 포함하고 있기 때문에 동양화에서는 정확한 작가를 판단하는데 매우 중요한 부분이라고 할 수 있다. 하지만 이러한 낙관은 (그림 1)에서 볼 수 있듯이 다양한 모양

<sup>1</sup> Department of Computer Software Engineering, Soonchunhyang University, Asan, 336-745, Korea.

<sup>2</sup> Department of Computer Science, Soonchunhyang University, Asan, 336-745, Korea.

<sup>3</sup> Department of IT Contents, Hanshin University, 01025, Osan, Korea.

<sup>4</sup> Department of Chinese Culture & Art Studies, Hanshin University, 01025, Osan, Korea.

\* Corresponding author (mhong@sch.ac.kr)

[Reviewed 14 November 2017, Reviewed 27 November 2017, Accepted 12 December 2017]

☆ 본 논문은 2017년도 한국인터넷정보학회 춘계학술발표대회 우수 논문 추천에 따라 확장 및 수정된 논문임.



(그림 1) 다양한 낙관의 모양  
(Figure 1) Various Shapes of Luo-kuan

으로 변형되어 사용되어 왔으며, 대부분이 알아보기 쉽지 않은 한문이나 도형으로 구성되어 있는 경우가 많다.

따라서 동양화를 감상하는 일반 사람들은 복잡하고 다양한 형태의 한문을 해석하기 어렵고 한문을 알아볼 수 없도록 그려진 낙관 때문에 작품에 대한 정보를 쉽게 파악할 수 없는 상황이다. 작품에 대한 다양한 정보를 알기 위해서는 일반적으로 낙관 및 동양화 전문가에게 해석을 요구하는 경우가 많다. 또한 최근 재테크와 작품 감상을 위한 소장용으로 동양화에 대한 관심이 증가함에 따라 다양한 낙관을 자동으로 해석해주는 애플리케이션에 대한 필요성이 증대되고 있다.

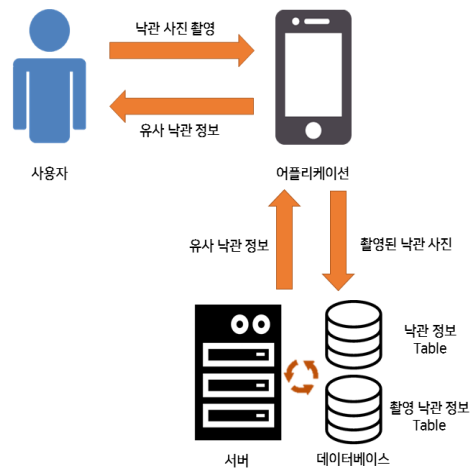
이에 따라 본 논문에서는 낙관 이미지를 분석해서 낙관의 정보를 제공하는 영상처리 기술에 대한 연구를 수행하였다. 본 연구의 결과로 개발된 안드로이드 스마트폰 애플리케이션으로 사진을 찍으면 해당 낙관 이미지의 특징들을 분석하여 촬영한 낙관과 가장 유사한 낙관을 검색하는 알고리즘을 구현하였다. 또한, 구현된 알고리즘의 정확도를 측정하기 위하여 다양한 낙관 이미지 입력 데이터를 바탕으로 판단된 결과에 대한 비교 및 분석을 수행하였다. 본 논문에서는 입력된 낙관 이미지에 대해서 서버에 저장되어 있는 데이터에서 가장 유사한 낙관을 찾아 그 낙관의 의미, 작가 등의 다양한 정보들을 제공하는 방법을 설계 및 구현하였다.

## 2. 낙관 인식 애플리케이션의 시스템 흐름도

### 2.1 클라이언트-서버 흐름도

제안된 낙관 인식 애플리케이션에 적용된 클라이언트와 서버 간의 데이터 흐름도는 (그림 2)와 같다[1]. 사용자는 애플리케이션을 통해 정보를 알고 싶은 낙관을 촬영한다. 촬영된 낙관 이미지는 애플리케이션에서 네트워크 통신을 통해 서버로 전송이 된다. 전송된 이미지는 데이터베이스에 저장되어 있는 낙관의 이미지들과의 비교

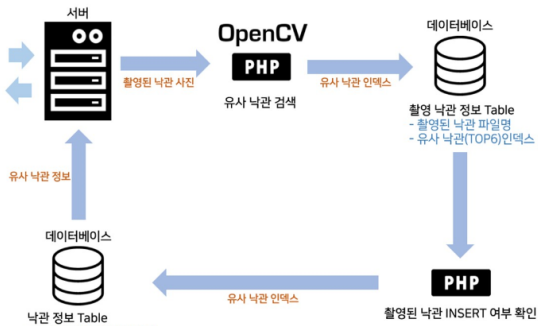
를 위해 서버 내의 데이터베이스에 저장된다. 서버의 데이터베이스에는 낙관의 이미지와 낙관과 관련된 모든 정보를 저장하고 있는 낙관 정보 Table과 촬영된 낙관의 관련된 정보를 저장하는 촬영 낙관 정보 Table로 구성된다. 이 데이터베이스의 Table을 이용하여 서버 내에서는 데이터 검색을 통해서 가장 유사한 낙관에 대한 정보를 찾는다. 만약, 데이터 검색을 통해서 유사한 낙관을 찾았다면, 그의 정보와 함께 사용자의 애플리케이션으로 전송하고, 사용자는 받은 데이터를 통해 유사 낙관에 대한 다양한 정보를 확인할 수 있다[2].



(그림 2) 애플리케이션 - 서버의 통신 흐름도  
(Figure 2) Flow Chart of Application - Server Communication

### 2.2 서버에서의 데이터 검색

애플리케이션을 통해 찾고자 하는 낙관의 이미지가 촬영되면, 서버 내에서 데이터 검색 시스템이 호출되며 데이터 검색 시스템의 흐름도는 (그림 3)과 같다. 데이터 검색 시스템은 위에서 설명한 바와 같이 애플리케이션을 통해 촬영된 사진이 서버로 전송되었을 때 호출된다. 서버는 사용자가 촬영한 낙관 이미지 데이터를 받아서 OpenCV를 통해 유사 낙관을 검색한다[3-4]. 검색된 유사 낙관은 유사도를 기준으로 상위 6개를 찾아서 촬영 낙관 정보 Table에 저장한다. 또한, 이들의 정보를 전송하기 위해서 PHP를 통해 낙관 정보 Table에 있는 유사 낙관 후보들의 관련 정보를 검색하여 검색된 정보들을 서버에서 네트워크 통신을 통해서 사용자의 애플리케이션으로 전송한다[5].



(그림 3) 데이터 검색 흐름도  
(Figure 3) Flow Chart of Data Search

### 3. 낙관의 유사도 검색 알고리즘

#### 3.1 알고리즘 구현

서버 내에서 수행되는 검색 알고리즘은 OpenCV의 Feature Matching을 통해 구현하였다. Feature Matching은 각 이미지의 수많은 피쳐(Feature)를 추출하고, 피쳐 간의 거리를 측정하여 두 피쳐의 거리가 일정 이하이면 같은 피쳐라고 판단하고 같은 피쳐의 개수를 찾는 방식이다.[6-8]. 본 논문에서 제안된 방식을 이용하여 사용자가 촬영한 이미지와 데이터베이스에 미리 저장한 낙관 이미지들의 피쳐를 비교해서 가장 유사한 낙관을 검색하였다 [9-10]. 기존의 낙관 이미지들은 이미 서버에 모두 저장되어 있는 상태이고, 사용자가 촬영하여 비교를 요청한 사진은 서버에 전송됨과 동시에 서버의 디렉토리에 저장한다. 저장된 사진의 파일명은 데이터베이스 Table에 저장되며 OpenCV에서는 이 파일명을 읽어 촬영된 이미지를 불러온다. 불러온 이미지는 기존에 저장되어 있는 총 170개의 사진들과 모두 Feature Matching 과정을 거치게 된다.



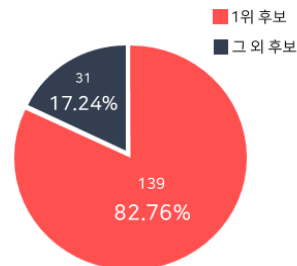
(그림 4) Feature Matching 결과 화면  
(Figure 4) Result Screen of Feature Matching

다음 (그림 4)는 Feature Matching 과정을 거친 후 나온 결과에 대한 예제 화면이다[11-12]. (그림 4)의 왼쪽은 일

치한 이미지들끼리의 비교한 결과를 나타내고, 오른쪽은 화면은 일치하지 않는 이미지들을 비교한 결과를 나타낸 모습이다. 파란색 선은 각 이미지에서 서로 일치하는 피쳐를 연결해놓은 것이다[13]. (그림 4)와 같이 일치하는 사진을 비교하면 파란색 선이 확연히 많은 것을 확인할 수 있다. 따라서 본 논문에서 제안하는 방법은 일치하는 피쳐의 개수를 계산하여 총 170개의 데이터에 일치하는 피쳐 개수가 가장 많은 순으로 정렬하고 상위 6개의 낙관 데이터만을 사용자에게 보여주도록 구현하였다. 데이터들의 파일명들을 모두 인덱스로 저장하여 상위 6개의 데이터 파일명을 데이터베이스로 전송할 수 있다.

#### 3.2 알고리즘 정확성 측정 및 원인분석

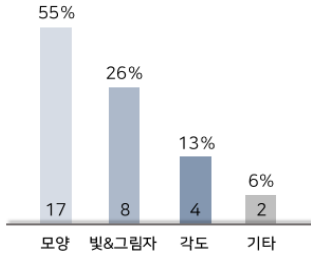
구현한 이미지 검색 알고리즘의 정확성을 측정하기 위해서 170개의 낙관 이미지 데이터를 하나씩 모두 입력으로 하여 저장되어 있는 낙관 이미지와 모두 비교하는 테스트를 하였다. 그 결과 (그림 5)와 같이 추천되는 6개의 후보 중 일치하는 낙관 이미지의 후보가 1위일 경우는 170번의 실험 중 139번인 82.76%로 나타났고, 2~6위의 후보로 추천된 경우는 31번인 17.24%로 측정되었다[14].



(그림 5) 낙관 알고리즘의 실험 결과  
(Figure 5) Experimental Result of Luo-kuan Algorithm

실험결과에 대해서 일치하는 낙관 이미지 데이터가 1위 후보로 나오지 않은 원인을 분석하기 위해 17.24%에 해당하는 31개의 데이터를 한 번씩 재시도하여 알고리즘을 수행하여 분석한 결과, 1위 후보로 선정되지 않은 오류의 원인은 다음 (그림 6)과 같았다.

낙관을 검색하기 위해 찍은 이미지에 대해 빛과 그림자에 따른 원인이 26%, 이미지의 촬영 각도에 따른 원인이 13%를 차지하였으며, 모양에 따른 원인이 55%로 가장 높은 비율을 차지했다. 낙관의 모양은 일반적으로 정사각



(그림 6) 오류 원인 분석 결과  
(Figure 6) Result Analysis for Error

형이 기본이지만, 직사각형이나 원형 등 다양한 모양의 낙관도 존재한다. 그러나 정사각형 이외의 모양의 낙관에 서는 검색의 정확성이 떨어지는 것을 확인할 수 있었다.

### 3.3 낙관 인식 알고리즘 개선

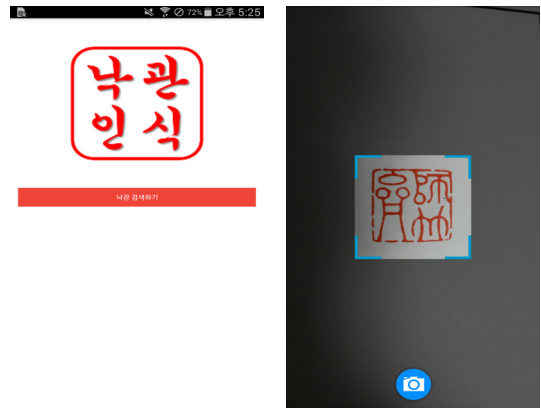


(그림 7) 서버의 데이터베이스에 저장된 사진(좌) / 입력으로 촬영된 사진(우)  
(Figure 7) Saved Photo Images in Database Server(Left) / Taken Photo Images for Input(Right)

위의 실험 결과 정사각형 모양의 낙관 이미지를 제외한 다른 이미지들에서 정확성이 떨어지는 이유는 비교하는 사진의 해상도가 다른 것이 가장 큰 원인으로 작용했다. (그림 7)과 같이 데이터베이스 서버에 저장되어 있는 이미지는 직사각형이지만, 촬영되어 비교하는 이미지는 정사각형으로 입력되기 때문에 두 이미지의 피쳐들의 위치가 다르게 측정되었다. 따라서 같은 이미지를 촬영하였지만 서로 다른 이미지라는 결과가 도출되는 것을 확인할 수 있었다. 이와 같은 원인을 제거하고 알고리즘의 성능을 개선하기 위해서 데이터베이스에 저장된 이미지들을 낙관 모양이 아니라 통일된 정사각형의 형태로 수정하여 저장하였다. 이러한 수정을 통해 오류의 55%에 해당하는 모양에 의한 오류를 전부 해결할 수 있었다.

## 4. 애플리케이션 구현 결과

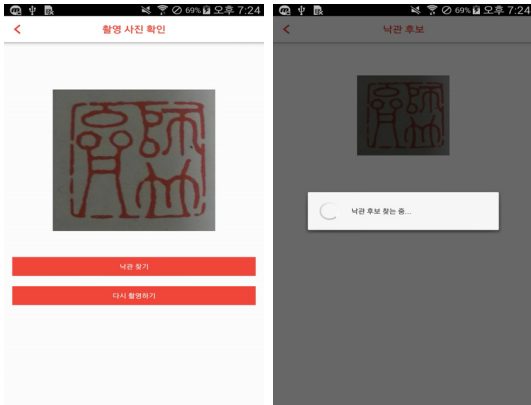
다음 (그림 8)은 애플리케이션의 결과 화면이다. 시작 화면에는 카메라 화면으로 전환할 수 있는 낙관 검색하기 버튼이 있어, 버튼을 누르면 애플리케이션 뷰가 카메라로 전환된다. 카메라는 애플리케이션 내에서 실행되는 카메라로, 애플리케이션 뷰의 중앙에 실제 이미지 검색으로 사용할 낙관의 이미지를 촬영하기 위한 영역이 존재하는데, 애플리케이션은 사용자가 이 영역에 맞춰서 낙관을 촬영하도록 유도하고 있다.



(그림 8) 시작 화면 / 카메라로 촬영된 낙관 이미지  
(Figure 8) Start Screen / Taken Input Image from Camera

(그림 9)는 촬영된 사진을 확인하는 화면과 낙관 찾기 버튼을 누른 후 나오는 로딩 화면이다. 촬영 확인 화면에서는 카메라를 이용하여 촬영한 이후의 영역에 맞춰서 수정되고 난 결과 이미지를 확인할 수 있다. 수정된 사진에 이상이 있을 경우 재촬영을 할 수 있도록 하였으며, 수정된 사진에 이상이 없어 해당 이미지를 바탕으로 유사 낙관을 검색하고 싶을 경우 낙관 찾기 버튼을 눌러서 진행하도록 하였다.

(그림 10)은 검색 알고리즘을 수행한 후 추천된 낙관들의 결과를 출력하는 화면과 6개의 낙관 검색 결과 중 사용자가 원하는 이미지에 대한 정보를 확인하는 화면이다. 이전 화면에서 낙관 찾기 버튼을 누르고 검색 알고리즘을 통해 선택된 후보들은 아래 이미지 버튼 형태로 출력된다. 또한, 사용자가 촬영한 이미지도 상단에 출력되도록 설정하여 사용자의 눈으로 직접 비교할 수 있도록 하였다. 후보 중에서 사용자가 원하는 후보의 이미지를 누



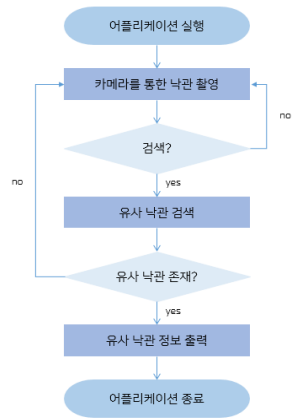
(그림 9) 촬영된 입력 이미지 확인 / 데이터베이스 검색 중인 화면  
 (Figure 9) Checking for Taken Input Image / Searching Image from the Database Server



(그림 10) 검색된 후보 이미지 / 검색된 낙관의 정보  
 (Figure 10) Searching Results from Database Server / Searched Detail Information for Input Luo-kuan image

르면 그에 대한 정보가 출력된다. 정보는 그 후보의 이미지가 상단에 출력되며, 아래에는 후보 낙관에 대한 정보(이름, 출처, 모양 등)가 출력되어 사용자가 확인할 수 있다. 상단의 버튼을 눌러 이전 화면으로 돌아가 다른 낙관에 대한 정보도 확인할 수 있다. 애플리케이션의 전체 프로세스에 대한 알고리즘의 순서도는 (그림 11)과 같다.

본 논문에서 제안하는 방법을 통한 최종 실험에서 170개의 데이터에 대한 170번의 실험 중 156번(91.76%)이 1위의 후보로 추천되었고 나머지 14번(8.24%)은 2~6위에 포함되어 추천되어 입력된 낙관 이미지에 대해 성공적으로 검색을 수행하였다.



(그림 11) 애플리케이션 실행 순서도  
 (Figure 11) Flow Chart of Application

## 5. 결 론

동양화에는 작품의 중요한 정보를 함축한 이미지인 낙관이 존재하는데, 대부분의 낙관들은 다양한 모양과 어려운 한자로 구성되어 있어 쉽게 해석하지 못하고 있는 상황이다. 따라서 본 논문에서는 낙관 이미지를 촬영하여 가장 유사한 후보들을 찾아내기 위한 검색 알고리즘과 그에 대한 정보를 출력해주는 낙관 인식 애플리케이션을 안드로이드 기반으로 구현하였다. 이전 연구에서는 **Template Matching**을 이용하여 임시로 합쳐놓은 이미지 중 유사한 이미지 3개만 검색했으며, 서버가 아닌 플랫폼 자체에서 검색하도록 하였다. 때문에 많은 양의 이미지를 검색하거나 비교할 수 없었으며, 정확한 결과가 나오지 않는다는 점을 고려해 이번 연구에서는 검색 알고리즘을 구현하여 서버에서 검색하고, 데이터를 주고받을 수 있도록 하였다. 검색 알고리즘은 **Feature Matching**을 사용하였으며, 피처를 이용해 유사 낙관을 검색하도록 하였다. 또한, 알고리즘에 대한 정확성을 분석하였고, 정확성이 낮아지는 가장 큰 원인인 낙관의 입력 모양을 해결하여 알고리즘을 개선하였다. 추후에는 해결하지 못한 빛과 촬영 각도 등의 원인에 대한 정확성을 개선하기 위하여 **OpenCV**의 다양한 작업을 적용해 최적화 알고리즘을 구현할 예정이며, 검색 알고리즘의 속도를 최적화시킬 수 있는 멀티쓰레딩 알고리즘을 구현할 계획이다. 이 기술을 바탕으로 빅 데이터 환경에서 자동으로 낙관 이미지를 분석하고 검색하는 플랫폼을 구축할 수 있을 것으로 기대한다.

## 참고문헌(Reference)

- [1] Sanya Patel, Aliya Sayyed, P.P Bastawade, "Smart Attendance Application Using Android and PHP", International Journal of Advanced Research in Computer and Communication Engineering, Vol. 6, Issue 3, 2017  
<https://doi.org/10.17148/ijarce.2017.6351>
- [2] Shoki Fukuda, Shun Kurihara, Shintaro Hamanaka, Masato Oguchi, Saneyasu Yamaguchi, "Accelerated test for applications with client application and server software", IMCOM '17 Proceedings of the 11th International Conference on Ubiquitous Information Management and Communication, No.100, 2017  
<http://dx.doi.org/10.1145/3022227.3022326>
- [3] HAN Lu, LI Zu-shu, CHEN Dong-yi, "Object detection module based on implementation of Java and OpenCV", Journal of Computer Applications, pp. 733-775, 2008  
<https://doi.org/10.3724/sp.j.1087.2008.00773>
- [4] Pavel JETENSKÝ, "Using Open Source OpenCV Library for practical courses of Computer Vision", Journal of Technology and Information Education, Vol. 5, Issue 1, 2013  
<https://doi.org/10.5507/jtie.2013.017>
- [5] R. Manoharan, S. Chandrakala, "Android OpenCV based effective driver fatigue and distraction monitoring system", Computing and Communications Technologies (ICCT), 2015 International Conference on, 2015  
<https://doi.org/10.1109/icct2.2015.7292757>
- [6] OpenCV, "Feature Matching",  
[http://docs.opencv.org/trunk/dc/dc3/tutorial\\_py\\_matcher.html](http://docs.opencv.org/trunk/dc/dc3/tutorial_py_matcher.html)
- [7] Wikipedia, "Feature detection(computer vision)",  
[https://en.wikipedia.org/wiki/Feature\\_detection\\_\(computer\\_vision\)](https://en.wikipedia.org/wiki/Feature_detection_(computer_vision))
- [8] Wikipedia, "Speeded up robust features(SURF)",  
[https://en.wikipedia.org/wiki/Speeded\\_up\\_robust\\_features](https://en.wikipedia.org/wiki/Speeded_up_robust_features)
- [9] S. Li "The Application of Face Recognition Based on OpenCV", Advanced Materials Research, Vols. 403-408, pp. 2350-2353, 2012  
<https://doi.org/10.4028/www.scientific.net/amr.403-408.2350>
- [10] Guobo Xie, Wen Lu, "Image Edge Detection Based On Opencv", International Journal of Electronics and Electrical Engineering, Vol.1, No.2, 2013  
<https://doi.org/10.12720/ijeee.1.2.104-106>
- [11] Kim, Gihong, Chong, Kyusoo, Youn, Junhee, "Automatic Recognition of Direction Information in Road Sign Image Using OpenCV", Journal of the Korean Society of Surveying, Geodesy, Photogrammetry and Cartography, Volume 31, Issue 4, pp. 293-300, 2013.  
<http://dx.doi.org/10.7848/ksgpc.2013.31.4.293>
- [12] Ye Zhao, Richang Hong, Jianguo Jiang, Jing Wen, Hengheng Zhang, "Image matching by fast random sample consensus", ICIMCS '13 Proceedings of the Fifth International Conference on Internet Multimedia Computing and Service, pp.159-162, 2013.  
<http://dx.doi.org/10.1145/2499788.2499852>
- [13] Raj Kumar Gupta, Alex Yong-Sang Chia, Deepu Rajan, Ee Sin Ng, Huang Zhiyoung, "Image colorization using similar images", MM '12 Proceedings of the 20th ACM international conference on Multimedia, pp. 369-378, 2012.  
<http://dx.doi.org/10.1145/2393347.2393402>
- [14] Yaron Lipman, Stav Yagev, Roi Poranne, David W. Jacobs, Ronen Basri, "Feature Matching with Bounded Distortion", ACM Transactions on Graphics (TOG), Volume 33, Issue 3, 2014.  
<http://dx.doi.org/10.1145/2602142>



◎ 저 자 소 개 ◎



**김 한 슬(Han-Syel Kim)**

2014년~현재 순천향대학교 컴퓨터소프트웨어공학과 재학  
관심분야 : 영상처리, 컴퓨터 그래픽스, 가상현실  
E-mail : hansyel@sch.ac.kr



**서 귀 빈(Kwi-Bin Seo)**

2017년 순천향대학교 컴퓨터소프트웨어공학과(공학사)  
2017년~현재 순천향대학교 컴퓨터학과 재학(석사과정)  
관심분야 : 영상처리, 바이오 인포매틱스, 머신 러닝, 센서 프로세싱  
E-mail : eirtjmlqls@sch.ac.kr



**강 민 구(Mingoo Kang)**

1986년 연세대학교 전자공학과(공학사)  
1989년 연세대학교 전자공학과(공학석사)  
1994년 연세대학교 전자공학과(공학박사)  
1985년~1987년 삼성전자 연구원  
2000년~현재 한신대학교 IT콘텐츠학과 교수  
관심분야 : 모바일융합시스템, 정보통신표준, 스마트 디바이스  
E-mail : kangmg@hs.ac.kr



**류 기 수(Gee Soo Ryu Kang)**

1984년 한국외국어대학 중국어과(문학사)  
1987년 홍콩대학교 중문과(철학석사)  
1992년 홍콩대학교 중문과(철학박사)  
1995년~현재 한신대학교 중국어문화학과 교수  
관심분야 : 중국문화, 동양화, 낙관  
E-mail : ryugisu@hs.ac.kr



**홍 민(Min Hong)**

1995년 순천향대학교 전산학과(공학사)  
2001년 University of Colorado at Boulder(공학석사)  
2005년 University of Colorado at Denver(이학박사)  
2006년~현재 순천향대학교 컴퓨터소프트웨어공학과 교수  
관심분야 : 컴퓨터그래픽스, 다이나믹 시뮬레이션, 바이오 인포매틱스, 영상처리  
E-mail : mhong@sch.ac.kr