

# Camera Model Identification Based on Deep Learning

Soo Hyeon Lee<sup>†</sup> · Dong Hyun Kim<sup>††</sup> · Hae-Yeoun Lee<sup>†††</sup>

## ABSTRACT

Camera model identification has been a subject of steady study in the field of digital forensics. Among the increasingly sophisticated crimes, crimes such as illegal filming are taking up a high number of crimes because they are hard to detect as cameras become smaller. Therefore, technology that can specify which camera a particular image was taken on could be used as evidence to prove a criminal's suspicion when a criminal denies his or her criminal behavior. This paper proposes a deep learning model to identify the camera model used to acquire the image. The proposed model consists of four convolution layers and two fully connection layers, and a high pass filter is used as a filter for data pre-processing. To verify the performance of the proposed model, Dresden Image Database was used and the dataset was generated by applying the sequential partition method. To show the performance of the proposed model, it is compared with existing studies using 3 layers model or model with GLCM. The proposed model achieves 98% accuracy which is similar to that of the latest technology.

**Keywords :** Deep Learning, Camera Model Identification, Convolutional Neural Network, High Pass Filter, Gray Level Co-Occurrence Matrix

## 딥러닝 기반 카메라 모델 판별

이 수 현<sup>†</sup> · 김 동 현<sup>††</sup> · 이 해 연<sup>†††</sup>

## 요 약

멀티미디어 포렌식 분야에서 영상을 촬영한 카메라 모델 판별을 위한 연구가 지속되어 왔다. 점점 고도화되는 범죄 중에 불법 촬영 등의 범죄는 카메라가 소형화됨에 따라 피해자가 알아차리기 어렵기 때문에 높은 범죄 발생 건수를 차지하고 있다. 따라서 특정 영상이 어느 카메라로 촬영되었는지를 특정할 수 있는 기술이 사용된다면 범죄자가 자신의 범죄 행위를 부정할 때, 범죄 혐의를 입증할 증거로 사용될 수 있을 것이다. 본 논문에서는 영상을 촬영한 카메라 모델 판별을 위한 딥러닝 모델을 제안한다. 제안하는 모델은 4개의 컨볼루션 계층과 2개의 전연결 계층으로 구성되었으며, 데이터 전처리를 위한 필터로 High Pass Filter를 사용하였다. 제안한 모델의 성능 검증을 위하여 Dresden Image Database를 활용하였고, 데이터셋은 순차분할 방식을 적용하여 생성하였다. 제안하는 모델을 3 계층 모델과 GLCM 적용 모델 등 기존 연구들과 비교 분석을 수행하여 우수성을 보였고, 최신 연구 결과에서 제시하는 수준의 98% 정확도를 달성하였다.

**키워드 :** 딥러닝, 카메라 모델 판별, 컨볼루션 뉴럴 네트워크, 고주파 통과 필터, 명암도 동시발생 행렬

## 1. 서 론

스마트폰의 보급과 모바일 네트워크 기술의 발달로 인해 SNS는 폭발적인 인기를 끌며 확산되었다. 대부분의 스마트폰 사용자들은 SNS 애플리케이션을 활용하고 있으며 이를

통해 하루 수십억 건의 게시글이 업로드되고 있으며 그 내용은 단순한 텍스트뿐만 아니라 영상, 동영상 등 다양한 멀티미디어 데이터를 포함한다. SNS를 제외하더라도 카페, 블로그, 커뮤니티 등 웹을 포함해서 네트워크를 통해 떠도는 데이터는 날이 폭증하고 있다. 그중 시각 데이터의 비율은 점점 증가하고 있으며 특히 영상 데이터는 생성이 간편하고 접근성이 뛰어난 점에 힘입어 더욱더 높은 비율을 차지하고 있다.

영상 데이터는 불과 10년 전만 해도 준 전문가의 영역이었다. DSLR, Mirrorless 등의 카메라는 높은 가격대를 유지했으며 SNS를 통한 공유보다는 소규모의 커뮤니티를 통한 공유가 대부분이었다. 하지만 기술의 발전으로 카메라의 가격이 하향 평준화되었고 스마트폰에 고화질, 고성능의 카메라

\* This work was supported by the Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(2017R1D1A1B03030432).

† 비 회 원 : 금오공과대학교 소프트웨어공학과 석사과정

†† 준 회 원 : 금오공과대학교 소프트웨어공학과 석사과정

††† 정 회 원 : 금오공과대학교 컴퓨터소프트웨어공학과 교수

Manuscript Received : March 5, 2019

First Revision : June 4, 2019

Accepted : August 10, 2019

\* Corresponding Author : Hae-Yeoun Lee(haeyeoun.lee@kumoh.ac.kr)

가 기본적으로 탑재되면서 누구나 언제 어디서든 고품질의 영상을 생산할 수 있게 되었다. 이러한 영상을 웹을 통해 게시하기보다는 모바일 디바이스에 편리한 SNS 애플리케이션을 통해 다양한 사람들과 일상을 공유하게 되었다.

하지만 사진 촬영이 쉬워졌다는 점을 악용하는 사례가 다양하게 등장하여 일반인들에게 여러 방면에서 피해를 끼치고 있다. 상대방의 동의 없이 사생활을 촬영하고 유포한다는 협박을 한다거나 웹 상에 게시하여 피해를 주는 등의 범죄가 대표적이다. 이를 막기 위해 처벌의 수위를 높이거나 영상 촬영 시 주변 사람이 인지할 수 있는 정도의 촬영음을 강제하고 있지만 이러한 노력에도 불구하고 몰래 카메라, 도둑 촬영 등의 범죄는 여전히 사회적 문제로 남아있다. 또한 촬영 후 유포 혹은 소지를 하다 걸렸을 때, 촬영자로 처벌받는 것을 회피하기 위해 촬영 사실을 부정하는 경우도 존재하기 때문에 어느 카메라로 촬영된 영상인지를 판별하여 범행 도구와 연관된 범죄자를 특정하기 위한 포렌식 기법으로 영상 획득 장치(카메라) 판별 기술이 사용된다.

본 논문에서는 특정 영상에 대한 카메라 모델 판별을 위한 딥러닝 모델을 제안한다. 기존의 카메라 모델 판별 알고리즘에 사용되는 센서 패턴 노이즈 추출 기술과 최근 다양한 분야에 접목되고 있는 딥러닝 기술 중 영상 데이터 처리에 높은 성과를 보이는 컨볼루션 뉴럴 네트워크를 이용하였다. 실험을 위해 오픈 영상 데이터베이스에서 추출한 12개의 영상 획득 장치의 데이터를 활용하였고 결과적으로 4개의 Convolution Layer를 사용한 모델을 통하여 최신 연구의 성능과 유사한 98%의 성능을 달성하였다.

본 논문의 구성은 다음과 같다. 2장에서는 관련된 연구들을 제시하고 3장에서는 딥러닝을 이용한 영상 획득 장치 판별 모델을 제시한다. 4장에서는 제안한 모델의 성능을 분석하고 마지막으로 5장에서는 결론을 맺는다.

## 2. 관련 연구 및 기반 기술

영상을 촬영한 장치를 판별하기 위한 연구는 국내외에서 꾸준히 진행되고 있다. 전통적인 방법으로는 카메라의 물리적 구조에서 비롯되는 특징들을 지정하여 이를 통해 카메라를 분류하는 방식이 연구되고 있다[1-6]. 하지만 최근에 딥러닝 분야에 대한 활용이 활발해지며 디지털 포렌식 분야에도 딥러닝을 이용한 연구들이 다양하게 진행되고 있다[7-14].

### 2.1 특징 분석 기반 연구

모든 디지털 카메라의 핵심은 렌즈와 이미지 센서이다. 디지털 카메라의 처리방식은 Fig. 1(a)와 같은 방식으로 진행된다. 이 과정에서 각 카메라 모델의 고유한 특징들이 삽입된다. 별도로 카메라 센서의 도메인 처리 방식이 RGB일 경우 Fig. 1(b)의 CFA가, YMCA일 경우 Fig. 1(c)의 CFA가 사용된다. 이러한 디지털 카메라의 구조 때문에 연구들은 주로 카

메라의 렌즈나 센서의 특징을 뽑아내어 이를 통해 분류하는 방식이 활발하였고 많은 방법이 제시되었다.

Kharrazi et al.[1]은 카메라마다 특유의 색상 필터 배열(Color Filter Array, CFA), Demosaicking 알고리즘(CFA 보간), 색상 처리/변환 등을 수행한다는 점에 착안하여 연구를 진행하였다. 각 카메라로 촬영한 영상에서 RGB 채널 평균값, RGB 짝 상관분석 등 34개의 통계적 특성을 추출하고 Support Vector Machine(SVM)을 통한 카메라 판별을 시도하였다. 카메라 2대의 촬영 영상에 대한 판별은 평균 98.73%의 정확도를 보였고, 촬영 영상을 Quality Factor 75로 재압축한 영상을 분류기에 학습시킨 결과로는 평균 93.42%의 정확도를 보였다. 그리고 카메라 5대의 촬영 영상에 대한 판별은 평균 88.02%의 정확도를 보였다.

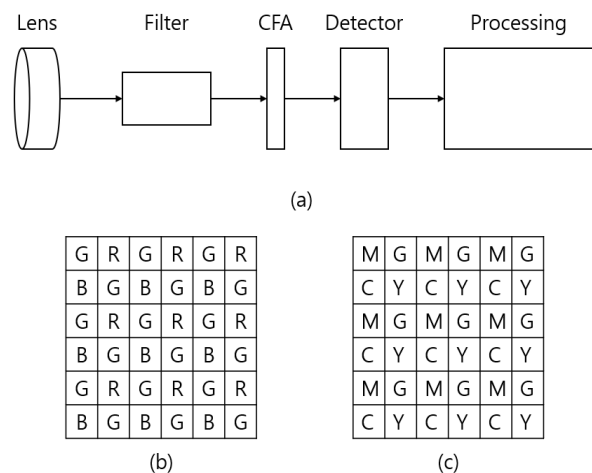


Fig. 1. (a) Major Camera Processing Stages, (b) CFA Pattern Using RGB Domain, (c) CFA Pattern Using YMCA Domain

Bayram et al.[2]은 Kharrazi et al.[1]의 특징 추출 방식 중 CFA에 관한 부분을 확장하는 방향으로 연구를 진행하였다. 카메라는 색상을 나타내기 위해 CFA를 사용하지만, CFA 사이의 보간 방법은 카메라마다 차이가 존재한다. 보간 방법의 특징 식별을 위해 영상에서 값의 변화가 적은 부분과 값의 변화가 심한 부분을 나누어 분석하였다. 특히, 특징을 추출하기 위해 Popescu et al.[3]이 업샘플링 흔적을 찾기 위해 사용한 Expectation/Maximization(EM) 알고리즘을 사용하였다. EM 알고리즘을 통해 보간 커널의 크기와 커널의 가중치 계수를 추정하고 이 특징을 카메라를 구분할 수 있는 특징으로 SVM에 학습시켜 카메라 분류를 시도하였다. Nikon사의 카메라와 Sony사의 카메라에 대한 판별을 시도하였으며 보간 커널 크기에 따라 성능이 차이가 존재했다. 3x3 커널에 대한 정확도는 89.3%, 4x4 커널에 대한 정확도는 92.9%, 5x5 커널에 대한 정확도는 95.7%를 달성했다. 위에서 언급한 2개의 카메라에 Canon사의 카메라를 포함한 5x5 커널은 83.3%의 정확도를 달성했다.

Choi et al.[4]은 렌즈의 생산과정에서 생기는 방사형 왜곡

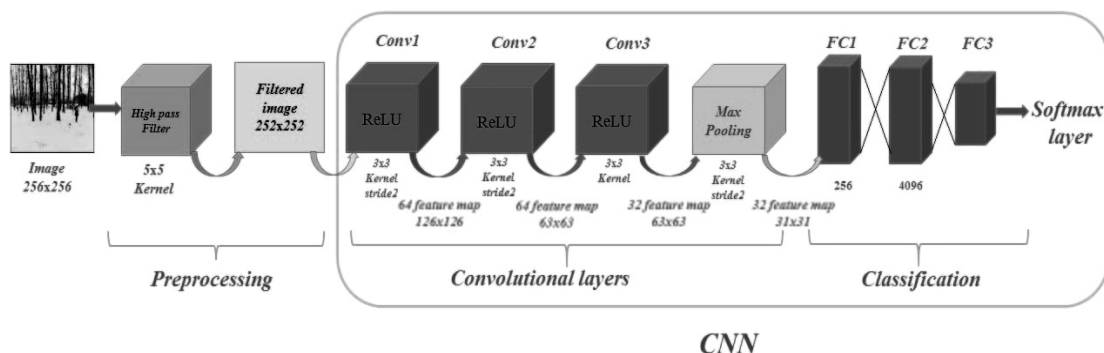


Fig. 2. The Layout of Tuama et al.[7]'s CNN for Camera Model Identification

을 이용하는 원본 카메라 판별 기술을 연구하였다. 3개의 카메라를 대상으로 했으며 각각 100장의 영상을 직접 촬영하여 수집하였고 SVM을 이용하여 학습을 진행하였다. Kharrazi et al.[1]의 특징 추출을 이용하여 실험한 결과는 평균 87.38%로 나왔지만, 렌즈 왜곡 기술을 병행할 경우 4%가 향상된 평균 91.39%의 성능을 보임으로써 성능의 개선을 입증했다. 렌즈 왜곡 기술만을 이용하여 특징을 추출한 방법의 분류 정확도는 91.53%를 달성하였다.

Lukas et al.[5]은 디지털 카메라 식별을 위해 센서 패턴 노이즈(Sensor Pattern Noise, SPN)를 활용하는 방향의 연구를 진행하였다. SPN은 De-Noise 필터를 통과한 영상과 원본 영상의 차영상을 통해 얻은 노이즈를 여러 영상에 대해 평균화하여 생성한다. 이렇게 생성된 SPN이 카메라 식별에 유의미한 결과를 보이는지 분석하기 위해 연구팀은 상관관계를 계산하였다. 그 결과 실험에 사용된 영상들의 SPN은 촬영에 사용한 각 카메라의 패턴과 높은 상관관계를 보임으로써 카메라 식별에 활용 가능한 특징임을 입증하였다. 또한 SPN이 카메라 촬영 과정에서 수행되는 Gamma 보정과 JPEG 압축에 대해서 충분히 강인함을 가지고 있다는 사실을 입증했다.

Bolouri et al.[6]은 IoT 기술에 사용되는 디지털 카메라 식별을 위해서 PRNU(Photo Response Non-Uniformity) 기반의 특징을 사용하였다. PRNU를 이용한 기존 연구들과 비교하여 특징 추출에 있어서 연산량과 실행 시간을 최적화하였다. 영상 데이터의 채널 수를 줄이기 위해 먼저 Color Channel을 고려하여 추출한 PRNU에 대해 Wavelet Transform 특징 중 'HH' 값만을 사용하였다. Color Channel을 고려한 PRNU의 성능은 97.07%를 나타내었고, 'HH' 값을 사용한 실험 결과는 90% 이상의 성능을 유지하면서 실행 속도를 상당히 개선하였다.

위의 연구들은 기계학습에 기반하여 분류기(일반적으로 SVM)에 얼마나 데이터의 특징을 잘 표현한 특징을 학습시킬 수 있는가를 초점으로 연구가 진행되었다. 분류기의 성능 향상보다는 추출한 특징과 카메라 간의 상관관계에 집중하였다.

## 2.2 딤러닝 기반 연구

딤러닝의 경우, 복잡한 데이터의 특징을 자동으로 추출하

는 장점이 있지만, 높은 컴퓨팅 성능이 요구된다는 단점에 의해 널리 사용되지 못하였다. 하지만 이후 하드웨어 성능의 발전에 힘입어 카메라 판별 연구 또한 딤러닝을 접목하는 방향으로 연구가 진행되었다.

Tuama et al.[7]은 딤러닝 기법 중 CNN을 이용한 카메라 판별 방법을 제안했다. 직접 특징을 뽑아 비교 후 학습시켰던 이전 방법들과 달리 판별 모델이 입력데이터에 대해 스스로 가중치를 조절하며 분류하는 딤러닝 기법을 적용했다. High Pass Filter(HPF)와 Wavelet Filter를 적용하는 방법으로 나누어 입력 영상에 대해 사전처리를 진행하였다. HPF는 고역 통과 필터라는 뜻으로 다음 수식과 같은 마스크 형태를 가지고 있다.

$$F^{HPF} = \frac{1}{12} \begin{pmatrix} -1 & 2 & -2 & 2 & -1 \\ 2 & -6 & 8 & -6 & 2 \\ -2 & 8 & -12 & 8 & -2 \\ 2 & -6 & 8 & -6 & 2 \\ -1 & 2 & -2 & 2 & -1 \end{pmatrix} \quad (1)$$

영상에서 대부분의 픽셀들은 선형적으로 완만한 변화값을 보이지만 노이즈가 있는 부분은 상대적으로 비선형적이거나 급격한 값의 변화를 보인다. 고역 주파수 부분만을 남겨 완만한 변화값을 보이는 부분을 제거하여 노이즈 부분을 추출한다. 이를 통해 기존의 연구들에서 시행된 특징 추출 부분을 생략한다.

Dresden Image Database를 사용하였으며 33개의 카메라를 대상으로 실험을 진행하였으며 딤러닝 모델 중 널리 알려진 AlexNet[8]과 GoogleNet[9]에 대한 실험 결과와 비교를 하였다. 기존에 깊은 계층으로 이루어진 AlexNet과 GoogleNet도 좋은 성능을 보여주었지만 Fig. 2와 같은 3계층의 모델만으로도 충분히 좋은 성능을 보여준다는 점을 입증했다. 12개 모델 판별에 대해서는 최고 98%의 성능을 보였으며, 14개 모델 판별에 대해서는 최고 97.1%의 성능을 보였다. 33개의 모델 판별은 91.9%의 성능을 보였다.

Bondi et al.[10]은 카메라 판별을 위해 Fig. 3과 같이 CNN과 SVM을 조합한 모델을 연구하였다. 64×64 크기의 Sub-Image를 원본 영상에서 추출하여 학습 데이터로 사용하고 4

계층의 CNN을 통과시켰다. CNN을 통과한 128개의 특징을 각각 1:1로 매칭 후 학습시켜 Binary SVM을 훈련시킨다. 18개 카메라에 대한 판별 성능은 93%를 나타내었다.

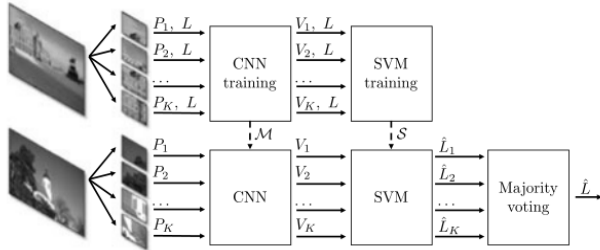


Fig. 3. The Pipeline of Bondi[10]'s CNN and SVM for Camera Model Identification

Freire-Obregón et al.[11]은 딥러닝 기법을 이용하여 모바일 기기의 카메라 식별을 연구했다. MICHE-I Dataset을 이용하여 모바일 디바이스에 대한 판별(디바이스 A, B, C 분류)과 센서에 대한 판별(디바이스 A의 전면카메라, 디바이스 A의 후면카메라 등 카메라 센서 분류), 2가지 실험을 진행하였다. 모델의 구조는 Fig. 4와 같으며 HPF를 거친 영상을 CNN 모델을 통해 학습시켰다. 모바일 디바이스 판별은 3대의 디바이스를 이용한 촬영 영상을 이용하였고 최고 98%의 성능을 보였다. 센서 판별은 5개의 센서에 대한 판별을 시도하였고 91.1%의 성능을 보였다.

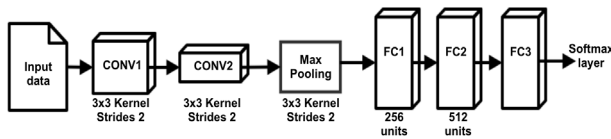


Fig. 4. The Layout of Freire-Obregón et al.[11]'s CNN for Mobile Device Camera Identification

Lee et al.[12]은 딥러닝 기법을 이용하여 프린터 장치 판별 기술을 연구했다. 프린터 장치 판별은 카메라 판별과 같은 영상 포렌식 기법에 속하며 영상에서 장치의 숨겨진 특징을 이용한다는 점에서 유사한 방법을 적용할 수 있다. CNN을 기반으로 입력 데이터에 명암도 동시발생 행렬을 적용한 후 처리하는 모델을 구성하였고 평균 99.8%의 성능을 달성하였다.

명암도 동시발생 행렬(Gray Level Co-occurrence Matrix, GLCM)[13]은 영상 내에서 픽셀 단위에서의 밝기값 관계를 추출하는 방법으로 인접 픽셀의 값을 좌표로 이용하여 같은 값이 반복해서 나오는 횟수를 특징으로 활용하며, 영상의 질감(Texture) 특성을 추출하는데 유리한 방법이다. GLCM의 Feature Map 크기는 원본 영상의 밝기값 범위에 의존하므로 8bit 영상의 경우 256x256의 크기로 생성된다. 또한 R, G, B 채널마다 수행되고 Fig. 5처럼 0°, 45°, 90°, 135° 4가지 방향에 대해 연산하기 때문에 최종적으로 256x256x12 크기의 특징이 추출된다. 이러한 크기는 하드웨어의 제약으로 인해 한 번에

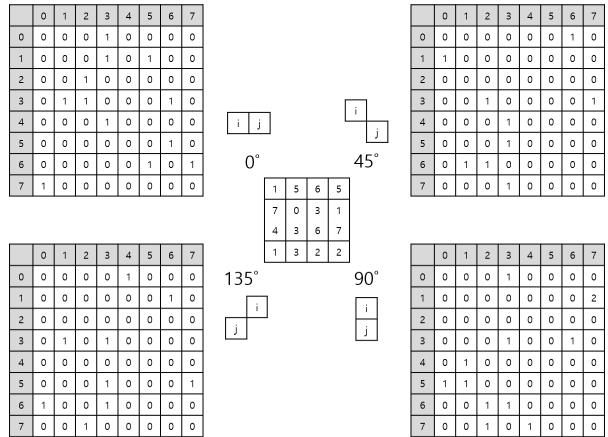


Fig. 5. Gray Level Co-occurrence Matrix Calculation (In Case of 3 Bits Image)

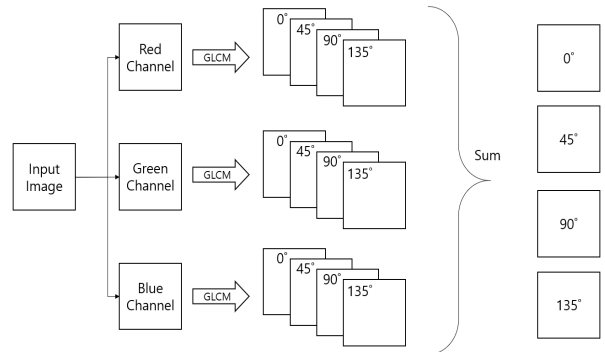


Fig. 6. GLCM Feature Map Processing for Data Reduction

처리할 수 없으므로 Fig. 6과 같은 방식으로 256x256x4 크기로 합산하였다.

Hosler et al.[14]은 사진에 비해 상대적으로 관심이 덜한 동영상에 대한 카메라 식별을 연구하였다. CNN을 이용하여 실험 데이터로는 동영상의 여러 프레임에서 추출한 패치 영상을 사용하였다. CNN 모델은 MISLnet 구조를 기반으로 자체 튜닝한 모델을 사용하였다. I-프레임과 P-프레임을 각각 단일로 사용하였을 때에 비해 두 프레임을 모두 사용하였을 때가 가장 좋은 성능을 보였다. CNN 결과를 fusion 알고리즘을 통해 평가하였고 20개의 카메라를 실험하여 최고 95%의 성능을 나타내었다.

### 3. 딥러닝 기반 카메라 모델 판별 알고리즘

카메라 모델 판별을 위한 딥러닝 모델은 Fig. 7과 같이 학습 과정과 판별 과정으로 나뉘어 진행된다. 학습 과정에서는 학습 영상 데이터와 레이블을 같이 입력하여 가중치를 조절하는 지도학습을 수행한다. 학습 영상 데이터를 입력한 후 모델을 거쳐 나온 추정값이 학습 영상의 레이블과 다를 경우 Convolution Layer의 Kernel 값을 수정하여 추정값과 실제값

의 차이인 Loss를 줄이는 방향으로 가중치를 조절한다. 학습 데이터셋을 지정된 횟수만큼 반복하며 학습을 진행한 후, Loss 값의 감소 추세가 없고 특정 범위에서 진동할 경우 판별 과정으로 넘어간다. 판별 과정은 임의의 모델로 획득한 판별 영상 데이터를 입력한 후, 모델을 거쳐서 나온 추정값이 실제 레이블과 같은지 비교하여 성능을 확인한다.

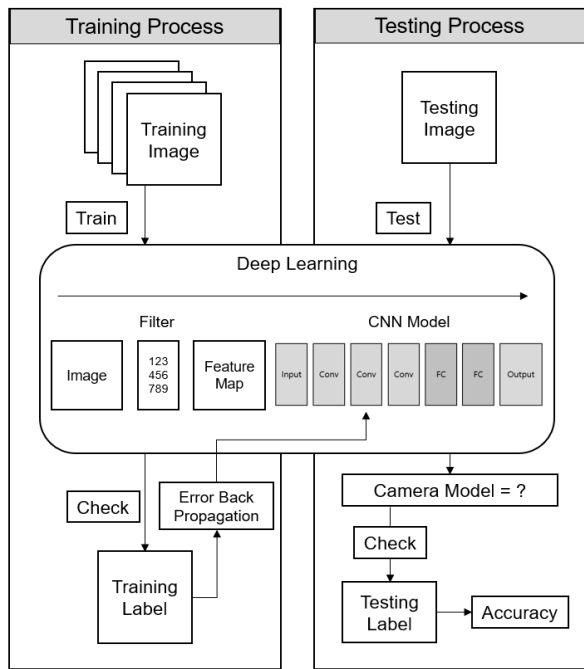


Fig. 7. Training and Testing Process of Proposed Algorithm

제안하는 모델의 구조는 Fig. 8에 나타내었다. 각 과정에 대한 상세한 설명은 다음 절에서 기술한다.

### 3.1 Input layer / Filter

기존 연구들은 대부분 각 영상별 특징을 추출한 후, 통계적 분석을 거친 데이터를 활용하였다. 하지만 딥러닝은 입력 데이터에서 스스로 특징을 추출하기 때문에 특징 추출과정 및 별도의 분석을 거칠 필요가 없다. 하지만 일반적으로 디지털 포렌식을 위한 딥러닝의 경우 원활한 학습과 효율성을 위해 특징을 추출하는 과정까지는 사람이 손을 대는 경우가 많다. 제안하는 모델에서는 HPF를 적용한 특징 데이터를 입력한다. 모델의 Convolution Layer 부분에서 입력한 특징 데이터는 강화되고 Fully Connected Layer 부분에서 특징 사이의 관계를 분석하게 된다.

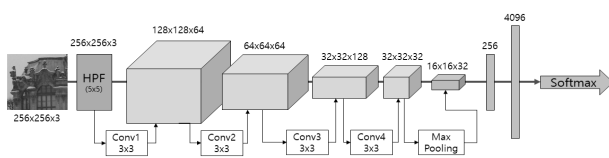


Fig. 8. CNN Model Structure for Camera Identification

### 3.2 Convolution Layer(Conv) / ReLU / Max Pooling

제안하는 모델은 4 계층의 Conv를 가진다. 기존 연구에서 제안한 3 계층의 Conv를 쓰는 모델[6]을 기반으로 다양한 변인을 수정하며 진행하였고 Conv의 수를 조절하여 2 계층, 4 계층과 5계층에 대해서도 실험을 수행하였다. 그 결과, Fig. 9와 같이 4 계층의 Conv가 가장 높은 성능을 보이는 것을 확인할 수 있다. 원활한 비교를 위해 각 계층별 최고 성능이 나온 시점 이후는 최고 성능 값으로 통일하였다.

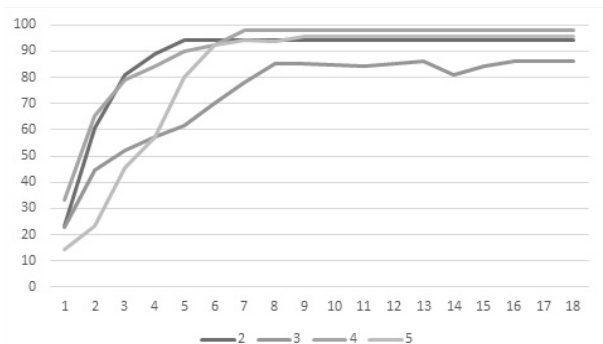


Fig. 9. CNN Model Performance by Number of Convolution Layer

각 Conv는 3x3 크기의 커널에 대해 Convolution을 수행하고 지정된 Strides 값에 따라 Feature Map의 크기를 줄이거나 유지한다. Feature Map의 깊이는 3 계층 모델[6]의 값을 그대로 사용했으며 그 외 계층의 모델은 임의의 값을 설정하여 실험을 수행하여 분석하고, 하드웨어가 감당 가능한 최대 수치를 적용하였다.

Conv를 통과한 값은 비선형 활성화 함수를 거치게 된다. 활성화 함수는 각 계층의 연산을 수행한 후 출력한 값을 다음 계층으로 넘기는 과정에서 정해진 규칙에 따라 값을 변형한다. 만약 활성화 함수를 적용하지 않는다면 중복된 계층이 나열될 경우 여러 계층을 적용하더라도 단일 계층과 같은 방식으로 처리된다. 이를 방지하기 위해 다양한 비선형 함수들이 제시되었고 제안하는 모델에서는 0 이하의 값을 소실시키고 0 초과 값의 값을 그대로 사용하는 ReLU (Rectified Linear Units[14])가 사용되었다.

마지막으로 Pooling을 적용하게 된다. 전체 모델의 처리 과정에서 가장 처리량이 많은 부분이 특징값들을 일렬로 나열하여 관계를 분석하는 Flatten 과정이기에 Pooling을 이용하여 Feature Map의 가로, 세로를 줄임으로써 연산량을 감소시키는 결과를 유도한다. 이 과정에서 Max, Min, Average 등의 규칙을 통해 특징을 강화하는 효과도 얻을 수 있다.

### 3.3 Fully Connected Layer(FC) / DropOut / Output Layer - Softmax

Conv를 거친 데이터는 3차원의 입체 형태의 데이터이다. 이를 FC와 연결하기 위해 먼저 1열로 늘어선 모양의 배열로 변환한다. 그 후 FC에서 입력 노드와 출력 노드를 완전 연결

을 하여 특징 사이의 관계를 분석하고 이어진 FC에서도 반복한다. Drop-Out[15]은 모든 FC에 적용되며 과적합을 회피하기 위해서 학습 후 가중치 갱신 과정에서 의도적으로 몇몇 노드를 누락시켜 비활성화하는 노드의 비율을 의미한다. 제안하는 모델에서는 0.5의 값을 사용하여 학습할 때마다 전체 노드 중 임의로 선택된 절반만큼의 연결 노드를 비활성화한다. 마지막 Output Layer에서는 입력 노드를 판별할 Class와 같은 개수의 출력 노드와 Softmax 연산으로 연결한다. 그 결과, 각 카메라 모델에 대한 Score를 얻게 되고 이를 총합 1에 대한 비율로 변환하여 최고 점수를 얻은 카메라 모델을 판별값으로 출력한다.

### 4. 실험 결과

#### 4.1 실험 환경

카메라 모델 판별을 위한 딥러닝 환경은 Windows 10 Pro를 이용하고 Google에서 배포한 딥러닝 라이브러리인 Tensorflow를 중심으로 구축하였다. 하드웨어는 Intel사의 Core-i7, Nvidia사의 10시리즈를 사용하였다. 하드웨어와 소프트웨어 환경 요약은 Table 1과 Table 2에 나타내었다.

Table 1. Hardware Specification

Hardware	Specification
CPU	Intel i7-7700K
VGA	NVIDIA GTX 1080 (VRAM: 8GB)
RAM	16GB (8GB * 2)

Table 2. Software Environment

Software	Environment
OS	Windows 10 Pro
CUDA/cuDNN	9.0 / 7.2.1
Python	3.6
Tensorflow	1.10 (ver.GPU)

#### 4.2 실험 데이터

실험 데이터는 카메라 기반 디지털 포렌식 기법들에서 공유되는 Dresden Image Database[16]를 이용하였고 해당 영상 데이터베이스의 카메라 모델 및 영상 정보는 Table 3과 같다.

실험은 12개의 카메라 모델을 대상으로 하였으며 하나의 디바이스에 200장 이상의 영상이 있는 모델을 선정하였다. 하나의 모델에서 여러 디바이스가 있는 경우, 첫 번째 디바이스를 사용하였다.

Table 4는 실험에 사용된 영상 획득 장치 및 각 장치의 영상에 대한 정보를 나타내며 Fig. 10은 카메라별 영상의 예시를 제시하였다. 모든 카메라에 대해 같은 영상이 없는 경우, 임의의 사진을 포함하였다.

Table 3. Dresden Image Database Information

Brand (14)	Count	Model (27)	Device	Count
Agfa	1357	DC-504	1	169
		DC-733s	1	281
		DC-830i	1	363
		Sensor505-x	1	172
		Sensor530s	1	372
Canon	979	Ixus55	1	224
		Ixus70	3	567
		PowerShotA640	1	188
Casio	925	EX-Z150	5	925
FujiFilm	630	FinePixJ50	3	630
Kodak	2391	M1063	5	2391
Nikon	2413	CoolPixS710	5	925
		D70	2	369
		D70s	2	367
		D200	2	752
Olympusmju	1040	1050SW	5	1040
Panasonic	931	DMC-FZ50	3	931
Pentax	830	OptioA40	4	638
		OptioW60	1	192
Praktica	1019	DCZ5.9	5	1019
Ricoh	854	GX100	5	854
Rollei	589	RCP-7325XS	3	589
Samsung	1332	L74wide	3	687
		NV15	3	645
Sony	1671	DSC-H50	2	541
		DSC-T77	4	725
		DSC-W170	2	405
Total				16961

Table 4. Test Dataset for Proposed Model

Label	Model	Count	Size
0	Agfa DC-733s	281	3072×2304
1	Agfa DC-830i	363	3264×2448
2	Agfa Sensor530s	372	2560×1920
3	Canon_Ixus55	224	2592×1944
4	Fujifilm FinePix J50	210	3264×2448
5	Kodak M1063	464	3664×2748
6	Nikon D200	372	3872×2592
7	Olympus M1050	204	3648×2736
8	Panasonic DMC-FZ50	265	3648×2736
9	Praktica DCZ 5.9	209	2560×1920
10	Samsung L74wide	232	3072×2304
11	Samsung NV15	217	3648×2736

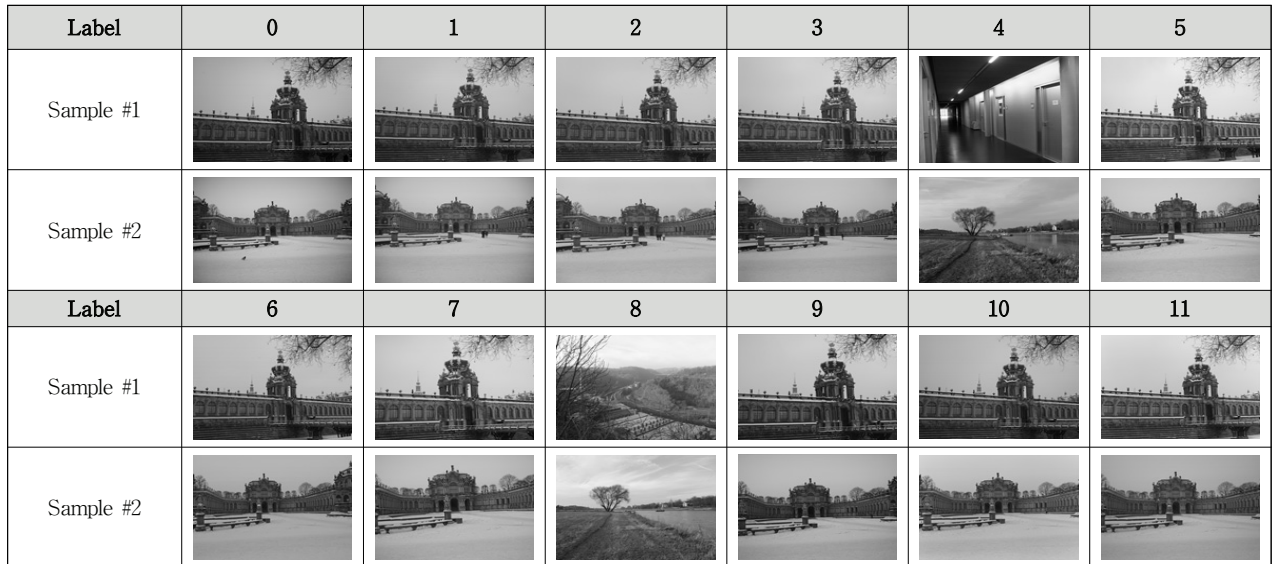


Fig. 10. Sample Image of Test Dataset

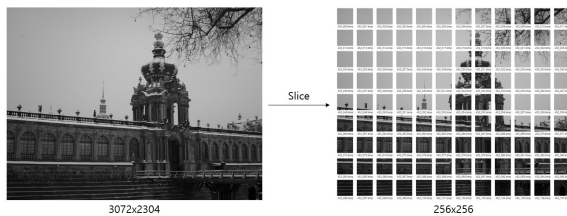


Fig. 11. 256×256 Sample Collection with Sequential Partition

영상 데이터베이스에서 추출한 영상들은 모두 2000×1500 이상의 고해상도이기 때문에 그대로 모델에 사용하기에는 하드웨어의 제약이 따르기 때문에 적절한 데이터 전처리를 수행해야 한다.

먼저, 순차분할 데이터셋을 생성하였고 그 방식은 Fig. 11과 같다. 영상을 256×256 크기로 중복되지 않도록 순차적으로 분할해서 RGB 3채널 영상으로 저장하였다. 256×256 크기보다 작은 부분은 포함하지 않았다. 전체 데이터의 수는 약 40만 장이며 학습 데이터와 판별 데이터는 전체 데이터 대비 80:20의 비율로 구성하여 각각 32만 장, 8만 장 정도의 영상으로 구성되어 있다. 모델별로 촬영 영상의 해상도가 달라서 1장의 영상에서 나오는 256×256 영상의 수는 70장에서 150장의 사이에 있다. 위와 같은 과정을 통해 생성한 영상을 Tensorflow에서 제공하는 TFRECORD 형식으로 저장하여 사용하였다. 해당 데이터셋을 바탕으로 학습을 수행하고 판별 성능을 측정하였다.

#### 4.3 실험 결과 및 분석

본 논문에서 제안하는 카메라 모델 판별 딥러닝 모델의 판별 성능 실험 결과를 기존의 방법들에서 범용적으로 사용하는 3계층 딥러닝 모델[7]과 프린터 판별에서 우수한 성능을 나타낸 GLCM 특징 기반의 딥러닝 모델[12]과 비교한 결과를

Fig. 12와 Table 5에 제시하였다. 그래프에서 x축은 전체 데이터셋의 반복 학습 횟수(epoch)를 의미하고 y축은 판별 데이터셋에 대한 정확도를 나타낸다. 각 모델별 최고 성능이 나온 시점 이후는 최고 성능 값으로 통일하였다.

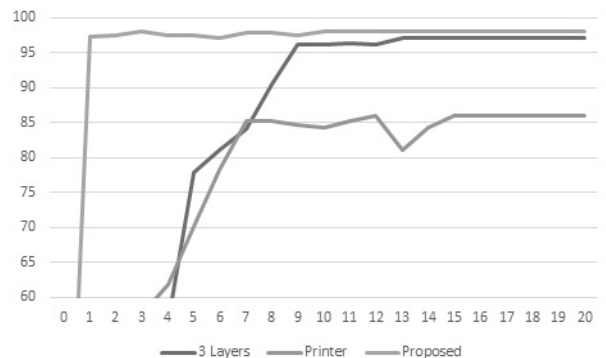


Fig. 12. Comparison of the Proposed Model with Other Models

Table 5. Parameters and Comparison of the Proposed Model with Other Models

Model	3 Layers [7]	Printer[12]	Proposed
Size	256×256		
Dataset	Sequential cropping		
Channel	3		
Filter	HPF	GLCM	HPF
Conv (3×3)	3 (64/64/32)	3 (64/64/32)	4 (64/64/128/32)
Pooling	Max		
FC	2 (256/4096)		
Accuracy	97.1%	86%	98.08%

Table 6. Accuracy Per Epoch of the Proposed Model

Epoch	Accuracy (%)	Epoch	Accuracy (%)
0	10.44	8	97.94
1	97.29	9	97.44
2	97.49	10	98.07
3	98.08	11	96.88
4	97.55	12	97.87
5	97.48	13	95.67
6	97.13	14	96.63
7	97.93	15	90.46
Max Accuracy		98.08	

먼저 기존 실험에서 제시한 값들을 그대로 반영한 모델[7]을 직접 구현하여 실험을 진행하였고 실험 결과는 97.1%로 해당 논문에서 제시한 98%보다는 1% 낮은 성능을 얻었다. 다음으로 최근 연구 중 프린터 장치 판별 연구[12]에서 사용된 모델의 경우, 프린터 판별에서는 99.8%의 높은 성능을 보였으나 카메라 판별을 시도하였을 때는 86%의 성능을 보였다. 프린터의 경우에는 영상에 대한 인쇄 색감 차이가 두드러지게 나타나기 때문에 값의 분포에 대한 특징인 GLCM이 유효하지만 카메라 영상의 경우에는 색감 차이보다는 센서 자체의 흔적(SPN)이 중점이기에 GLCM에서는 그 부분을 잡아 내지 못하는 것으로 보인다.

기존 실험[7]의 모델에서는 컨볼루션 계층을 3번 쌓아서 이용하였지만 2, 3, 4, 5 계층에 대한 실험 결과는 계층 수와 성능은 비례하지 않는다는 사실을 확인하였다. 계층 수에 따른 정확도는 4 > 3 > 5 > 2 순이었으며, 최고 정확도에 도달하는 시간은 2 < 4 < 5 < 3 순으로 성능과 속도 모두 계층 수와는 관계가 없는 것으로 나타났다. 실험의 목적과 데이터

셋에 따라 항상 다양한 실험을 해서 최적의 모델을 찾아야 한다는 점을 확인하였다.

제안한 모델의 학습 횟수에 따른 정확도는 Table 6에 나타내었고, 최고 성능이 나온 시점에서 각 카메라 모델에 대한 정확도는 Table 7에 정리하였다.

### 5. 결 론

최근 초소형 카메라 등을 이용한 몰래카메라 범죄가 사회적 문제로 대두되고 있다. 이러한 범죄를 방지하고 범죄가 발생할 경우, 범인을 찾기 위해선 카메라 모델 판별 기술이 필수적이다. 수사 과정에서 제 3의 피해자가 없도록 하고 범죄자가 아닌 사람이 누명을 쓰는 일은 없어야 한다. 이를 위해서는 높은 정확도를 가진 기술이 꼭 필요하다.

본 논문에서는 카메라 모델 판별을 위한 딥러닝 모델을 제안하였다. 기존 연구를 기반으로 구현한 모델을 확장하여 실험을 진행하였다. 계층의 변화가 성능에 영향을 끼친다는 점을 다양한 계층의 모델을 활용한 시험을 통해 확인하고 조금이나마 성능이 더 높은 모델을 제안하였다. 또한 디지털 포렌식 중 유사한 분야인 프린터 판별 관련 모델을 차용하여 HPF와 GLCM, 2가지 특징에 대해 실험을 하였으며 HPF가 카메라 모델 판별에는 조금 더 적합하다는 것을 보였다.

차후 연구목표로는 SPN 연구에서 사용되는 위치 동기화 개념을 바탕으로 한 데이터셋을 구축하여 딥러닝 모델을 통한 실험을 진행할 것이다. 또한 SPN 기반이 아닌 JPEG 압축 과정 중에 드러나는 8x8 블록 단위 양자화에 의한 격자성을 바탕으로 카메라 모델 판별을 위한 추가적인 실험을 시도할 것이다.

Table 7. Camera Identification Accuracy for Each Camera Using the Proposed Model

		Prediction												Accuracy
		0	1	2	3	4	5	6	7	8	9	10	11	
Truth	0	5709	94	1	2	1	5	1	3	2	183	44	24	94.07
	1	284	7358	119	0	0	7	57	2	0	3	4	6	93.85
	2	5	59	11001	0	3	38	8	0	0	0	0	2	98.97
	3	15	2	0	3097	4	9	1	2	0	3	2	0	98.79
	4	0	1	1	0	4520	12	1	0	0	0	0	0	99.67
	5	18	27	4	0	0	12897	33	1	0	8	0	3	99.28
	6	4	11	58	0	0	12	11071	1	0	0	0	2	99.21
	7	0	0	0	0	0	16	31	5657	0	0	0	7	99.05
	8	11	11	0	0	13	0	0	1	7374	9	0	0	99.39
	9	5	94	0	8	1	12	1	2	0	2738	46	18	93.61
	10	3	1	1	0	0	10	1	1	0	37	4934	1	98.90
	11	1	3	0	1	0	38	16	9	0	9	0	5998	98.73



References

[1] M. Kharrazi, H.-T. Sencar, and N. Memon, "Blind source camera identification," *Proceedings of the International Conference on Image Processing*, Vol.1, pp.709-712, 2004.

[2] S. Bayram, H. Sencar, N. Memon, and I. Avcibas, "Source camera identification based on CFA interpolation," *Proceedings of the IEEE International Conference on Image Processing*, Vol.3, pp.III-69, 2005.

[3] A. Popescu and H. Farid, "Exposing Digital Forgeries by Detecting Traces of Re-sampling," *IEEE Transactions on Signal Processing*, Vol.53, No.2, 2005.

[4] K.-S. Choi, E.-Y. Lam, and K.-K. Wong, "Source camera identification using footprints from lens aberration," *Proceedings of SPIE, Digital Photography II*, Vol.6069, pp.6069J, 2006.

[5] J. Lukas, J. Fridrich, and M. Goljan, "Digital camera identification from sensor pattern noise," *IEEE Transactions on Information Forensics and Security*, Vol.1 No.2, pp.205-214, 2006.

[6] K. Bolouri, A. Azmoodeh, A. Dehghantanha, and M. Firouzmand, "Internet of things camera identification algorithm based on sensor pattern noise using color filter array and wavelet transform," In *Handbook of Big Data and IoT Security*, Springer, Cham, pp.211-223, 2019.

[7] A. Tuama, F. Comby, and M. Chaumont, "Camera model identification with the use of deep convolutional neural network," *Proceedings of the IEEE International Workshop on Information Forensics and Security*, pp.1-6, 2016.

[8] A. Krizhevsky, I. Sutskever, and G.-E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, pp. 1097-1105, 2012.

[9] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S.-E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-9, 2015.

[10] L. Bondi, L. Baroffio, D. Güera, P. Bestagini, E.-J. Delp, and S. Tubaro, "First steps toward camera model identification with convolutional neural networks," *IEEE Signal Processing Letters*, Vol.24, No.3, pp.259-263, 2017.

[11] D. Freire-Obregón, F. Narducci, S. Barra, and M. Castrillón-Santana, "Deep learning for source camera identification on mobile devices," *Pattern Recognition Letters*, Vol.126, pp.86-91, 2018.

[12] S.-H. Lee and H.-Y. Lee, "Printer Identification Methods Using Global and Local Feature-Based Deep Learning,"

*KIPS Transactions on Software and Data Engineering*, Vol. 8, No.1, pp.37-44, 2019.

[13] J.-Y. Baek, H.-S. Lee, S.-G. Kong, J.-H. Choi, Y.-M. Yang, and H.-Y. Lee, "Color Laser Printer Identification through Discrete Wavelet Transform and Gray Level Co-occurrence Matrix," *The KIPS Transactions: PartB*, Vol.17, No.3, pp 197-206, 2010.

[14] B. Hosler, O. Mayer, B. Bayar, X. Zhao, C. Chen, J.-A. Shackelford, and M.-C. Stamm, "A Video Camera Model Identification System Using Deep Learning and Fusion," In *ICASSP 2019-2019 IEEE International Conference on Acoustics*, pp.8271-8275, 2019.

[15] V. Nair and G.-E. Hinton, "Rectified linear units improve restricted boltzmann machines," *Proceedings of the International Conference on Machine Learning*, pp.807-814, 2010.

[16] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, Vol.15, No.1, pp.1929-1958, 2014.

[17] Dresden Image Database, [Internet], <http://forensics.inf.tu-dresden.de/ddimgdb/>



이수현

<https://orcid.org/0000-0002-3372-5660>

e-mail : dark0487@kumoh.ac.kr

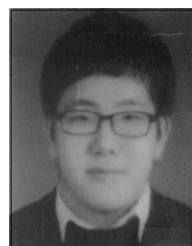
2018년 금오공과대학교

컴퓨터소프트웨어공학과(학사)

2018년~현 재 금오공과대학교

소프트웨어공학과 석사과정

관심분야 : Image Processing, Deep Learning



김동현

<https://orcid.org/0000-0002-0693-431X>

e-mail : knight2995@naver.com

2016년 금오공과대학교

컴퓨터소프트웨어공학과(학사)

2016년~현 재 금오공과대학교

소프트웨어공학과 석사과정

관심분야 : Image Processing, Digital Forensics, Deep Learning



## 이 해 연

<https://orcid.org/0000-0002-6081-1492>

e-mail : haeyeoun.lee@kumoh.ac.kr

1997년 성균관대학교 정보공학과(학사)

1999년 한국과학기술원 전산학과(공학석사)

2006년 한국과학기술원 전자전산학과  
(공학박사)

2008년~현재 금오공과대학교 컴퓨터소프트웨어공학과 교수

관심분야: IoT, Image Processing, Digital Forensics