

GUI에 기반한 모바일 앱 사용상태 구분

GUI-based Detection of Usage-state Changes in Mobile Apps

강 량 경*, 석 호 식*

Ryangkyung Kang*, Ho-Sik Seok*

Abstract

Under the conflicting objectives of maximum user satisfaction and fast launching, there exist great needs for automated mobile app testing. In automated app testing, detection of usage-state changes is one of the most important issues for minimizing human intervention and testing of various usage scenarios. Because conventional approaches utilizing pre-collected training examples can not handle the rapid evolution of apps, we propose a novel method detecting changes in usage-state through graph-entropy. In the proposed method, widgets in a screen shot are recognized through DNNs and 'onverted graphs. We compared the performance of the proposed method with a SIFT (Scale-Invariant Feature Transform) based method on 20 real-world apps. In most cases, our method achieved superior results, but we found some situations where further improvements are required.

요 약

모바일 앱의 신뢰성 향상과 개발 환경 변화라는 제약 조건을 모두 만족시키려면 모바일 앱의 동작을 자동으로 검증할 필요가 있다. 모바일 앱의 동작 검증 과정에서 다양한 이슈가 발생하나, 사용 상태 변화 탐지도 중요한 이슈 중 하나이다. 본 논문에서는 모바일 앱의 사용 상태 변화 탐지를 위하여 딥뉴럴넷을 이용하여 모바일 앱 GUI의 UI 위젯을 인식한 후 인식된 위젯간의 관계를 그래프로 변환하고, 변환된 그래프의 그래프 엔트로피를 계산하여 사용 상태 변화를 감지하는 방법을 제안한다. 제안 방법은 SIFT(Scale-Invariant Feature Transform)에 기반한 감지 방법과 비교되었으며 20개의 실제 모바일 앱의 동작 데이터를 통해 검증한 결과 대부분의 경우 제안 방법이 우수하나, 엔트로피 계산이 어려울 때는 제안 방법의 성능이 저하됨을 확인하였다.

Key words : Automated app testing, Deep neural nets, Graph entropy, Stream analysis, Change detection

* Dept. of Computer and Communication Engineering,
Kangwon National University

★ Corresponding author

E-mail : hsseek@kangwon.ac.kr, Tel : +82-33-250-6381

※ Acknowledgment

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (2018R1D1A1B07047156).

Manuscript received Jun. 6, 2019; revised Jun. 24, 2019; accepted Jun. 25, 2019.

This is an Open-Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. 서론

스마트폰의 보급률 및 사용 범위가 커짐에 따라 모바일 앱 시장이 급속도로 성장하고 있다. 'AppBrain'에 따르면 2019년 6월 Google Play에 등록된 앱 수가 260만 개가 넘었다[1]. 스마트폰 앱 시장이 발전하면서, 시간과 비용이 많이 드는 QA(Quality Assurance) 업무를 자동화하기 위하여 다양한 접근법이 시도되고 있다[2][3]. 다양한 동작 검증 방법 중 앱의 GUI (Graphical User Interface)를 인식한 후 가능한 기능을 실행시키고 문제 발생 여부를 관찰하는 방식이 주목받고 있는데[4], GUI에 기반을 둔 동작 검증 과정에서 중요한 기능 중 하나가 앱 사용 상

태 변화를 탐지하는 것이다. 앱의 사용 상태란 온라인 쇼핑물을 예로 들 때, 로그인, 상품열람, 후기 확인, 결제와 같은 사용자의 앱 활용 상태를 의미한다. 모바일 앱의 동작을 자동으로 검증하려면 특정 상태에서의 기능을 확인한 후, 이전 상태로 돌아간 다음 새로운 상태에서의 동작을 확인하는 시나리오가 필수적이기 때문에 사용 상태 변화 감지가 매우 중요하다.

사용 상태 변화 감지를 스트림 데이터에서의 변화 감지로 해석할 경우 [5]에서 소개한 바와 같이 그래프 변화 프로세스를 활용하는 방안도 가능하지만, 모바일 앱 데이터의 경우 훈련 데이터의 확보가 쉽지 않고 기능 및 디자인의 변화가 크기 때문에 훈련 데이터의 유효성이 오래 지속되지 못한다는 문제가 있다.

본 논문에서는 훈련 데이터를 활용하지 않는 방법의 하나로 그래프 엔트로피(Graph entropy)를 이용하여 사용 상태의 변화를 감지하는 방법을 제안한다. 제안 방법은 (1) 모바일 앱의 변화 속도, 새로운 기능 소개 속도, 사용자 사용 상태의 다양함을 고려하여, 훈련 데이터를 요구하지 않는 무감독학습 방식으로 사용 상태 변화를 탐지한다. 또한, (2) DNNs (Deep Neural Networks)을 이용하여 GUI를 구성하는 UI 위젯 (User Interface-widget)을 인식한 후 인식된 UI 위젯 간의 관계를 활용하는 방식을 취하여 GUI에 대한 의미 정보를 최대한 반영하였다는 특징이 있다.

모바일 앱 동작 데이터는 일반적인 이미지 데이터와 비교했을 때 잡음의 영향으로 인한 이미지의 품질 저하가 적다는 장점이 있다. 그러나 메뉴 확장이나 화면 스크롤, 중첩 메뉴 등의 현상으로 인하여 SIFT(Scale-Invariant Feature Transform)와 같은 분석 방법을 사용할 경우, 동일 사용 상태를 서로 다른 상태로 판단할 가능성이 크다는 어려움이 있다. 모바일 앱 데이터의 특성을 반영하면서 사용 상태 변화를 탐지할 수 있도록 제안 방법에서는 GUI를 구성하는 UI 위젯을 노드(Node)로 하고, 노드와 노드 사이의 거리 정보를 에지(Edge)로 하는 그래프로 GUI 정보를 표현하였다. 마지막으로 연속된 그래프 사이의 엔트로피 변화를 이용하여 사용 상태의 변화를 결정하였다.

본 논문에서는 널리 사용되는 안드로이드 앱(Android app) 20개를 대상으로 제안 방법의 성능

을 검증하였다. 특히 SIFT를 이용한 변화 감지법에 대하여 제안 방법의 성능을 비교하였으며 감지 결과 중 주목할만한 현상에 대한 분석 결과를 보고한다.

II. 관련 연구

그래프에서의 엔트로피는 그래프를 구성하는 노드 집합의 확률 분포를 이용하여 정의된 값으로[6], 자연언어의 이해[7], 화학구조 분석[8] 등 다양한 분야에서 활용되고 있다. 본 논문에서는 특히 [9]에서 정의한 그래프 엔트로피를 활용하였으며 구체적인 정의는 다음과 같다.

확률 변수 X 와 X 를 활용하여 정의된 그래프 G 가 주어졌을 때, 그래프 G 를 구성하는 노드 집합의 원소가 서로 인접하지 않은 경우 해당 집합은 독립(Independent)라고 하고, 그래프 G 의 독립 집합(Independent set)의 모음을 $\Gamma(G)$ 라고 표시한다. 이와 같은 정보가 주어졌을 때 [6]에서는 X 의 그래프 엔트로피를

$$H_G(X) = \min_{X \in W \in \Gamma(G)} \{I(W; X)\}$$

와 같이 정의했다. 목표 도메인인 모바일 앱 동작 검증 문제에서, GUI 데이터로부터 추출한 그래프의 독립 집합들은 서로 다른 확률로 생성되지는 않으므로 [9]에서 소개한 표현식을 활용하여

$$H_G(X) = H(W) - H(WX) \quad (1)$$

로 설정하였다. 여기서 W 는 그래프 G 의 극대 독립 집합(Maximal Independent Set)이다.

일련의 이미지 데이터가 연속적으로 입력되는 스트림 데이터의 분석은 일반적으로 쉽지 않은 작업이나 이런 상황에서 SIFT를 적용하여 문제를 해결하려는 연구가 존재한다. [10]에서는 TISLF(Target Image Search based on Local Features)라는 방법을 소개하였는데, TISLF에서는 SIFT를 활용하여 씬(Scene)의 변화 가능성을 확률로 표현하였다. [11]은 SAR(Synthetic Aperture Radar) 데이터에서의 변화를 무감독학습 방식으로 감지하기 위하여 SIFT를 사용하였다.

III. GUI정보에 기반한 사용상태 변화탐지

1. 사용 상태 변화와 사용 데이터

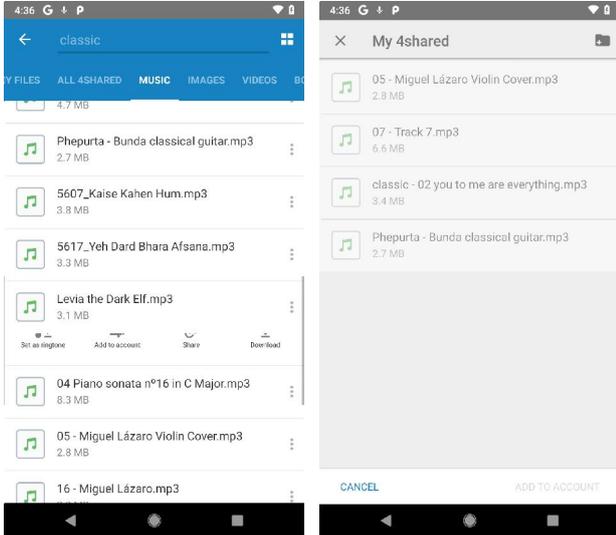


Fig. 1. An example of two images in the different states.

그림 1. 다른 상태로 정의되는 두 이미지 예시

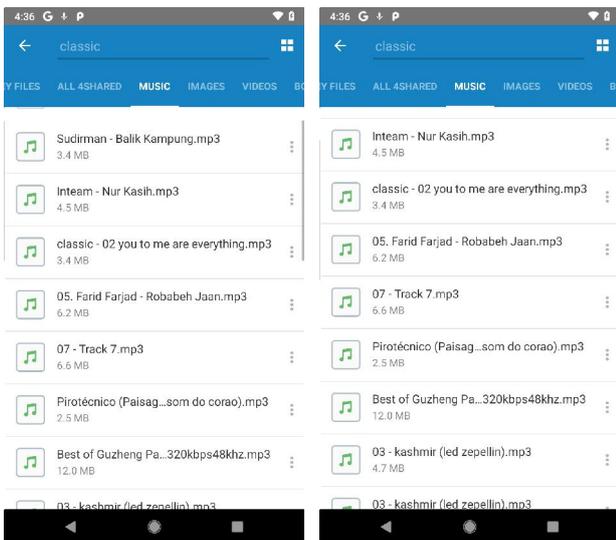


Fig. 2. An example of two images in the same state.

그림 2. 같은 상태로 정의되는 두 이미지 예시

모바일 앱에서의 사용 상태 변화는 정의가 어렵다. 예를 들어 그림 1의 경우 음악 파일의 집합이라는 점에서 동일한 정보를 제공한다고 볼 수도 있지만, 왼쪽 그림에서 오른쪽 그림으로 변환하는 과정에서 UI 위젯의 실행이라는 동작이 요구되고, 해당 위젯의 실행을 통해 프로그램 실행 측면에서 새로운 메소드의 호출 혹은 자료구조 생성이 요구되므로 서로 다른 상태라고 판단할 수 있다. 하지만 그림 2의 경우 화면에 표시되는 음악 파일의 정보는

변하였지만 현 상태로 진입할 때 주어진 목록은 같고 화면의 표시 정보 제한 때문에 서로 다른 정보가 표현되는 것이므로 이때는 동일 상태라고 판단하는 것이 합당하다.

상태를 엄밀하게 정의하는 것이 쉽지 않기 때문에 본 논문에서는 먼저 Google Play store의 인기 앱 목록[12]을 참조하여 선정된 20개의 앱을 다운받아 설치한 후 인간 사용자가 직접 앱을 실행하면서 데이터를 수집하고, 수집된 동작 데이터에 대하여 인간 사용자가 상태 변화 지점을 정의한 후 해당 지점의 감지 여부를 측정하는 방식으로 실험을 진행하였다. 실험에 사용한 데이터의 특성은 표1과 같다.

Table 1. Number of images and transition points.

표 1. 앱별 이미지 수 및 변환점 수

App name	images	transition points
4shared	232	20
Airbnb	223	17
Alba	93	8
Albamon	177	10
Amazon Shopping	202	6
Baemin	136	6
Carrot Market	114	4
Cbs Sports App	253	11
Cgv	155	16
Coupang	134	9
Ebay	150	9
Emart Mall	159	8
ESPN	98	7
Facebook	226	18
FLO	177	13
Google Translate	190	7
Hwahae	197	15
Home and Shopping	178	19
Kakao Bus	280	29
Korail	213	23

2. 엔트로피 기반 사용 상태 변화 감지 방법

특정 앱이 주어졌을 때 사용상태 감지를 위해 본 논문에서 제안한 방법은 기본적으로 (1) Faster R-CNN[13]을 활용한 UI 위젯 인식, (2) 인식된 UI 위젯 사이의 관계를 활용하여 앱 동작 화면을 그래프로 변환 (3) 그래프의 엔트로피 계산 (4) 엔트로피 차이를 활용한 사용 상태 변화 결정의 과정으로 구성된다(알고리즘 1).

Algorithm 1. Change Detection

Input: GUI images of a mobile app

1: UI widget detection using Faster R-CNN.
 For a GUI image X_{i+1}

2: Transform X_{i+1} into graph G_{i+1}

3: Compute dissimilarity d_i using graph entropy

$$d_i = abs(H_G(X_{i+1}) - H_G(X_i))$$

if

4: $d_i > \frac{\min(d_{1:i}) + \max(d_{1:i})}{2}$
 set (G_i, G_{i+1}) as a transition point

스텝 1에서는 UI 위젯 인식을 위해, 레이블이 부착된 훈련 데이터를 이용하여 그림 3의 총 13개 카테고리의 UI 위젯을 인식한다. 스텝 2에서는 이미지 X_{i+1} 를 그래프 G_{i+1} 로 변환한다. 그래프 변환을 위하여 UI 위젯을 노드로 설정하고, 노드 사이의 거리를 가중치로 갖는 완전 그래프(Complete graph)를 만든 후, Prim의 알고리즘을 이용하여 그래프를 최소신장트리(Minimum spanning tree)로 변환한다. 이때, 거리는 위젯 사이의 최소 거리를 사용하였고, 두 위젯이 붙어 있거나 겹쳐 있으면, 겹쳐 있는 면적을 역수로 하여 거리를 계산한다. 스텝 3에서는 이전 이미지에서 확보한 그래프 G_i 와 현재 이미지의 그래프 G_{i+1} 을 비교하는데 이 과정에서 수식 (1)를 계산한다. 수식 (1) 계산 과정에서는 NetworkX[14]를 활용하여 극대 독립 집합을 확보하였다. 스텝 4에서는 상태 변화 여부를 판단한다. 상태 변화 여부 판단을 위해 스텝 3에서 확보한 그래프 엔트로피의 변화량의 절댓값을 현재까지 관찰한 그래프 엔트로피의 최솟값과 최댓값의 평균과 비교한 후 평균보다 큰 경우 해당 지점에서 변화가 발생하였다고 판정하였다.

SIFT에 기반한 사용상태 변화 판정은 [10]에서 제안한 방법을 활용하여 SIFT 정보 비율을 계산한 후, 마찬가지로 최솟값과 최댓값의 평균과 비교하여 평균보다 큰 경우 해당 지점에서 변화가 발생하였다고 판정하였다. 두 방법 모두, 실제 변화 지점 바로 앞, 뒤 이미지에서 변화가 일어났다고 판정한 경우는 옳게 판정한 것으로 판단하였다.

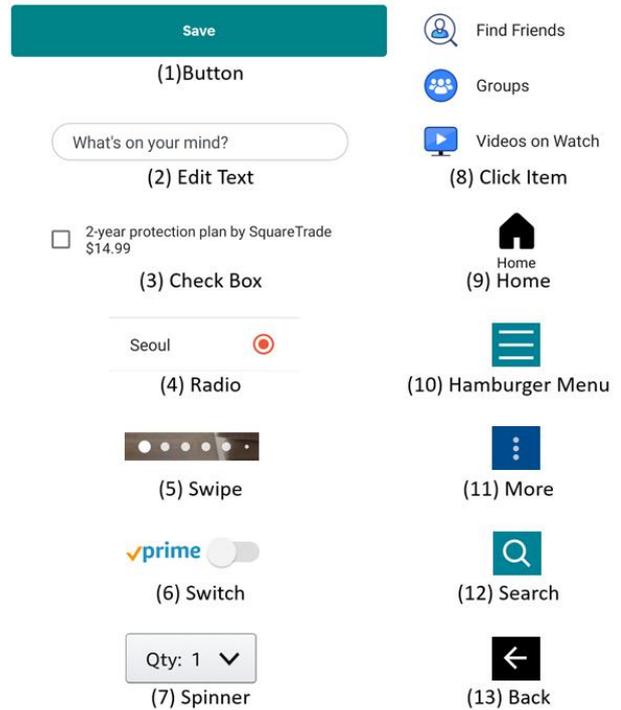


Fig. 3. Categories of UI widgets.

그림 3. UI 위젯 카테고리

VI. 실험결과 및 분석

Table 2. Detection performance.

표 2. 사용 상태 감지 성능

App name	Precision(Graph)	Precision(SIFT)
4shared	0.894736842	0.41025641
Airbnb	0.272727273	0.75
Alba	0.5	0.1875
Albamon	0.315789474	0.333333333
Amazon Shopping	0.714285714	0.5
Baemin	0.157894737	0.071428571
Carrot market	0.090909091	0.444444444
Cbs Sports App	0.172413793	0.06779661
Cgv	0.647058824	0.611111111
Coupang	0.5	0.222222222
Ebay	0.333333333	0.3
Emart mall	0.285714286	0.7
ESPN	0.7	0.217391304
Facebook	0.777777778	0.5
FLO	0.666666667	0.571428571
Google Translate	0.217391304	0.128205128
Hawhae	0.28	0.229166667
Home and Shopping	0.55	0.533333333
Kakao bus	0.5625	0.285714286
Korail	0.571428571	0.288888889

20개의 앱을 대상으로 사용 상태 감지 결과의 정밀도(precision)를 계산해 봤을 때(표 2), 16개 앱에서는 그래프 엔트로피 기반 방법의 변화 감지 성능이 우수하며 4개 앱은 SIFT 기반 방법의 성능이 우수한 것으로 관찰되었다. 각 앱의 GUI 데이터를 고려하여 분석한 결과 다음과 같은 원인이 사용 상태 변화 감지에 영향을 미침을 확인하였다.

1. 그래프 엔트로피 기반 방법이 우수한 경우

가. UI 위젯의 위치 변화

스크롤 업/다운 등의 이유로 화면을 구성하는 UI 위젯은 동일하지만 위치가 변하는 경우가 많은데, SIFT 기반 방법에서는 이런 상황을 사용 상태 변화가 발생하였다고 판단하기 쉬운 반면, 그래프 엔트로피 기반 방법은 동일한 사용 상태로 판단하는 경우가 많았다. 그래프 엔트로피 기반 방법과 SIFT 기반 방법의 성능 차이가 앱마다 크데, 그래프 엔트로피 기반 방법과 SIFT 기반 방법의 정밀도 차이가 크게 나는 앱은 앱의 구성 특성 상 스크롤링 등의 동작이 많아 데이터에서 UI 위젯 위치 변화가 많이 나타나는 앱이었다.

2. SIFT 기반 방법이 우수한 경우

가. 인식된 UI 위젯이 없는 경우

딥뉴럴넷의 객체 인식으로 GUI 화면에서 UI 위젯을 찾을 때, 화면에서 UI 위젯을 전혀 인식하지 못하는 경우가 발생한다. 이 경우 그래프가 구성되지 못하기 때문에 그래프 엔트로피 기반 방법을 적용할 수 없었다.

나. UI 위젯 인식에 오류가 발생한 경우

연속된 이미지에서 같은 UI 위젯이 위치만 달라졌을 때, 어떤 이미지에서는 딥뉴럴넷에서 UI 위젯을 찾지만, 다른 이미지에서는 위젯을 찾지 못하는 경우가 있었다. 이 때, 최소신장트리의 모양이 완전히 바뀌면 그래프 엔트로피 값이 크게 변화하여 실제 화면변환이 일어나지 않았지만 화면 변환점으로 찾는 경우가 있었다.

또한, 딥뉴럴넷에서 UI 위젯을 인식은 하였으나 잘못된 카테고리의 UI 위젯으로 인식하는 경우도 있었다. 같은 위젯을 다른 위젯으로 계속 잘못 인식하는 경우 그래프 엔트로피 값이 크게 달라지지 않지만, 연속된 이미지에서 한 위젯을 잘 못 인식

하기도 하고, 제대로 인식하기도 하는 경우, 그래프 엔트로피 값이 계속 변화하기 때문에 실제 화면변환이 일어나지 않았지만 화면 변환점으로 찾는 경우가 있었다.

다. 단일 UI 위젯으로 GUI가 구성되는 경우

한 종류의 UI 위젯이 GUI 화면을 구성하는 상황에서 상태 변화가 발생하면 엔트로피 변화가 없기 때문에 사용 상태 변화 판정이 불가능하다.

V. 결론

모바일 앱의 자동 동작 검증은 사용자 만족도 유지 측면에서 매우 중요한 기능이나 현재 만족스럽게 적용할 수 있는 기술이 확립되지 못한 상태이다. 본 논문에서는 모바일 앱의 자동 검증 과정에서 중요한 기능 중 하나인 사용 상태 변화 감지를 위하여 그래프 엔트로피를 활용하는 방법을 제안하였다. 제안 방법은 딥뉴럴넷의 객체 인식 기능을 활용하여 인식된 UI 위젯 관계를 그래프로 변환한 후 그래프 엔트로피 변화를 활용하는 방법으로, 기본적인 이미지 분석 방법인 SIFT를 활용한 방법과 비교했을 때 대부분의 상황에서 우수한 탐지 기능을 보고하였다. 그러나 그래프 엔트로피 기반 방법을 적용하기 어려운 상황이 존재함을 확인하였는데, 딥뉴럴넷의 객체 인식 기능 부족으로 인한 인식 오류 및 극단적인 위젯 구성의 상황에서는 그래프 엔트로피 계산이 어렵기 때문에 사용 상태 변화를 감지할 수 없다는 것을 확인하였다. 본 연구결과를 바탕으로 추후 연구에서는 (1) UI 위젯 인식 결과의 불확실성을 반영할 수 있도록 그래프 생성 프로세스를 추정 후 상태 변화를 결정하는 방법을 개발할 것이며 (2) 엔트로피 계산이 어려운 상황에서 상태 변화를 판정할 수 있는 규칙을 생성하고자 한다.

References

- [1] AppBrain, "Number of Android apps on Google Play," <https://www.appbrain.com/stats/number-of-android-apps/>
- [2] M. Linares-Vásquez, K. Moran, and D.

Poshyvanyk, "Continuous, Evolutionary and Large-Scale: A New Perspective for Automated Mobile App Testing," in Proc. of the *IEEE International Conference on Software Maintenance and Evolution(ICSME)*, 2017.

DOI: 10.1109/ICSME.2017.27

[3] K. Mao, M. Harman, and Y. Jia, "Sapienz: multi-objective automated testing for Android applications," in Proc. of the *25th International Symposium on Software Testing and Analysis*, pp.94-105, 2016. DOI: 10.1145/2931037.2931054

[4] Y. Li, Z. Yang, Y. Guo, and X. Chen, "A Deep Learning based Approach to Automated Android App Testing," *arXiv:1901.02633*, 2019.

[5] D. Grattarola, D. Zambon, C. Alippi, and L. Livi, "Change Detection in Graph Streams by Learning Graph Embeddings on Constant-Curvature Manifolds," *arXiv:1805.06299v3*, 2019.

[6] K. János, "Coding of an information source having ambiguous alphabet and the entropy of graphs," in Proc. of the *6th Prague conference on information theory*, 1973.

[7] A. Mehler, A. Lücking, and P. Weiß, "A network model of interpersonal alignment in dialog," *Entropy*, vol.12, pp.1440-1483, 2010.

DOI: 10.3390/e12061440

[8] M. Dehmer, N. Barbarini, K. Varmuza, and A. Graber, "A large scale analysis of information-theoretic network complexity measures using chemical structures," *PLoS ONE*, vol.4, no.12, e8057, 2009. DOI: 10.1371/journal.pone.0008057

[9] A. Orlitsky and J. R. Roche, "Coding for computing," *IEEE Trans. Info. Theory*, vol.47, no.3, pp.903-917, 2001. DOI: 10.1109/SFCS.1995.492580

[10] B. Guan, H. Ye, H. Liu, and W. Sethares, "Target image video search based on local features," *arXiv:1808.03735v2*, 2019.

[11] Y. Wang, L. Du, and H. Dai, "Unsupervised SAR image change detection based on SIFT keypoints and region information," *IEEE Trans. Geosci. Remote Sens.*, vol.13, no.7, pp.931-935, 2016. DOI: 10.1109/LGRS.2016.2554606

[12] <https://play.google.com/store/apps/top/>

[13] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.39, no.6, pp.1137-1149, 2016. DOI: 10.1109/TPAMI.2016.2577031

[14] <https://networkx.github.io/>

BIOGRAPHY

Ryangkyung Kang (Member)



2015 : BS degree in English Education,
Kangwon National University.

Ho-Sik Seok (Member)



1999 : BS degree in Computer Engineering, Seoul National University.
2001 : MS degree in Electrical Engineering and Computer Science, Seoul National University.

2012 : PhD degree in Electrical Engineering and Computer Science, Seoul National University.

2001~2004 : Instructor, Korea Military Academy.

2016~present : Assistant professor, Dept. of Computer and Communications Engineering, Kangwon National University.