Journal of KIIT. Vol. 17, No. 8, pp. 11-19, Aug. 31, 2019. pISSN 1598-8619, eISSN 2093-7571
http://dx.doi.org/10.14801/jkiit.2019.17.8.11

변형된 DenseNet과 HPF를 이용한 카메라 모델 판별 알고리즘

이수현*[1], 김동현*[2], 이해연**

Camera Model Identification Using Modified DenseNet and HPF

Soo-Hyeon Lee*[1], Dong-Hyun Kim*[2], and Hae-Yeoun Lee**

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).

요 약

영상 관련 범죄가 증가하고 고도화됨에 따라서 고수준의 디지털 포렌식 기술이 요구된다. 그러나 기존의 특징 기반 기술은 인간이 고안한 특징을 활용함으로서 새로운 기기 특징에 쉽게 대응하기 어렵고, 딥러닝 기반 기술은 정확도 향상이 요구된다. 본 논문에서는 딥러닝 모델 분야의 최신 기술인 DenseNet을 기반으로 카메라 모델 판별을 위한 딥러닝 모델을 제안한다. 카메라의 센서 특징을 획득하기 위해 HPF 특징 추출 필터를 적용하였고, 카메라 판별에 적합하도록 기존 DenseNet에서 계층 반복 수를 조정하였다. 또한 연산량을 줄이기 위한 Bottleneck layer와 압축 연산 처리를 제거하였다. 제안한 모델을 Dresden 데이터베이스를 사용하여 성능 분석을 하였고, 14개 카메라 모델에 대해 99.65%의 정확도를 달성하였다. 기존 연구들보다 높은 정확도를 달성하였으며 기존에 동일한 제조사에서 정확도가 낮아지는 단점을 극복하였다.

Abstract

Against advanced image-related crimes, a high level of digital forensic methods is required. However, feature-based methods are difficult to respond to new device features by utilizing human-designed features, and deep learning-based methods should improve accuracy. This paper proposes a deep learning model to identify camera models based on DenseNet, the recent technology in the deep learning model field. To extract camera sensor features, a HPF feature extraction filter was applied. For camera model identification, we modified the number of hierarchical iterations and eliminated the Bottleneck layer and compression processing used to reduce computation. The proposed model was analyzed using the Dresden database and achieved an accuracy of 99.65% for 14 camera models. We achieved higher accuracy than previous studies and overcome their disadvantages with low accuracy for the same manufacturer.

Keywords
camera model identification, deep learning, DenseNet, high-pass filter

* 금오공과대학교 소프트웨어공학과
- ORCID[1]: https://orcid.org/0000-0002-3372-5660
- ORCID[2]: https://orcid.org/0000-0002-0693-431X
** 금오공과대학교 컴퓨터소프트웨어공학과 교수 (교신저자)
- ORCID: http://orcid.org/0000-0002-6081-1492

· Received: Jun. 30, 2019, Revised: Aug. 08, 2019, Accepted: Aug. 11, 2019
· Corresponding Author: Hae-Yeoun Lee
Dept. of Computer Software Engineering, Kumoh National Institute of Technology, Korea
Tel.: +82-54-458-7548, Email: haeyeoun.lee@kumoh.ac.kr

Ⅰ. 서 론

고성능 카메라를 이용한 촬영과 카메라에서 획득한 영상에 대한 편집은 전문가들만이 가능한 분야였다. 하지만 점점 개인이 쉽게 사서 다룰 수 있는 카메라의 성능이 점점 고도화되었고, 일반인도 쉽게 고품질의 영상을 획득할 수 있게 되었다. 이에 전문가가 아닌 사람도 영상을 쉽게 편집할 수 있는 프로그램들이 많이 개발되었다. 이러한 변화는 참신하고 새로운 콘텐츠의 생성을 유도하게 되었다.

하지만 카메라의 발달과 편집 프로그램의 범용화로 영상 관련 범죄를 저지르는 사람들이 쉽게 범죄 도구를 구할 수 있게 되었고 실제로 다양한 범죄가 발생하고 있다[1]. 대표적으로 영상 소유자의 입장에서는 올린 영상을 다른 사람이 다운받아 소유자에 대한 정보를 삭제하고 다른 곳에 업로드 하는 방식으로 저작권 및 소유권을 침해받을 수 있다. 영상에 찍힌 사람의 입장에서는 카메라의 소형화에 따라 자신도 모르게 찍힌 영상이 웹을 통해 유포되는 등 초상권을 침해받을 수 있다. 실제로 몰래 카메라 등의 범죄는 현 사회의 심각한 문제로 대두되고 있다.

전자기기를 사용하여 일어난 범죄에 대해 수사를 하는 과정에서 디지털 포렌식 기술이 필요하다. 디지털 포렌식은 디지털 정보에 관한 범죄를 수사하기 위한 기법으로 영상, 동영상, 음성 등 다양한 분야에서 연구되고 있다.

디지털 포렌식에서 카메라 모델 판별은 꾸준한 연구가 요구되는 분야이며 실제로 범죄 수사 중 증거 판별, 위조 영상 판별 등의 과정에 큰 영향을 끼친다. 이전에는 영상 파일에 자동으로 기록되는 정보를 이용할 수 있었지만, 현재는 편집 프로그램을 이용해서 메타 데이터를 지울 수 있기에 영상의 픽셀 데이터만을 이용하여 카메라 모델을 특정할 수 있는 기술이 필요하다[1].

본 논문에서는 딥러닝 모델 중 합성곱 신경망(Convolution Neural Network, CNN) 분야 최신 기술인 DenseNet[2]에 기반하는 카메라 모델을 판별하는 딥러닝 모델을 제안한다. 카메라 센서 정보를 획득하기 위해 특징 추출 필터를 도입하였고, 카메라 판별에 적합하도록 DenseNet 계층을 조정하고 파라미터를 설정하였다. 성능 평가를 위하여 공개 카메라 데이터셋인 Dresdren[3]에서 14개 카메라 모델을 사용하여 제안하는 모델을 학습시킨 후, 검증하는 과정으로 실험을 수행하여 99% 이상의 높은 성능을 달성하였고 그 결과를 분석하였다.

본 논문의 구성을 다음과 같다. 2장은 영상을 이용한 카메라 모델 판별에 관련된 국내외의 기존 연구를 제시한다. 3장에서는 본 논문에서 사용된 딥러닝 모델에 대해 상세히 기술하고 제안하는 카메라 모델 판별 알고리즘에 대해 설명한다. 4장에서는 실험 환경과 실험 결과, 그리고 기존 연구들과의 비교를 통해 결과에 대해 분석한다. 끝으로 5장에서는 결론 및 향후 연구 방향에 대해 기술한다.

Ⅱ. 관련 연구

영상을 획득한 장치에 대한 정확한 판별은 포렌식 분야에서 꾸준히 요구되어 온 기술이다. 그와 관련해서 다양한 연구들이 진행되었고 다양한 실험을 통해 뛰어난 성능을 보이는 연구들이 계속 진행되고 있다.

2.1 특징 기반 카메라 판별

카메라 판별을 위해 영상이 생성되는 과정에서 발생하는 고유한 흔적을 찾기 위한 연구들이 진행되었다.

Bayram et al.[4]은 Color Filter Array(CFA)를 통과한 영상 데이터 사이에서 일어나는 보간 작업이 모델마다 다르다는 점을 이용하여 연구를 수행하였다. Choi et al.[5]은 동일한 공정에서 생산된 렌즈라도 고유한 왜곡이 생기는 점에서 착안하여 카메라를 판별하는 연구를 진행하였다. Tomioka et al.[6]는 하나의 카메라 안에 있는 수많은 이미지 센서가 같은 밝기에도 동일하게 반응하지 않기에 발생하는 Photo Response Non-Uniformity(PRNU) 특징을 이용하여 연구를 수행하였다. 이 외에도 High-Pass Filter(HPF), Wiener Filter, Gray Level Co-Occurrence Matrix(GLCM) 등 다양한 특징 값을 이용해 카메라를 판별하는 연구[1][7][8]들이 수행되고 있다.

2.2 딥러닝 기반 카메라 판별

딥러닝은 여러 번의 암흑기가 있던 인공신경망 기술이 기존의 한계를 극복함에 따라 급속도로 사회 전반으로 퍼지게 되었다.

가장 기본적인 심층신경망(Deep Neural Network, DNN)을 뿌리로 삼아 학습할 데이터의 종류와 풀어야 하는 문제의 종류에 따라 다양한 모델이 제안되고 있다. 데이터 형식을 나누면 N차원 데이터와 시계열 데이터로 나뉘며, 문제의 종류는 분류, 생성, 검출 등이 있다. 카메라 판별과 관련 있는 분야는 N차원 데이터-분류에 해당하여 2차원 데이터, 즉 영상 데이터는 CNN 계열의 모델을 적용하였을 때, 높은 성능을 보인다는 것이 앞선 연구들을 통해 밝혀졌다. CNN은 DNN의 앞에 지역적 특징 추출을 위한 합성곱(Convolution) 연산 계층이 존재하는 것이 특징이다. CNN 계열 모델 중 분류/판별 분야에 관해 대표적인 모델은 AlexNet[9], GoogleNet[10], ResNet[11], DenseNet[2]이 있다. 카메라 판별 관련 연구는 이를 응용하는 연구들이 진행되고 있다.

Baroffio et al.[12]은 기본적인 CNN 구조인 합성곱 연산 계층 3개와 완전 연결 계층 2개를 연결하여 모델을 구성하여 카메라 디바이스 단위의 판별 연구과 모델 단위의 판별 연구를 진행하였다. 디바이스 단위에서는 30%의 낮은 정확도를 보이며 딥러닝 모델을 이용한 카메라 판별에서 디바이스 단위 판별은 난이도가 높다는 것을 보였다. Tuama et al.[13]는 기본적인 CNN 구조의 모델과 기존에 연구되었던 특징 기반 카메라 판별의 성능 비교를 진행하였다. Kuzin et al.[14]은 위에서 언급한 DenseNet을 응용하여 10대의 카메라를 판별 연구를 진행하였고 Rafi et al.[15]는 특징을 추출하기 위한 HPF나 Wiener Filter 같은 필터를 사용하지 않고 영상 자체의 차이값을 사용하는 RemNet을 제안하였다. 또한 널리 알려진 DenseNet을 여러 영상 크기에 적용해서 합친 후 판별하는 연구도 진행하였다[16]. Yao et al.[17]는 카메라 모델 판별 정확도를 올리기 위해 최종 결과 산출을 각 영상의 조각들로 다수결을 진행하여 높은 정확도를 보이는 방법을 제안하였다. 또한 영상에서 피사체가 위치할 가능성이 높은 부분을 사용하는 방법을 이용하였다.

위에서 제시한 대부분의 연구는 DSLR 카메라를 대상으로 진행하였지만 최근 스마트폰의 보급으로 스마트폰 카메라에 대한 연구[18][19]들 또한 활발히 진행되고 있으며 영상이 아닌 동영상을 대상으로 하는 연구[20]도 진행되고 있다.

III. 카메라 모델 판별 알고리즘

3.1 카메라 판별 딥러닝 알고리즘

제안하는 카메라 모델 판별 알고리즘은 지도학습의 방식을 따르며 그림 1과 같이 학습과 검증의 단계를 거친다.

학습 과정에서는 학습 영상과 정답을 모델에 입력한 후, 모델이 예측한 값과 정답을 비교하여 loss 값을 계산한다. 틀린 결과를 학습 과정에 반영하는 오류 역전파 과정을 통해 이 loss 값을 낮추는 방향으로 모델의 가중치를 수정한다. 충분한 횟수의 학습이 이루어져서 loss 값의 변화가 극히 적어지면 학습을 중단하고 검증을 시작한다. 검증 단계에서는 모델에 검증 영상만을 입력하여 모델이 예측한 값과 정답을 비교하여 판별 정확도를 산출한다.

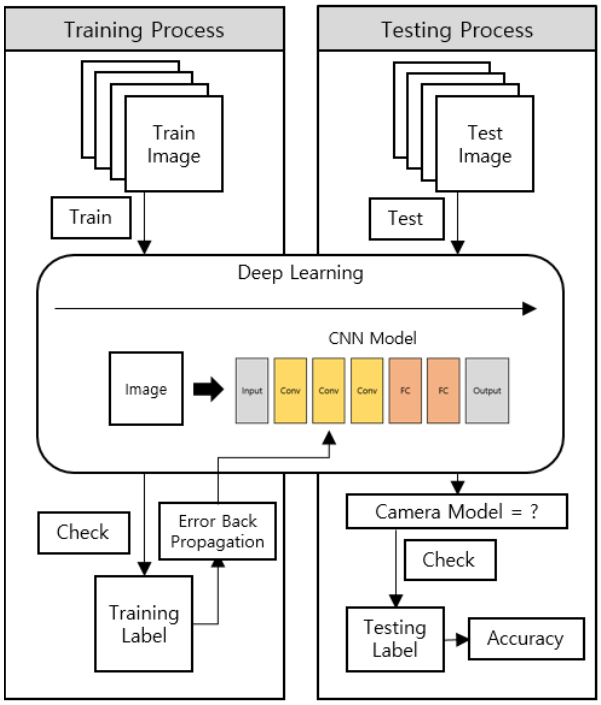


그림 1. 딥러닝 알고리즘
Fig. 1. Algorithm of deep learning

3.2 DenseNet

제안하는 알고리즘의 딥러닝 모델은 DenseNet으로서 카메라 모델 판별에 적합하도록 변형하였다. Huang et al.이 제안한 DenseNet 딥러닝 모델의 일반적인 CNN 모델과의 차이점은 그림 2와 같다[2].

기존의 CNN 모델은 합성곱 연산을 수행하면 이전 값을 고려하지 않는다. 합성곱 연산의 출력값을 이용하여 다음 계층의 입력값으로 쓰기 때문이다. 하지만 DenseNet은 Dense block이라는 개념을 도입해서 이 문제를 해결했다. Dense block 안의 Feature map은 모두 같은 크기를 가지고 Dense block의 출력값에 모두 합쳐서 나가게 된다. 이를 통해 초기 정보를 유의미한 수준으로 유지하며 학습을 계속할 수 있다. 그림 2의 Dense block은 layer 반복 횟수가 4로 설정된 예시이다.

초기 정보를 모두 가지고 있는다는 것은 학습에 유리한 것이 사실이지만 이는 반대로 연산량이 증가하여 학습 시간 및 필요 하드웨어 사양이 높아지는 결과를 초래할 수 있는 요소이다. DenseNet에서는 이를 막기 위해 BottleNeck layer와 압축의 개념을 도입하였다. BottleNeck layer은 합성곱 연산 중간에 Conv 1x1 연산을 집어넣어 차원 수를 줄이고 본래 진행하려던 연산을 수행한 후, 다시 Conv 1x1 연산을 통해 본래 채널 수로 돌아오게 한다. 이 과정을 거치면 바로 연산을 수행할 때보다 파라미터의 수가 줄어들게 되고 곧 연산량이 줄어드는 효과를 낸다. 압축 또한 연산량을 줄이기 위한 기법으로 Dense block 사이에 위치하는 Transition layer에서 채널의 수를 압축 계수만큼 줄이는 과정을 말한다.

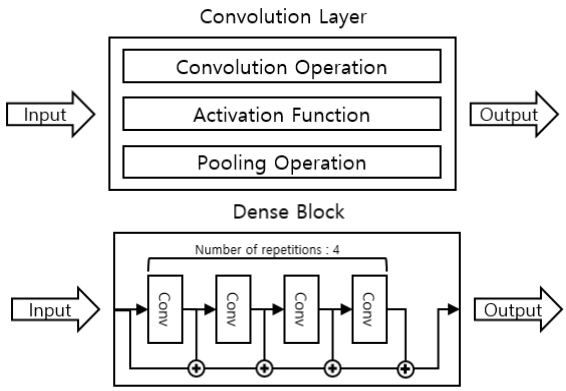


그림 2. 기존의 CNN과 DenseNet 비교
Fig. 2. Comparison between general CNN and DenseNet

3.3 카메라 모델 판별 DenseNet 설계

DenseNet을 기반으로 데이터에 적합하게 조정한 모델을 사용한다. 제안하는 알고리즘 모델의 구조는 그림 3과 같고 하이퍼 파라미터는 표 1과 같다.

입력 영상에 먼저 3채널 각각에 대해 HPF를 통과시킨다. 영상의 크기와 학습 소요시간을 고려하였을 때, 연산량을 충분히 감당 가능한 실험 환경을 가정하여 모델에 Bottleneck layer 부분과 압축 부분을 제거하였다.

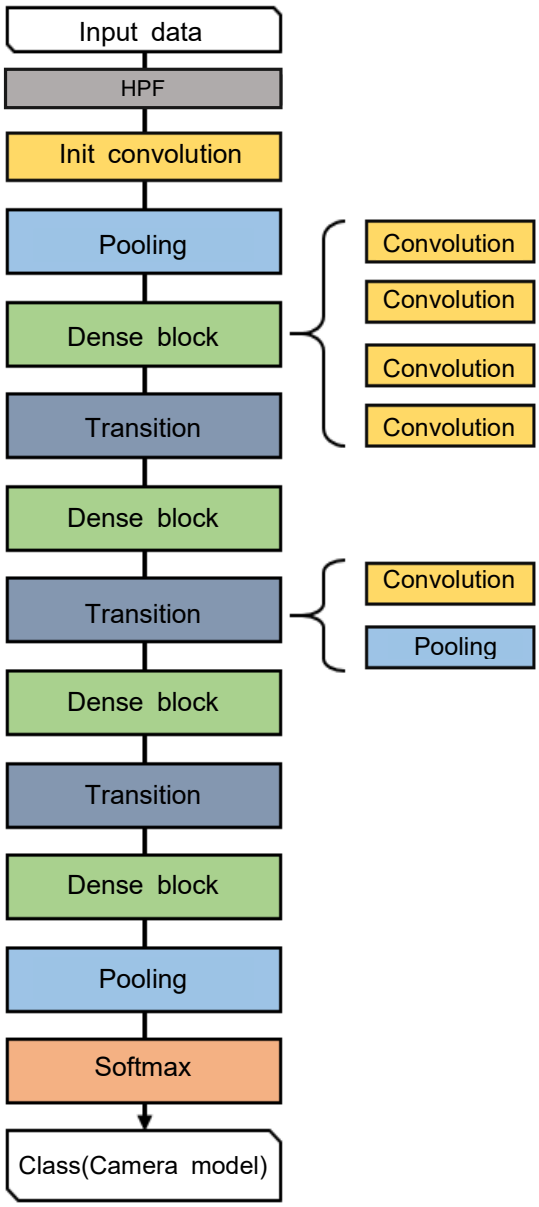


그림 3. 제안하는 알고리즘의 모델 구조
Fig. 3. Model structure of the proposed algorithm

표 1. 제안하는 모델의 하이퍼 파라미터
Table 1. Hyper parameters of proposed model

Layer	Output size	Modified DenseNet (k=32)
Input	256×256	3 channel HPF conv
Convolution	128×128	7×7 conv, strides 2
Pooling	64×64	3×3 max pool, strides 2
Dense block (1)	64×64	(3×3 conv, strides 1) × 4
Transition	64×64	1×1 conv
	32×32	2×2 avg pool, strides 2
Dense block (2)	32×32	(3×3 conv, strides 1) × 4
Transition	32×32	1×1 conv
	16×16	2×2 avg pool, strides 2
Dense block (3)	16×16	(3×3 conv, strides 1) × 4
Transition	16×16	1×1 conv
	8×8	2×2 avg pool, strides 2
Dense block (4)	8×8	(3×3 conv, strides 1) × 4
Classification	1×1	8×8 grobal avg pool
		fully-connected, softmax

또한 기존 연구 사례를 살펴보면 영상 판별의 경우, 깊은 모델이 필요 없기에 각 Dense block의 반복 횟수를 4로 통일하였다. 학습률은 1e-4로 시작하여 4 epoch마다 1/10을 곱하였다.

Ⅳ. 실험 환경 및 결과

4.1 실험 환경

본 논문에서 제안한 모델을 실험하기 위한 하드웨어 사양으로는 CPU는 Intel의 7세대 i7을 사용하였고 GPU는 11GB의 V-RAM을 가진 NVIDIA의 GTX 1080 Ti를 사용하였다. RAM은 8GB 모듈 2개로 구성하여 총 16GB이다. 운영체제는 윈도우 10 RS4, 구현 언어는 Python 3.6, 프레임워크는 코드가 간단하고 빠른 모델 구성 및 조정이 가능한 Keras(2.2.4)를 사용하였다. Keras의 Backend 프레임워크는 Tensorflow(1.13)를 사용했다.

4.2 실험 데이터

실험을 위한 데이터셋은 공개형 데이터베이스인 Dresden image database[3]를 이용하였다. 표 2는 Dresden에 포함된 카메라의 목록을 나타낸다.

표 2. Dresden image database 카메라 목록
Table 2. List of camera in dresden image database

Brand	Model	Device index	Total
Agfa	DC-504	0	169
	DC-733s	0	281
	DC-830i	0	363
	Sensor505-x	0	172
	Sensor530s	0	372
Canon	Ixus55	0	224
	Ixus70	0 1 2	567
	PowerShotA640	0	188
Casio	EX-Z150	0 1 2 3 4	925
FujiFilm	FinePixJ50	0 1 2	630
Kodak	M1063	0 1 2 3 4	2391
Nikon	CoolPixS710	0 1 2 3 4	925
	D70	0 1	369
	D70s	0 1	367
	D200	0 1	752
Olympusmju	1050SW	0 1 2 3 4	1040
Panasonic	DMC-FZ50	0 1 2	931
Pentax	OptioA40	0 1 2 3	638
	OptioW60	0	192
Praktica	DCZ5.9	0 1 2 3 4	1019
Ricoh	GX100	0 1 2 3 4	854
Rollei	RCP-7325XS	0 1 2	589
Samsung	L74wide	0 1 2	687
	NV15	0 1 2	645
Sony	DSC-H50	0 1	541
	DSC-T77	0 1 2 3	725
	DSC-W170	0 1	405

총 14개의 카메라 브랜드, 27개의 카메라 모델, 74개의 카메라 디바이스가 포함된 데이터셋은 약 17,000장의 영상을 포함하고 있다. 모든 영상을 사용하여 학습을 진행하기에는 카메라 장치별, 디바이스별 데이터의 불균형이 심각하기 때문에 적절한 기준을 두어 실험을 진행하였다. 각 장치마다 여러 디바이스를 포함하지만 단일 디바이스에 포함된 영상이 200장이 넘는 14개의 카메라 모델만을 골라서 실험을 진행하였다. 표 3은 데이터셋 중 사용한 카메라 모델과 각 카메라의 영상 수를 나타내며 그림 4는 실험 영상의 예시이다. 동일한 영상이 없는 경우, 임의의 영상으로 대체하였다.

데이터베이스의 영상 크기는 3000×2000 내외이다. 하드웨어의 한계 상, 원본 크기 그대로 영상을 입력할 수 없으므로 영상 크기를 작게 할 필요가 있다.

표 3. 실험에 사용한 카메라 목록
Table 3. List of camera used in experiments

Brand	Model	Size	Total	Label
Agfa	DC-733s	3072x2304	200	0
	DC-830i	3264x2448	200	1
	Sensor530s	2560x1920	200	2
Canon	Ixus55	2592x1944	200	3
FujiFilm	FinePixJ50	3264x2448	200	4
Kodak	M1063	3664x2748	200	5
Nikon	D200	3872x2592	200	6
Olympusmju	1050SW	3648x2736	200	7
Panasonic	DMC-FZ50	3648x2736	200	8
Praktica	DCZ5.9	2560x1920	200	9
Samsung	L74wide	3072x2304	200	10
	NV15	3648x2736	200	11
Sony	DSC-H50	3456x2592	200	12
	DSC-W170	3648x2736	200	13

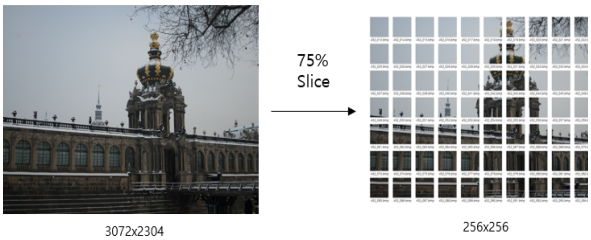


그림 5. 75% 면적에 대한 중복된 부분이 없는 분할
Fig. 5. Non-overlap slice on 75% area of image

영상에서 피사체가 위치할 확률이 높은 영상의 중앙 면적 75%을 영상의 내용이 중복되지 않도록 그림 5와 같이 256×256 크기로 순차 분할하여 데이터셋을 만들었다. 전체 데이터셋에서 학습과 검증 데이터의 비율은 8:2로 설정하여 대략 200,000장 : 50,000장 정도의 영상을 이용했다.

4.3 실험 결과 및 분석

본 논문에서 제안하는 카메라 모델 판별 알고리즘의 실험 결과를 그림 6과 표 4에 나타내었다. 그림의 가로축은 학습의 반복 횟수를 나타내는 epoch이며 세로축은 정확도를 나타낸다.

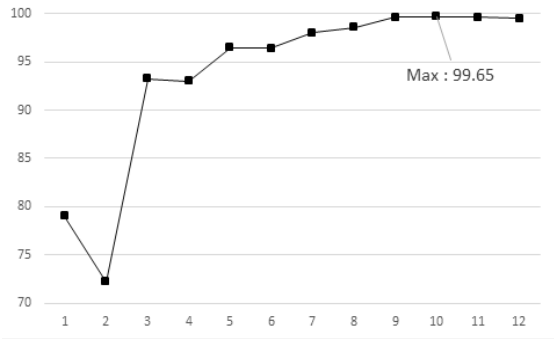


그림 6. 제안하는 모델의 학습 횟수에 따른 정확도
Fig. 6. Accuracy per epoch of proposed model

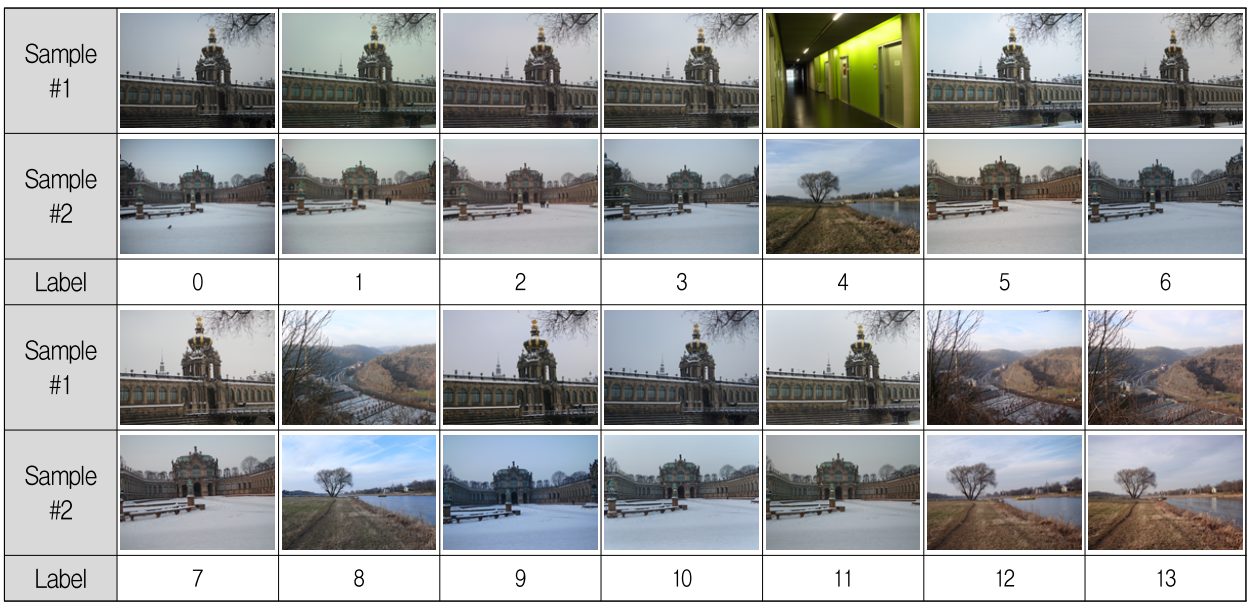


그림 4. 실험에 사용하는 카메라 모델의 예시 영상
Fig. 4. Sample images of camera models used in experiments

표 4. 제안하는 모델의 학습 횟수에 따른 정확도
Table 4. Accuracy per epoch of the proposed model

Epoch	Accuracy(%)	Epoch	Accuracy(%)
1	79.04	7	97.98
2	72.19	8	98.56
3	93.26	9	99.59
4	92.97	10	**99.65**
5	96.44	11	99.63
6	96.41	12	99.52

표 5. Dresden 영상을 사용하는 연구들과의 비교
Table 5. Comparison with studies using dresden images

Model	Camera	Accuracy(%)
Baroffio[12]	27	72.9
Tuama[13]	12	98
Rafi[15]	18	97.03
Kamal[16]	19	over 99
Proposed-14	**14**	**99.65**
Proposed-12	**12**	**99.62**

최고 성능은 99.65%였으며 12 epoch 이후로는 학습 데이터셋에 대한 정확도는 미미하게 올라갔지만 검증 데이터셋에 대한 정확도가 급락하는 과적합의 양상을 보였다. 실험 초기의 학습률은 1e-4로 Adam Optimizer를 그대로 사용하였지만 50 epoch 이상의 학습에도 성능이 96~97% 수준에서 진동하였다. 이에 학습률의 유지 구간을 좁혀서 loss값의 빠른 수렴을 유도하였다. 그 결과 12 epoch 정도의 학습에도 100%에 가까운 높은 정확도를 얻었다.

기존 연구 중 Dresden image database를 사용한 실험들과 제안하는 모델을 비교하였고 그 내용은 표 5와 같다.

다양한 연구들과 비교해서 높은 성능을 보임을 확인할 수 있다. 동일한 DenseNet을 기반으로 응용하여 설계한 모델을 실험한 Kamal et al.[16]의 연구에서는 동일한 제조사의 카메라에 대해 판별 성능이 낮아지는 단점이 있다. 본 논문에서 제안하는 모델은 제조사와 관련 없이 98.88%인 카메라 하나를 빼고는 모두 99% 이상의 성능을 보여주고 있다. 또한 Tuama et al.[13]와 동일한 데이터를 이용하여 진행한 실험(Proposed-12)의 실험 결과 또한 기존의 연구보다 높은 성능을 보인다는 것을 나타내고 있다. Proposed-14 모델의 실험에서 각 카메라에 대한 오차 행렬은 표 6에 나타냈다.

최근에 스마트폰이 범용적으로 사용되고 있으나 Dresden image database에 포함되지 않아서 실험에는 포함하지 않았다. 그러나 스마트폰 영상이 카메라와 그 특성이 크게 차이가 나지 않아서 제안한 모델에 적용하더라도 성능 차이가 없을 것으로 예상되며 Lee[1] 및 Freire-Obregón et al.[18]에서도 카메라 영상과 스마트폰 영상에 대한 차별성이 크지 않음을 확인할 수 있다.

표 6. 제안한 모델의 검증 데이터셋에 대한 오차 행렬 (14개 카메라)
Table 6. Error matrix for the test data set of the proposed model (14 cameras)

Confusion matrix		Prediction														
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	Accuracy(%)
Truth	0	**2740**	0	4	5	0	2	3	0	0	0	9	0	0	0	99.17
	1	0	**3539**	0	0	0	0	1	0	0	0	0	0	0	0	99.97
	2	0	0	**4312**	0	0	0	17	0	0	0	0	0	0	0	99.61
	3	0	0	0	**1961**	0	0	3	0	0	0	0	0	0	0	99.85
	4	0	1	0	0	**3494**	0	1	0	0	1	0	0	0	0	99.91
	5	0	0	3	0	1	**4319**	40	3	0	0	0	2	0	0	98.88
	6	0	0	1	0	0	0	**4071**	0	0	0	0	0	1	0	99.95
	7	0	0	2	0	0	0	5	**4231**	0	0	0	1	0	0	99.81
	8	0	0	0	0	0	0	0	0	**4363**	0	0	1	0	4	99.89
	9	0	0	0	0	0	1	4	1	0	**1947**	0	7	0	0	99.34
	10	1	0	0	1	0	0	7	2	0	1	**2728**	1	0	0	99.53
	11	0	0	0	0	0	0	8	22	0	0	0	**4232**	0	0	99.30
	12	0	0	0	0	0	0	3	0	0	0	0	0	**3581**	0	99.92
	13	0	0	0	0	0	0	3	0	1	0	0	0	0	**4228**	99.91

Ⅴ. 결론 및 향후 과제

영상처리 기술의 발달은 콘텐츠 생산 및 소비 측면에서는 긍정적인 효과를 가져오고 있다. 하지만 영상 관련 범죄 또한 고도화되고 있는 것이 사실이다. 이에 디지털 포렌식 기술 중 하나로 영상을 이용한 범죄를 수사하기 위해 쓰이는 카메라 모델 판별 분야가 딥러닝의 발전에 따라 많은 변화를 이루고 있다. 딥러닝의 접목이 성능 향상에 직접적인 도움이 되고 있기에 새로운 모델이 발표될 때마다 빠르게 적용되는 모습을 보인다.

기본적인 CNN 구조를 조정하며 진행되었던 연구들은 ResNet, RemNet, DenseNet 등이 발표되면서 최신 기술을 응용하며 성능이 개선되어왔다. 새로운 모델이 발표될 때마다 높은 성능을 얻었으나 대부분의 연구는 발표된 모델을 그대로 사용하는 방식을 택했고 그에 따라 점점 많은 연산량과 학습 시간이 필요로 했다.

본 논문에서 제안하는 모델은 초기 정보를 유지한다는 DenseNet의 개념을 사용했다. 그에 따라 증가한 연산량을 DenseNet이 사용한 BottleNeck Layer와 압축을 사용하지 않고 Layer 반복 횟수 조절, 영상의 전처리 등의 방법을 통해 해결함으로 연산량과 성능의 Trade-Off를 막았다. 또한 많은 연산량을 처리하면서도 loss값이 빠르게 수렴하도록 조정하였다. Layer 반복 횟수 조절은 카메라나 프린터 기기 판별 같은 디지털 포렌식 분야의 영상 분류는 너무 깊은 모델의 성능이 오히려 떨어지는 모습을 보인다는 기존의 연구결과와도 일치한다. 이와 같은 조정의 결과, 14개의 카메라에 대해 99.65%의 높은 정확도를 얻음을 확인할 수 있었다.

차후 연구는 본 논문에서 제안한 모델을 활용하여 확장하는 방향으로 진행할 예정으로 실험에 사용한 카메라 모델의 수를 확장할 필요성이 있다. 또한 기존 연구들뿐만 아니라 현재 모델이 변형에 강인성이 취약하여 정상적인 영상이 아닌 회전, 왜곡, 재압축 등에 대해 판별률이 낮다. 이를 극복하기 위하여 본 논문에서 사용한 HPF 외에 LBP나 LQP와 같이 영상의 변형에 강인성을 갖는 특징을 추출하기 위한 다양한 필터를 탐색할 필요성이 있다.

References

[1] H. Y. Lee, "Imaging Device Identification using Sensor Pattern Noise Based on Wiener Filtering", The Transactions of the Korean Institute of Electrical Engineers, Vol. 65, No. 12, pp. 2153-2158, Sep. 2016.

[2] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks", Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, Honolulu, HI, USA, pp. 4700-4708, Jul. 2017.

[3] Dresden Image Database, http://forensics.inf.tu-dresden.de/ddimgdb/ [accessed: Jun. 26, 2019]

[4] S. Bayram, H. T. Sencar, and N. Memon, "Source camera identification based on CFA interpolation", Proc. of IEEE Int. Conf. on Image Processing, Genova, Italy, Vol. 3, pp. 69-72, Sep. 2005.

[5] K. S. Choi, E. Y. Lam, and K. K. Y. Wong, "Automatic source camera identification using the intrinsic lens radial distortion", Optics Express, Vol. 14, No. 24, pp. 11551-11565, Nov. 2006.

[6] Y. Tomioka and H. Kitazawa, "Digital camera identification based on the clustered pattern noise of image sensors", Proc. of IEEE Int. Conf. on Multimedia and Expo, Barcelona, Spain, pp. 1-4, Jul. 2011.

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

[8] A. Jeyalakshmi and D. R. Chitra, "Source Camera Identification using Image Features", International Journal of Applied Engineering Research, Vol. 13, No. 1, pp. 490-504, Jan. 2018.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks", Advances in neural information processing systems, Vol. 25, No. 2, pp. 1097-1105, Jan. 2012.

[10] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, and A. Rabinovich, "Going deeper with convolutions", Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, Boston, MA, USA, pp. 1-9, Jun. 2015.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition", Proc. of IEEE Conf. on Computer Vision and Pattern Recognition, Las Vegas, USA, pp. 770-778, Jun. 2016.

[12] L. Baroffio, L. Bondi, P. Bestagini, and S. Tubaro, "Camera identification with deep convolutional networks", arXiv preprint arXiv:1603. 01068, Mar. 2016.

[13] A. Tuama, F. Comby, and M. Chaumont, "Camera model identification with the use of deep convolutional neural networks", Proc. of IEEE Int. Workshop on Information Forensics and Security, Abu Dhabi, United Arab Emirates, pp. 1-6, Dec. 2016.

[14] A. Kuzin, A. Fattakhov, I. Kibardin, V. I. Iglovikov, and R. Dautov, "Camera Model Identification Using Convolutional Neural Networks", Proc. of IEEE Int. Conf. on Big Data, Seattle, USA, pp. 3107-3110, Dec. 2018.

[15] A. M. Rafi, T. I. Tonmoy, R. Hoque, and M. Hasan, "RemNet: Remnant Convolutional Neural Network for Camera Model Identification", arXiv preprint arXiv:1902.00694, Feb. 2019.

[16] U. Kamal, A. M. Rafi, R. Hoque, S. Das, A. Abrar, and M. Hasan, "Application of DenseNet in Camera Model Identification and Post-processing Detection", arXiv preprint arXiv:1809.00576, Sep. 2018.

[17] H. Yao, T. Qiao, M. Xu, and N. Zheng, "Robust multi-classifier for camera model identification based on convolution neural network", IEEE Access, Vol. 6, pp. 24973-24982, May 2018.

[18] D. Freire-Obregón, F. Narducci, S. Barra, and M. rillón-Santana, "Deep learning for source camera identification on mobile devices", Pattern Recognition Letters, Available online 9, Jan. 2018.

[19] M. Darvish Morshedi Hosseini, and M. Goljan, "Camera Identification from HDR Images", Proc. of ACM Workshop on Information Hiding and Multimedia Security, pp. 69-76, Jul. 2019.

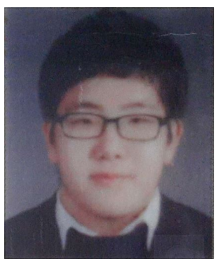
[20] C. Meij and Z. Geradts, "Source camera identification using Photo Response Non-Uniformity on WhatsApp", Digital Investigation, Vol. 24, pp. 142-154, Mar. 2018.

저자소개

이 수 현 (Soo-Hyeon Lee)

2018년 2월 : 금오공과대학교 컴퓨터소프트웨어공학과 (학사)
2018년 2월 ~ 현재 : 금오공과대학교 소프트웨어공학과 석사과정
관심분야 : Image Processing, Deep Learning

김 동 현 (Dong-Hyun Kim)



2016년 2월 : 금오공과대학교 컴퓨터소프트웨어공학과 (학사)
2016년 2월 ~ 현재 : 금오공과대학교 소프트웨어공학과 석사과정
관심분야 : 이미지 처리, 포렌식, 딥 러닝

이 해 연 (Hae-Yeoun Lee)

1997년 : 성균관대학교 정보공학과 (학사)
1999년 : 한국과학기술원 전산학과 (공학석사)
2006년 : 한국과학기술원 전자전산학과 (공학박사)
2008년 ~ 현재 : 금오공과대학교 컴퓨터소프트웨어공학과 교수
관심분야 : Digital Forensics, Image Processing, IoT