Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.33, NO.4, Aug. 2023 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2023.33.4.671

이미지 조작 탐지를 위한 포렌식 방법론*

이 지 원,[1†] 전 승 제,[1] 박 윤 지,[1] 정 재 현,[1] 정 두 원[2‡]
[1,2]동국대학교 (대학원생, 교수)

A Forensic Methodology for Detecting Image Manipulations*

Jiwon Lee,[1†] Seungjae Jeon,[1] Yunji Park,[1] Jaehyun Chung,[1] Doowon Jeong[2‡]
[1,2]Dongguk University (Graduate student, Professor)

요 약

인공지능이 이미지 편집 기술에 적용되어 조작 흔적이 거의 없는 고품질 이미지를 생성할 수 있게 되었다. 그러나 이러한 기술들은 거짓 정보 유포, 증거 인멸, 사실 부인 등의 범죄 행위에 악용될 수 있기 때문에 이에 대응하기 위한 방안이 필요하다. 본 연구에서는 이미지 조작을 탐지하기 위해 이미지 파일 분석과 모바일 포렌식 아티팩트 분석을 수행한다. 이미지 파일 분석은 조작된 이미지의 메타데이터를 파싱하여 Reference DB와 비교분석을 통해 조작 여부를 탐지하는 방법이다. Reference DB는 이미지의 메타데이터에 남는 조작 관련 아티팩트를 수집하는 데이터베이스로서, 이미지 조작을 탐지하는 기준이 된다. 모바일 포렌식 아티팩트 분석은 이미지 편집 도구와 관련된 패키지를 추출하고 분석하여 이미지 조작을 탐지하도록 한다. 본 연구에서 제안하는 방법론은 기존의 그래픽적 특징 기반 분석의 한계를 보완하고, 이미지 처리 기법과 조합하여 오탐을 줄일 수 있도록 한다. 연구 결과는 이러한 방법론이 디지털 포렌식 조사 및 분석에 유의미하게 활용될 수 있음을 보여준다. 또한, 조작된 이미지 데이터셋과 함께 이미지 메타데이터 파싱 코드와 Reference DB를 제공하여 관련 연구에 기여하고자 한다.

ABSTRACT

By applying artificial intelligence to image editing technology, it has become possible to generate high-quality images with minimal traces of manipulation. However, since these technologies can be misused for criminal activities such as dissemination of false information, destruction of evidence, and denial of facts, it is crucial to implement strong countermeasures. In this study, image file and mobile forensic artifacts analysis were conducted for detecting image manipulation. Image file analysis involves parsing the metadata of manipulated images and comparing them with a Reference DB to detect manipulation. The Reference DB is a database that collects manipulation-related traces left in image metadata, which serves as a criterion for detecting image manipulation. In the mobile forensic artifacts analysis, packages related to image editing tools were extracted and analyzed to aid the detection of image manipulation. The proposed methodology overcomes the limitations of existing graphic feature-based analysis and combines with image processing techniques, providing the advantage of reducing false positives. The research results demonstrate the significant role of such methodology in digital forensic investigation and analysis. Additionally, We provide the code for parsing image metadata and the Reference DB along with the dataset of manipulated images, aiming to contribute to related research.

Keywords: Image manipulation detection, Manipulated image dataset, Mobile forensics, Forensic Methodology

Received(07. 05. 2023), Modified(07. 21. 2023), Accepted(07. 23. 2023)

* 본 논문은 2023년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No.2022-0-00281, 인공지능 기술 활용 디지털증거 분석 기법 개발)

† 주저자, jiwon2750@dgu.ac.kr
‡ 교신저자, doowon@dgu.ac.kr(Corresponding author)

I. 서 론

디지털 기술의 발전으로 이미지 편집 도구가 보급되면서 누구나 쉽게 이미지를 편집할 수 있게 되었다. 편집에 쓰이는 대표적인 애플리케이션에는 Adobe Photoshop, Snapseed, Meitu 등이 있으며, 앱 스토어에서 서드파티 앱을 다운받지 않고도 스마트폰 기본 애플리케이션을 통해 이미지 편집 기능이 제공된다.

최근에는 특정 객체나 배경을 지우는 등과 같이 인공지능을 활용하여 이미지를 편집할 수 있도록 하는 기술이 등장하였다. 이미지에서 객체를 지우는 기술은 일반적으로 object removal 또는 inpainting이라는 기술로 알려져 있으며, 주로 딥러닝 알고리즘을 사용하여 작동된다[1]. 이는 이미지에서 지울 객체 위치의 픽셀값을 확인한 후, 지워질 영역 주변의 픽셀값을 이용해 기존 값을 대체하는 방식으로 이루어진다. 이때, 대체하는 과정에서는 인접한 픽셀값들과의 관계를 고려해 자연스럽게 이미지를 만들도록 한다. 배경을 지우는 기술도 이와 유사한 원리로 객체와 배경을 구분하는 작업을 수행한 후, 배경으로 인식된 부분을 제거하는 방식으로 이루어진다.

과거 전문가의 조작을 필요로 했던 기술들을 현재는 이미지 편집 도구를 통해 누구나 손쉽게 사용할 수 있게 됨으로써, Fig. 1과 Fig. 2.와 같이 사진의 선명도, 색상 표현 등과 같은 요소들도 편집에 의한 영향이 눈에 띄지 않을 만큼 높은 퀄리티를 보여주는 것을 확인할 수 있다.

그러나 이러한 기술은 거짓 정보 생성 및 배포, 증거 인멸 및 사실 부인 등과 같은 범죄 행위에 악용될 수 있다. 이러한 문제를 해결하기 위해 이미지 포렌식 분야에서는 이미지 조작 탐지(image manipulation detection) 기술을 활발히 연구하고 있다[3]. 그러나 이미지 자체의 그래픽적 분석에 초점을 맞추고 있어 디지털 포렌식 측면에서 종합적인 이미지 조작에 대한 탐지 방법론을 제시하지 못한다. 따라서 본 논문에서는 이미지 조작 탐지를 위한 이미지 파일 분석과 더불어 모바일 포렌식 아티팩트를 분석함으로써 관련 범죄 수사에 적극적으로 대응할 수 있도록 한다.

본 논문은 다음과 같이 구성된다. 2장에서는 이미지 조작 기술과 탐지 방안에 대한 선행 연구를 분석한다. 3장에서는 조작된 이미지 데이터셋 생성 방법과 이미지 파일 분석을 통해 조작을 탐지할 수 있는 방안에 대해 설명하며, 4장에서는 모바일 포렌식 아티팩트를 분석함으로써 조작을 탐지할 수 있는 방법을 제시한다. 5장에서는 본 연구에서 제시하는 이미지 조작 탐지를 위한 방법론을 제시하며 이에 대한 논의를 진행한다. 6장에서는 본 논문의 요점과 후속 연구의 방향성에 대해 논한다.

Fig. 1. Original image (Left) and edited image with the person object removed (Right)

Fig. 2. Original image (Left) and edited image where the original background has been removed and replaced with another one (Right)

II. 관련 선행 연구

2.1 선행 연구

이미지 조작(image manipulation)은 디지털 기기에서 이미지 편집 도구로 디지털 이미지에 수행하는 모든 작업을 말한다. Zheng et al.[2]와 Thakur and Rohilla[3]는 Fig. 3.과 같이 이미지 조작을 이미지 위조(image forgery), 이미지 변조(image tampering), 이미지 생성(image generatin

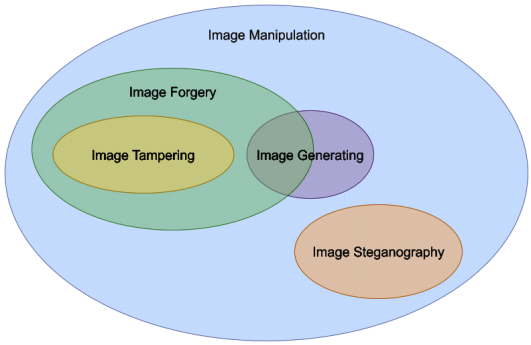


Fig. 3. Image manipulation category

g), 이미지 스테가노그라피(image steganography)와 같은 하위 범주로 분류하였다. 이미지 위조는 과거에 일어난 사실을 속이기 위해 가짜 그래픽 콘텐츠를 생성하는 이미지 조작을 의미하며, 이미지 변조는 이미지 그래픽의 특정 부분을 변경하는 특수한 유형의 이미지 위조를 의미한다. 이미지 생성은 컴퓨터를 통해 위조된 이미지를 생성하는 것을 의미하며, 이미지 스테가노그라피는 그래픽적 특성을 이용해 눈속임을 의도하는 것이 아닌, 이미지의 특정 픽셀을 특정 값으로 변경하거나 채워 넣어 이미지에 추가 정보를 포함하는 것을 의미한다.

이미지 조작 기술에는 대표적으로 Copy-move와 Cut-paste를 포괄하는 용어로 사용되는 Splicing과 Erase-fill이라는 용어로 사용되는 Inpainting 기술이 포함된다[2]. Copy-move는 이미지 내의 특정 객체 혹은 영역을 복사하여 동일한 이미지의 다른 위치에 붙여넣는 기술이며, Cut-paste는 잘라낸 영역을 다른 이미지에 붙여넣는 기술이다. Erase-fill은 이미지 내의 특정 객체나 영역을 지우고 그 부분을 주변 환경에 맞게 채워 넣음으로써 이미지를 조작하는 기술이다.

이러한 기술로 조작된 이미지를 탐지하기 위해 Noise[4], ELA(Error Level Analysis)[5], PCA(Principal Component Analysis)[6], Luminance Gradient[7]와 같은 이미지 피쳐 기반의 연구가 꾸준히 진행되고 있다. 최근에는 딥러닝 기반으로 이미지 조작을 탐지하는 연구가 진행되고 있다[8]. Deep fake 기술 악용 사례가 증가하면서 조작된 얼굴 이미지를 탐지하는 연구들이 진행되고 있으며[9], Rossler et al.[10]은 표정을 바꾸거나 얼굴을 교체하는 Face Expression과 Face Swap을 딥러닝에 기반하여 탐지하는 연구를 발표하였다. Bappy et al.[11]는 하이브리드 CNN-LSTM 모델을 기반으로 조작된 영역과 조작되지 않은 영역을 구분하는 기법을 제시하였다.

이미지의 그래픽적 특징이 아닌 이미지 파일 포맷이 지닌 특징에 주목한 연구도 존재한다. 특히, 스마트폰에서 많이 사용되는 JPEG 파일 포맷의 이미지 생성방식을 통해 촬영 기기, 애플리케이션을 탐지하는 방법 또한 연구되고 있다. 김민식 등[12]은 JPEG 이미지 생성 과정에서 이미지 자체와 이미지 썸네일에 생성되는 DQT(Define Quantization Table)를 종합 분석하여 이미지를 촬영한 단말기, 저장에 사용한 애플리케이션 등을 식별할 수 있다는 것을 확인하였다. 허욱 등[13]은 JPEG 압축 알고리즘을 분석하여 이미지를 유포한 기기에 따른 MMS 및 메신저 이미지의 처리 과정을 분석하여 이미지 유포에 사용된 애플리케이션 및 기기 제조사를 특정하는 연구를 진행하였다. 선행 연구는 이미지 분석에 초점을 두어 주로 메타데이터나 압축 알고리즘을 활용하여 이미지 출처를 식별하는 데 주목한다. 그러나 이는 이미지 파일에 기록되는 정보 자체에만 초점을 두기 때문에 이미지 조작을 탐지하기 위한 포괄적인 방법론을 제시하지 못한다는 한계가 있다.

본 논문에서는 알고리즘 개발 및 구동에 필요한 리소스가 딥러닝 기반 방법론에 비해 상대적으로 적은 방식을 우선으로 고려하며, 디지털 포렌식의 특수성을 고려한 포괄적인 이미지 조작 탐지의 포렌식 방법론을 제시한다.

2.2 연구 범위 및 필요성

디지털 포렌식에서의 주요 분석 대상인 저장매체에는 수많은 조작된 이미지가 존재한다. 이미지 조작에서 이미지 위조를 식별하기 위해서는 해당 행위를 탐지하여 선별하는 작업이 필요하다. 그러나 이미지에 가하는 모든 행위에 대하여 이미지 위조라 볼 수 없기 때문에, 본 연구에서는 이미지 위조에 비해 보다 넓은 범위인 이미지 조작 탐지를 연구 범위로 설정한다.

이미지 조작을 식별하기 위한 연구는 대부분 원본 이미지가 주어졌을 때 이미지 조작을 탐지하는 방법에 초점을 맞추고 있으며, 이미지의 그래픽적 특성을 분석하여 조작 여부를 탐지하도록 한다. 또한 메타데이터의 분석을 통해 이미지의 마지막 출처를 식별하도록 하지만 이는 촬영 및 유포 시 발생한 정보를 통

해 기기를 식별하는 것에 주목한다.

이에 본 연구에서는 이미지 파일 분석을 통해 조작 시 발생한 정보를 식별하여 조작 여부 및 조작에 사용된 애플리케이션을 식별하는 것에 집중하며, 이미지 메타데이터를 고려하는 과정에서 이미지 조작 탐지의 기준이 되는 Reference DB를 구축하도록 한다. 이는 분석 대상 이미지의 메타데이터와 DB에 삽입된 조작시 나타나게 되는 메타데이터셋들과의 비교 분석을 용이하게 하여 조작 탐지를 더욱 효과적으로 수행할 수 있게 한다. 따라서 조작된 이미지 그 자체로의 분석을 통해 조작을 식별할 수 있도록 하며, 원본 이미지가 주어지지 않거나, 비이상적인 그래픽적 탐지가 불가능하고, 탐지 알고리즘을 모르는 경우에도 이미지 조작 여부를 쉽게 판단 할 수 있도록 한다.

또한 디지털 포렌식에서는 이미지 조작이 발생한 다양한 시나리오나 상황이 가정된다. 예를 들어, 법원에서 사건 증거물로 사용되는 디지털카메라로 촬영한 사진이나 영상에 대한 조작 여부가 의심될 때, 모바일 포렌식 아티팩트를 분석하여 이미지가 언제, 어떻게 조작되었는지 확인해야 한다. 또한, 온라인상에서 유포되는 불법적인 성인물에 대해서도 이미지뿐만 아니라 유포자의 스마트폰을 압수하여 조작 행위를 식별해야 할 필요가 있다. 이와 같은 상황에서는 이미지 분석만으로는 충분한 정보를 제공하기 어렵기 때문에 이미지 위조자가 어떠한 이미지를 사용하여 어떠한 조작을 가했는지 등과 관련한 행위의 구체적인 정보에 대한 파악이 필요하다.

이에 본 연구에서는 이미지 편집 도구와 관련한 모바일 포렌식 아티팩트를 분석함으로써 이미지 조작 행위를 구체적으로 파악할 수 있도록 한다.

따라서 이미지 파일 분석과 더불어 모바일 포렌식 아티팩트 분석을 시행함으로써 이미지 조작 탐지를 보다 정확하게 수행할 수 있도록 한다.

III. 이미지 조작 탐지를 위한 이미지 파일 분석

본 장에서는 조작된 이미지 데이터셋을 생성하고 이미지 파일의 메타데이터를 분석하여 조작 관련 흔적을 파악한다. 또한 분석 결과를 이용하여 이미지 조작 탐지를 위한 기준이 되는 Reference DB를 생성한다.

3.1 조작된 이미지 데이터셋

본 연구에서는 이미지 파일 분석을 위해 11개의 이미지 편집 도구를 사용하여 조작된 이미지 데이터셋 생성을 생성하였다. 도구는 구글 플레이 스토어에

Table 1. Image editing tools information

APP	Corp.	Version	Package	Function
Snapseed	Google LLC	2.19.1.3030 51424	com.niksoftware.snapseed	Object Removal
Meitu	Meitu(China) Limited	9.7.5.5	com.mt.mtxx.mtxx	Object Removal
Remove Unwanted Object	BG.Studio	1.3.8	vn.remove.photo.content	Object Removal
SnapEdit	SnapEdit Team	3.4.1	snapedit.app.remove	Object Removal
Adobe Photoshop Fix	Adobe	1.1.0	com.adobe.adobephotoshopfix	Object Removal
Photoshop Express	Adobe	8.8.17	com.adobe.psmobile	Object Removal
Samsung Photo Editor	Samsung	3.0.25.33	com.sec.android.mimage. photoretouching	Object Removal
removebg	Kaleido AI	1.1.4	bg.remove.android	Background Removal
Backgroun d Eraser (Inshot)	InShot	2.122.33	photoeditor.cutout.backgrounderaser	Background Removal
Background Eraser (handy)	handy Closet	4.1.0	com.handycloset.android.eraser	Background Removal
Photo Studio	KVADGroup App Studio	2.6.2.1178	com.kvadgroup.photostudio	Background Removal

Table 2. Post-processing functions and save methods considered for generating manipulated images

APP	Setting Values	Save Methods	Samples
Snapseed	Image Size (800, 1366, 1920, 2000, 4000, No Resizing) Format & Quality (JPG 100%, 95%, 80%, PNG)	Save, Export, Export to another folder	539
Meitu	Image Quality (UHD, Standard)	Save, Quick Save	44
Remove Unwanted Object	-	Save	11
SnapEdit	-	High, Standard	22
Adobe Photoshop Fix	-	Save to Gallery	11
Photoshop Express	Image Size (600x800, 1125x1500, 1500x2000, 300x4000,No Resizing, Square)	Save	66
Samsung Photo Editor	Image Size (20%, 40%, 60%, 80%, No Resizing)	Save, Save as another file	11
removebg	-	Download (preview image)	110
Backgroun d Eraser (Inshot)	Image Size (1080, 1920)	Save	22
Background Eraser (handy)	Smooth Edge (0, 1, 2, 3, 4, 5)	Save	66
Photo Studio	Image Size (Normal, Small, No Resizing)	JPG, PNG	66

등록된 평점 4.0 이상, 100만회 이상 다운로드 된 애플리케이션 10종과 삼성 갤럭시에서 지원하는 포토 에디터 1종으로 구성되며, Table 1.은 사용한 이미지 편집 도구의 정보를 보여준다. 11개 애플리케이션 중, 7개는 Object Removal 기능을 사용하였으며, 나머지 4개 애플리케이션을 통해서는 Background Removal 기능을 사용하여 조작된 이미지를 생성하였다.

이미지 조작 시, 이미지 설정값에 따라 남는 메타데이터와 이미지 표현을 위한 구성 값들이 달라지기 때문에 이미지 편집 도구가 이미지 저장 과정에서 제공해주는 후처리 기능과 저장 방식을 고려하였다. Table 2.는 각 이미지 편집 도구에서 고려한 설정값과 생성한 이미지 샘플의 총 개수를 보여준다. 조작된 이미지는 11개의 원본 이미지를 Galaxy S10e(Android12)로 촬영한 후, 이미지 편집 도구별 개별적으로 제공하는 이미지 후처리 과정과 저장 방식을 모두 고려하여 생성하였다. Fig. 4.는 조작된 이미지 생성에 사용되는 11개의 원본 이미지에서 지운 대상이 되

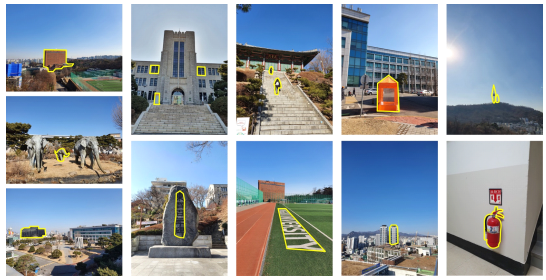


Fig. 4. Segmentation of objects

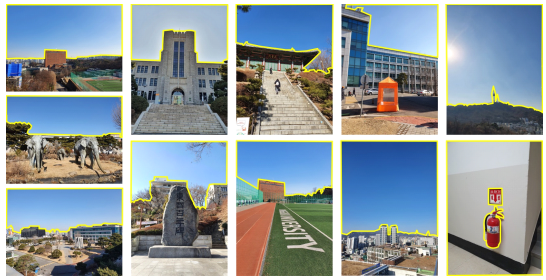


Fig. 5. Segmentation of backgrounds

는 객체를 표시한 것이며, Fig. 5.는 지운 대상이 되는 배경을 표시한 것이다.

본 연구에서 생성한 데이터셋 및 각 이미지의 저장 방법에 대한 정보가 포함된 파일은 GitHub repository[1] 에서 다운로드 가능하다.

3.2 이미지 파일 분석

이미지 조작을 탐지하기 위한 이미지 메타데이터를 식별하며, 생성한 데이터셋을 활용하여 조작의 흔적을 분석한다. 본 연구에서는 디지털 이미지 처리에 널리 사용되는 표준 파일 포맷인 JPEG(Joint Photographic Experts Group)을 대상으로 연구를 수행한다.

3.2.1 Exif (Exchangeable Image File Format)

Exif(Exchangeable Image File Format)는 JPEG 파일 내에 포함된 추가적인 메타데이터 정보를 나타낸다. 이 정보는 디지털 카메라나 스마트폰 등 디지털 기기로 촬영된 사진에 대한 세부 정보를 담고 있다. 이미지 편집 도구로 편집된 이미지는 각 애플리케이션의 시그니처를 Exif에 남기게 되며 이를 통해 이미지 조작 여부를 식별할 수 있다. Table 3. 은 Exif에 시그니처를 남기는 이미지 편집 도구를 보여준다. 본 연구에서 다룬 11개의 이미지 편집 도구 중, 4개의 애플리케이션에서 관련 시그니처를 담는 것을 확인할 수 있다. 이와 같이 이미지의 마지막 출처 또는 편집 정보를 확인할 수 있지만, 사람이 쉽게 조작할 수 있다는 단점이 존재한다.

Table 3. Exif containing signatures

APP	Exif
Snapseed	Software : Snapseed 2.0
Meitu	Artist : Meitu Software : Meitu 9755
Remove Unwanted Object	Software : AdvaSoft TouchRetouch
Photoshop Express	Software : Adobe Photoshop Express (Android)

1) https://github.com/allinonee/Manipulated-Image-Dataset

3.2.2 DQT (Define Quantization Table)

DQT(Define Quantization Table)는 JPEG 파일 포맷에서 이미지의 압축률을 결정하는 양자화 테이블이다. 이 테이블은 8x8 크기의 정수 배열로 표현되며, DCT(Discrete Cosine Transform)를 수행한 결과로 얻은 픽셀 블록의 계수를 양자화하는 데 사용된다. DQT는 일반적으로 밝기와 색상에 대한 2개 테이블로 구성되며, 이는 카메라 모델 및 제조사에서 파생될 수 있는 고유한 디지털 식별자로 활용될 수 있다[14,15]. 이러한 특성은 이미지 편집 도구를 식별하는데에도 적용될 수 있다. 이미지 편집 도구는 이미지 퀄리티를 결정하기 위해 특정 DQT를 사용하므로, 이미지의 마지막 출처 및 조작 여부를 식별하는 기준점으로 활용될 수 있다. Table 4.는 조작된 이미지 데이터셋의 분석 결과로, 이미지 편집 도구에

Table 4. Representative DQT used by image editing tools

APP	DQT							
Snapseed	**Luminance (ID: 0, Length: 67)**							
	DQT, Row #0	1	1	1	1	1	1	1
	DQT, Row #1	1	1	1	1	1	1	1
Meitu	DQT, Row #2	1	1	1	1	1	1	1
	DQT, Row #3	1	1	1	1	1	1	1
	DQT, Row #4	1	1	1	1	1	1	1
	DQT, Row #5	1	1	1	1	1	1	1
Adobe Photoshop Fix	DQT, Row #6	1	1	1	1	1	1	1
	DQT, Row #7	1	1	1	1	1	1	1
	Chrominance (ID: 1, Length: 67)							
Background Eraser (Inshot)	DQT, Row #0	1	1	1	1	1	1	1
	DQT, Row #1	1	1	1	1	1	1	1
	DQT, Row #2	1	1	1	1	1	1	1
	DQT, Row #3	1	1	1	1	1	1	1
	DQT, Row #4	1	1	1	1	1	1	1
Photo Studio	DQT, Row #5	1	1	1	1	1	1	1
	DQT, Row #6	1	1	1	1	1	1	1
	DQT, Row #7	1	1	1	1	1	1	1
Remove Unwanted Object	**Luminance (ID: 0, Length: 67)**							
	DQT, Row #0	3	2	2	3	5	8	10	12
	DQT, Row #1	2	2	3	4	5	12	12	11
	DQT, Row #2	3	3	3	5	8	11	14	11
	DQT, Row #3	3	3	4	6	10	17	16	12
	DQT, Row #4	4	4	7	11	14	22	21	15
SnapEdit	DQT, Row #5	5	7	11	13	16	21	23	18
	DQT, Row #6	10	13	16	17	21	24	24	20
	DQT, Row #7	14	18	19	20	22	20	21	20
	Chrominance (ID: 1, Length: 67)							
	DQT, Row #0	3	4	5	9	20	20	20	20
	DQT, Row #1	4	4	5	13	20	20	20	20
Photo Studio	DQT, Row #2	5	5	11	20	20	20	20	20
	DQT, Row #3	9	13	20	20	20	20	20	20
	DQT, Row #4	20	20	20	20	20	20	20	20
	DQT, Row #5	20	20	20	20	20	20	20	20
	DQT, Row #6	20	20	20	20	20	20	20	20
	DQT, Row #7	20	20	20	20	20	20	20	20

서 제공해주는 대표적인 DQT를 보여준다. 서로 동일한 DQT를 사용하거나 자체적인 DQT를 사용하는 것으로 나타났으며, 이러한 패턴은 이미지 조작을 감지하는 데에 유용하게 활용될 수 있다. 예를 들어, 특정 DQT 패턴이 다른 이미지에서도 나타난다면 해당 이미지가 조작되었을 가능성을 높일 수 있으며, 이는 이미지 조작 탐지 기술의 정확도를 향상시킬 수 있다.

3.2.3 Filename signature

본 연구에서는 이미지 조작 여부를 식별하는 것뿐만 아니라 사용된 이미지 편집 도구를 확인할 수 있도록, 편집된 이미지의 파일명을 분석한다. Table 5.는 파일명 분석 결과이다. 이를 통해 편집된 이미지의 파일명에는 대부분 파일의 생성일자을 포함하고 있으며, 편집에 사용된 원본 이미지의 파일명, 이미지 편집 도구의 시그니처 등을 포함하는 것을 확인할 수 있다. 파일명은 사용자가 상대적으로 쉽게 변경할 수 있는 정보이기 때문에, 단독으로 포렌식 분석을 수행하기에는 한계가 있다고 할 수 있다. 그러나 파일명이 보존되고 있는 경우에는 편집 애플리케이션이 생성하는 특정한 파일명이 단서로 작용할 수 있다는 점도 고려해야 한다.

3.3 Reference Database

앞서 살펴본 이미지 파일의 메타데이터를 종합적으로 고려하여 이미지 조작 및 이미지 마지막 출처를 식별할 수 있도록 한다. 본 연구에서 사용한 DQT Parser는 조작된 이미지의 Exif와 DQT를 파싱하며,

Table 5. Edited images filename analysis

APP	Save	Edited image filename info	Signature
Snapseed	Save	(Original_image_filename)-(Number).jpeg	-
	Export to another folder	(Original_image_filename)_edited.jpeg or png	edited
Meitu	Save	MTXX_MH(Editing_image_creation_datetime).jpg	MTXX, MH
	Quick Save	MTXX_formula(Editing_image_creation_datetime).jpg	formula
Remove Unwanted Object	Save	WipeOut(Editing_image_creation_minute)_(Editing_image_creation_day)_(Editing_image_creation_year)_(Editing_image_creation_time).jpg	WipeOut
SnapEdit	Save	(Editing_image_creation_datetime).png	-
Adobe Photoshop Fix	Save	PSFix_(Editing_image_creation_date)_(Editing_image_creation_time).jpeg	PSFix
Photoshop Express	Save	PSX_(Editing_image_creation_date)_(Editing_image_creation_time).jpg	PSX
Samsung Photo Editor	Save	(Original_image_creation_date)_(Original_image_creation_time).jpg	-
	Save as	(Editing_image_creation_date)_(Editing_image_creation	-
removebg	Download	ei_(Editing_image_creation_datetime)-removebg-preview.png	removebg
Background Eraser (Inshot)	Save	BackgroundEraser_(Editing_image_creation_date)_(Editing_image_creation_time).jpg or png	BackgroundEraser
Background Eraser (handy)	Save	(Editing_image_creation_datetime).png	-
Photo Studio	Save	photostudio_(Editing_image_creation_datetime).jpg or png	photostudio

Reference DB에 데이터를 삽입한다. 추가적으로 이미지 파일명에 이미지 편집 도구 시그니처를 남기는 경우를 고려하여 직접 삽입할 수 있도록 한다. 본 연구에서는 초기 Reference DB에 조작 관련 데이터를 삽입하기 위해 앞서 생성한 데이터셋을 활용한다. 이렇게 생성된 Reference DB는 이미지 조작 및 이미지 마지막 출처 식별 기준이 된다. 그러나 소프트웨어의 업데이트에 따라 메타데이터가 변경될 수 있기 때문에 본 연구에서는 조작된 이미지를 정확히 탐지할 수 있도록 정기적인 DB 업데이트를 수행한다.

Fig. 6.은 Reference DB에 대한 스키마를 보여준다. Image_Editors 테이블은 이미지 편집 도구의 이름과 해당 버전을 저장하는 참조 테이블 역할을 한다. Parsed_Exif_DQT 테이블은 이미지 편집 도구별 Exif와 DQT 정보를 저장한다. Exif의 경우, exiftool을 활용하여 이미지 편집 도구의 시그니처를 남기는 정보를 파싱하여 삽입하였다. DQT의 경우 JPEG 이미지에서의 DQT 헤더 시그니처를 통해 파싱한 후 MD5 해시를 적용하여 삽입하도록 하였다. Editor_Signature 테이블은 이미지 파일명에 이미지 편집 도구의 시그니처가 남는 경우 해당 정보를 삽입하도록 하였다.

본 연구에서 활용 및 생성한 DQT Parser와 Reference DB에 대한 정보는 GitHub repository[2)]에서 확인할 수 있다.

Fig. 6. Schema for the Reference DB

IV. 이미지 조작 탐지를 위한 모바일 포렌식 아티팩트 분석

본 장에서는 모바일 포렌식 아티팩트 분석을 통해 이미지 조작을 탐지하는 방법을 살펴본다. 이를 위해 이미지 조작과 관련하여 어떠한 정보를 획득 및 고려할 수 있는지 확인한다.

4.1 이미지 조작 행위 식별 정보

이미지 조작 행위 식별을 위한 정보는 다음과 같다. 먼저 '편집된 이미지 존재 여부'이다. 편집된 이미지는 갤러리에 저장된 이미지가 아닌, 이미지 편집 도구의 앱 데이터가 저장된 경로에 존재하는 이미지 편집 도구 자체가 남기는 아티팩트를 의미한다. 애플리케이션에서 이미지를 편집하면 자동으로 갤러리에 저장이 되지만, 이를 삭제할 경우 이미지는 확인할 수 없게 된다. 따라서 삭제 시에도 편집된 이미지가 남는다면 이는 증거 인멸 등의 안티포렌식 행위에 대응할 수 있는 유의미한 정보가 된다. 두 번째는 '편집된 영역 식별 가능 여부'이다. 디지털 이미지가 증거로 사용될 경우, 정확하게 어느 부분이 조작되었는지를 확인할 수 있다면 이는 수사에 매우 유용한 정보가 된다. 세 번째는 '편집에 사용된 원본 이미지 확인 가능 여부'이다. 이는 어떠한 이미지가 편집에 사용되었는지 알 수 있는 정보가 되며, 편집된 이미지와 원본 이미지의 유사성을 비교하여 편집된 이미지가 원본 이미지에서 파생되었는지 여부를 판단할 수 있도록 한다.

네 번째는 '편집 로그 존재 여부'이다. 해당 이미지에 대한 편집 시점과 방법, 편집된 결과물 등의 정보를 파악할 수 있는 중요한 정보가 된다. 특히 편집 로그는 이미지 조작 행위에 대한 확실한 정보를 주기 때문에 아티팩트 분석에서 매우 유용한 정보가 된다. 다음은 '이미지 캐싱 여부'이다. 대부분의 애플리케이션에서는 성능 향상, 서버 부하 줄임, 용량 문제 해소 등을 위해 캐시이미지를 저장한다. 이는 사용자가 안티포렌식 행위를 했을 경우, 유용하게 쓰일 수 있다. 이미지 편집 도구는 편집된 이미지를 갤러리에 저장하기 위해 첫 사용시, 갤러리 접근 권한 허용 여부를 묻는다. 이는 필수적으로 거치는 과정이므로 캐시에는 앱 자체에서 쓰이는 이미지뿐만 아니라, 갤러리에 저장되어 있는 이미지 또한 남게 된다. 갤러리에서 증거 인멸을 위해 이미지를 삭제하더라도, 앱에서 남기는 캐시 이미지를 통해 갤러리에 남아있던 이미지를 확인할 수 있게 된다. 본 연구에서는 추가적으로 '계정 정보, 설치시간, 앱 최근 사용 시간'을 확인함으로써 조작 행위 식별에 도움이 될 수 있도록 한다. 이미지 편집 도구 관련 아티팩트에 대한 종합적인 분석 결과는 Table 6.에서 확인 가능하다.

2) https://github.com/allinonee/DQT-Parser

Table 6. Information that can be confirmed by image editing tool (The triangle shape indicates that only the most recently edited record can be checked)

APP	Edited image	Manipulated region	Original image	Edited log	Image caching	Account info	Installation time	Recent usage time
Snapseed	△	-	-	-	-	-	-	-
Meitu	○	-	○	○	○	-	○	○
Remove Unwanted Object	-	-	-	-	○	-	○	○
SnapEdit	○	○	○	-	-	-	○	○
Adobe Photoshop Fix	-	-	○	○	○	○	○	○
Photoshop Express	-	-	-	-	○	○	○	○
Samsung Photo Editor	-	-	○	-	-	-	-	-
removebg	-	-	○	-	-	-	○	○
Background Eraser (Inshot)	△	-	△	○	○	-	○	○
Background Eraser (handy)	-	-	△	-	-	-	○	○
Photo Studio	○	○	○	○	○	-	○	○

4.2 모바일 포렌식 아티팩트 분석

본 장에서는 앞서 살펴본 고려 사항들에 대한 실제 분석 결과를 살펴본다. 이미지 조작 관련 상세 정보를 수집하기 위해 이미지 편집 도구의 앱 데이터가 저장되는 경로를 중심으로 분석을 진행한다. 안드로이드의 '/data/data/{package_name}/' 경로는 해당 패키지명을 가진 앱의 데이터가 저장되는 경로로 앱의 설정, 로그, 캐시, 데이터베이스 등이 저장된다. 본 연구에서는 3장에서 사용된 11개의 이미지 편집 도구를 대상으로 편집을 진행한 후, 위 경로를 추출하여 분석을 진행하였으며 이 외에도 타 경로에서 유의미한 데이터를 포함하는 경우, 추가적인 분석을 진행하였다.

4.2.1 편집된 이미지

11개 애플리케이션 중, 5개의 애플리케이션에서 편집된 이미지를 확인할 수 있었다. Snapseed의 경우, 앞선 경로에서 임의의 숫자로 되어있는 편집된 이미지 파일을 JPEG 시그니처 식별을 통해 확장자를 변경해줌으로써 확인할 수 있으며, Meitu와 Background Eraser(Inshot)의 경우 개인 외부 저장소(Private External Storage)인 '/storage/emulated/0/Android/data/{package_name}' 경로에서 확인 가능하였다. 또한 SnapEdit와 Background Eraser(Inshot)는 가장 최근에 편집한 이미지만을 확인할 수 있었으며, Photo Studio는 편집에 대한 설정과 작업 내용을 저장해주는 기능으로 이미지를 편집할 시, 편집된 이미지를 남기는 것을 확인할 수 있었다.

4.2.2 조작된 영역

대부분의 애플리케이션에서 조작된 영역을 식별할 만한 아티팩트를 남기지 않았다. 그러나 SnapEdit의 경우 Fig. 7.의 위측과 같이 객체를 제거한 영역에 대한 Mask를 남기는 것을 볼 수 있었으며, Photo Studio에서도 Fig. 7.의 아래측과 같이 (편집시간정보).jpg로 가장 최근에 편집한 이미지에 대해서

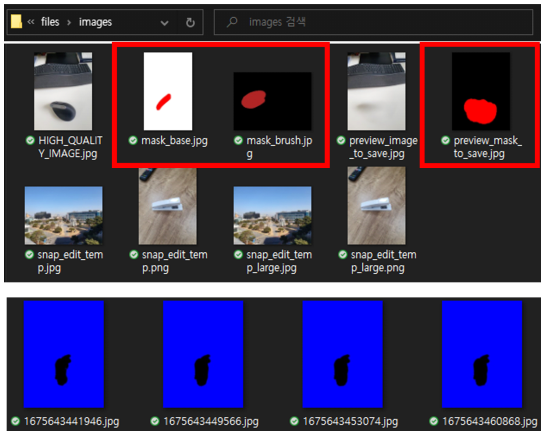


Fig. 7. Identification of manipulated regions through mask

Mask를 저장함을 확인할 수 있었다. 이는 사용자가 어떠한 객체를 지웠는지 또는 남겼는지에 대한 실제 행위에 대한 정보를 명확히 식별할 수 있도록 한다.

4.2.3 원본 이미지

11개 애플리케이션 중 8개 애플리케이션에서 편집에 사용된 원본 이미지를 확인할 수 있었다. 대부분 패키지가 저장되는 경로에 원본 이미지가 존재했으나 Meitu와 Background Eraser(Inshot)의 경우 개인 외부 저장소(Private External Storage)에서 확인이 가능했다. Samsung PhotoEditor 또한 특정 경로에서 원본 이미지를 확인할 수 있었다. 이 에디터는 원본 파일 자체에 편집된 설정을 덮어쓰는 '저장'과 '다른 파일로 저장' 두 가지의 저장 방식을 지원한다. 일반적으로 '저장' 기능을 사용할 경우, Fig. 8.과 같이 원본 복원(Revert) 기능을 통해 편집하기 전으로 복원할 수 있다. 이때 원본 이미지는 'data/sec/photoeditor/0/storage/emulated/0/DCIM/Camera/{원본파일명 파일}' 경로에 저장된다. Samsumg PhotoEditor는 삼성 스마트폰에 탑재된 기본 사진 편집 앱 중 하나로, 서드파티앱을 다운로드하지 않아도 고품질의 사진 편집 작업을 수행할 수 있도록 한다. 이러한 접근성이 높은 기본 앱에 대한 분석은 포렌식 조사에 크게 기여할 수 있다.

Fig. 8. Original image identification path when using 'Save' function

4.2.4 편집 로그

편집 로그는 이미지 조작 행위를 식별하는 데 중요한 역할을 한다. Meitu에서는 편집된 이미지의 저장 경로, 사용된 원본 이미지 파일명, 편집된 이미지 파일명, 편집 시작 시간, 사용한 편집 기능 등이 로그로 기록된다. Adobe Photoshop Fix와 Photo Studio는 이미지를 프로젝트 단위로 편집할 경우, 프로젝트명, 프로젝트 생성 시간, 수정 시간 등이 로그로 남는다. Background Eraser(Inshot)의 경우에도 편집된 이미지 파일명, 편집 시작 시간, 저장 시간 등이 확인 가능했다. 이러한 로그 파일들은 대부분 Timestamp를 포함하여 저장되므로, 이미지 조작 행위 식별에 큰 도움이 된다.

4.2.5 캐싱 이미지

이미지 편집 도구는 편집이미지를 갤러리에 저장하기 위해 첫 사용시, 접근 권한 허용 여부를 묻는다. 이때 갤러리에 저장되어 있는 이미지 또한 캐싱이 되며, 갤러리에서 이미지를 삭제하더라도 갤러리에 남아있던 이미지를 확인할 수 있다. Fig. 9.는 캐싱된 이미지가 저장되는 '/data/data/{package_name}/cache/image_manager_disk_cache/' 경로에 존재하는 파일들을 보여준다. '(random_num).0'이라

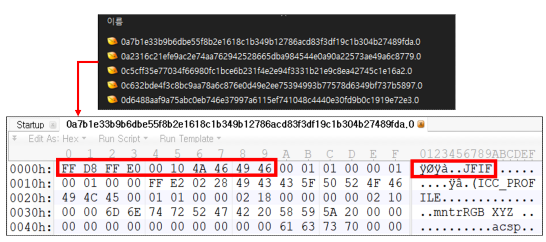


Fig. 9. Cache file with JPEG signature

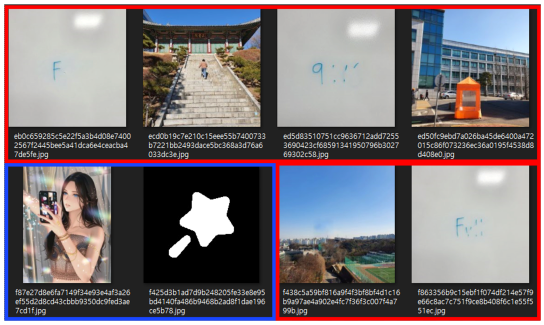


Fig. 10. Image editing tool that even caches gallery images (red borders)

는 파일명으로 존재하는 캐싱된 이미지는 JPEG 시그니처를 가지며 Fig. 10.과 같이 앱 자체에서 사용하는 이미지와 갤러리에 저장된 이미지로 구성된다. 이는 사용자의 안티포렌식 행위 식별에 유용하게 쓰일 수 있다.

V. 방법론

본 장에서는 분석 내용을 종합하여 이미지 조작을 탐지하기 위한 방법론을 제시한다.

5.1 제안하는 방법론

본 연구에서 제안하는 이미지 조작 탐지 방법론은 세 단계로 구성되며, 그중 이미지 처리 기술 적용은 이미지 조작 탐지를 위해 사용되고 있는 기존 기술의 활용을 의미한다.

첫 번째 단계인 이미지 파일 분석에서는 분석 대상 이미지를 획득한 후, 이미지가 조작되었는지 확인하기 위해 조작 탐지를 위한 기준인 Reference DB를 참조한다. 이를 위해 분석 대상 이미지의 Exif, DQT, 그리고 Filename Signature를 파싱 및 분석하여 조작 여부를 판단한다. 이를 통해 조작 여부와 어떠한 이미지 편집 도구를 사용하였는지에 대한 이미지 마지막 출처를 확인할 수 있다.

두 번째 단계인 이미지 처리 기술 적용에서는 기존의 이미지 처리 기법을 적용하여 조작된 영역을 탐지할 수 있도록 한다. 조작 탐지에 사용되는 기법에는 Noise Analysis, ELA(Error Level Analysis), PCA(Principal Component Analysis), Luminance Gradient 등이 있다. 이러한 기술을 통해 그래픽적 불규칙성, 불일치를 파악하여 조작으로 인한 변형된 영역을 식별할 수 있다.

마지막 단계인 모바일 포렌식 아티팩트 분석에서는 이미지 편집 도구가 설치된 모바일 기기를 획득할 경우, 관련 패키지가 설치된 경로에서 데이터를 추출 및 분석함으로써 이미지 조작을 탐지한다. 구체적인 조작 식별을 위해 고려해야 할 정보는 분석이 필요한 이미지 편집 도구 아티팩트가 되며, 추가적으로 모바일 기기 자체의 시스템 아티팩트를 분석함으로써 도구 실행 및 이미지 삭제 등과 같은 행위 관련 정보를 식별할 수 있도록 한다.

5.2 논의

본 장에서는 방법론에서 제시된 각 기법의 장단점을 검토하고, 이러한 기법 간의 상호보완의 필요성에 대한 논의를 진행한다. 이미지 파일 분석과 이미지 처리 기술 적용은 이미지 파일을 대상으로 분석을 수행하며, 모바일 디지털 포렌식 아티팩트 분석은 모바일 기기를 대상으로 분석을 수행한다.

이미지 파일 분석은 파일 내부의 메타데이터 시그니처를 기반으로 하여 간단하고 빠른 분석이 가능하다는 장점이 있다. 그러나 새로운 앱에 대한 사전 분석이 필요하며, 파일 전송이나 화질 변경과 같은 재인코딩이 발생하는 경우 메타데이터가 손상될 수 있다. 또한, 사용자가 쉽게 의도적으로 변경할 수 있는 단점이 있다.

이미지 처리 기술 적용은 이미지의 그래픽 데이터 자체를 분석하여 파일 내부의 메타데이터 변경에 강건한 분석이 가능하며, 조작된 영역에 대해서도 탐지할 수 있는 장점이 있다. Fig. 12.는 Separable Median Filter를 통해 조작된 영역에서 노이즈 패턴을 생성하는 Noise analysis를 자동화한 도구를 사용하여 조작된 영역을 탐지한 결과이며, (a)는 탐지된 예를 보여준다. 그러나 이는 (b)와 같이 항상 오탐과 미탐의 가능성이 내제되어 있으며, AI의 발전으로의 정교한 이미지 편집 기술로 인해 이미지 처리 기술을 통한 조작 탐지가 더욱 어렵다는 한계를 지닌다. 이러한 정교한 조작 기술이 스마트폰 앱에서 기본적으로 제공되는 기능을 통해 쉽게 생성될 수 있어 단점이 더욱 부각된다.

모바일 포렌식 아티팩트 분석은 이미지 조작이 언제 이루어졌는지와 조작된 영역을 확인할 수 있으며, 경우에 따라 조작된 이미지의 원본 파일도 획득할 수 있다. 이는 조작과 관련된 구체적인 정보를 제공하지

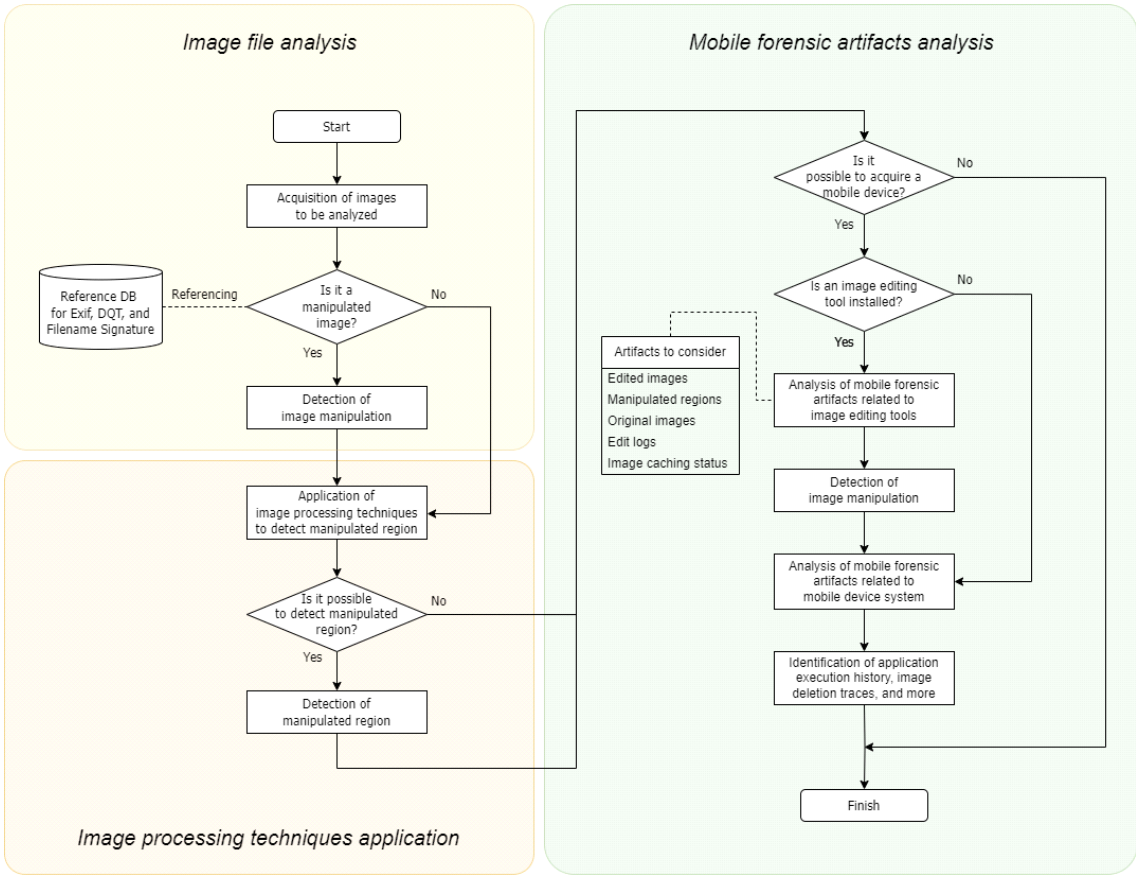


Fig. 11. Methodology for image manipulation detection

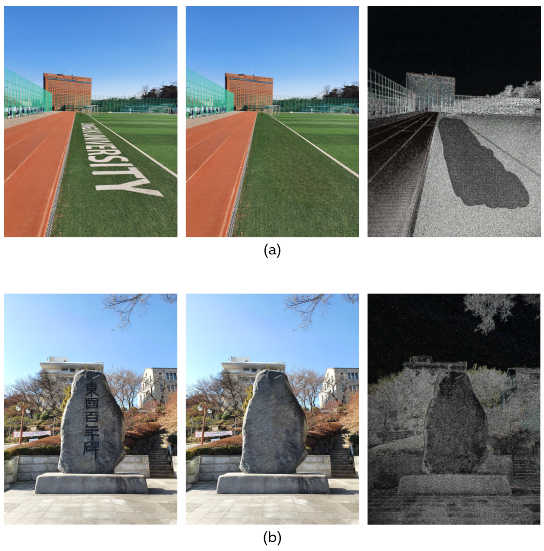


Fig. 12. Examples of object removal and noise analysis

만, 분석 대상 이미지를 편집한 스마트폰을 확보해야 하고, 해당 앱이 삭제되지 않은 경우에만 적용할 수 있다는 단점이 있다.

따라서 본 연구에서는 각 기법들을 하나의 방법론으로 통합하여 단점을 상호 보완할 수 있도록 하였다. 예를 들어, 메타데이터가 손상되더라도 이미지 처리 또는 모바일 포렌식 아티팩트를 통해 이미지 조작 여부와 조작 영역을 확인할 수 있게 되며, 이미지 처리 기술로 이미지 조작이 감지되지 않는 경우에는 제시한 다른 단계에서 조작을 확인할 수 있도록 한다. 예로 이미지 처리 기술로 조작을 확인할 수 없는 (b)의 경우, Exif 정보에서 이미지 편집 도구 시그니처인 'AdvaSoft TouchRetouch'(Table 3. 참조)이 확인이 되고, DQT 분석을 통해서 'Remove Unwanted Object'(Table 4. 참조) 이미지 편집 도구인 것이 식별되므로 이미지 조작을 확인할 수 있다.

VI. 결 론

본 연구는 이미지 파일 분석, 이미지 처리 기법 적용, 모바일 포렌식 아티팩트 분석을 통해 이미지 조작 탐지를 위한 포괄적인 접근 방식을 제공한다.

이미지 파일 분석에서는 파싱된 Exif와 DQT, Filename Signature를 Reference DB의 참조를 통해 이미지 조작을 탐지하도록 하며, 모바일 포렌식 아티팩트 분석을 통해서는 이미지 조작 탐지를 위해 고려해야 할 정보를 기반으로 이미지 편집 도구 아티팩트를 분석함으로써 이미지 조작 행위를 식별하도록 한다. 이미지 편집 도구에서 확인된 이미지 조작 관련 아티팩트는 디지털 포렌식 수사 및 분석 과정에서 유의미하게 활용될 수 있음을 보여준다. 제안한 방법론은 각 기법들에 대한 단점을 상호보완할 수 있도록 하여 이미지 조작 탐지의 정확도를 향상시키는 데 크게 기여할 수 있도록 한다.

본 연구에서 생성한 조작된 이미지 데이터셋은 다양한 응용 프로그램의 후처리 단계에서 그래픽 차이를 분석하는 연구 논문이나 조작 영역 탐지에 초점을 맞춘 논문에 활용될 수 있으며, 이미지 조작 탐지의 기준이 되는 Reference DB와 이미지 메타데이터 파싱 코드는 이미지 조작 탐지를 목적으로 하는 연구에 유의미하게 활용될 수 있다.

향후 연구에서는 이미지 파일 포맷의 다양성을 고려하여 분석 범위를 확장하고, 모바일 기기에서 유의미한 사용자 행위를 식별할 수 있도록 모바일 시스템 아티팩트를 분석할 예정이다.

References

[1] Z. Qin, Q. Zeng, Y. Zong, and F. Xu, "Image inpainting based on deep learning: A review," Displays, vol. 69, Sep. 2021.

[2] L. Zheng, Y. Zhang, and V. L. L. Thing, "A survey on image tampering and its detection in real-world photos," Journal of Visual Communication and Image Representation, vol. 58, pp. 380-399, Jan. 2019.

[3] R. Thakur and R. Rohilla, "Recent advances in digital image manipulation detection techniques: A brief review," Forensic Science International, vol. 312, Jul. 2020.

[4] J. Fan, H. Cao, and A. C. Kot, "Estimating EXIF Parameters Based on Noise Features for Image Manipulation Detection," IEEE Transactions on Information Forensics and Security, vol. 8, no. 4, pp. 608-618, Apr. 2013.

[5] N. A. N. Azhan, R. A. Ikuesan, S. A. Razak, and V. R. Kebande, "Error Level Analysis Technique for Identifying JPEG Block Unique Signature for Digital Forensic Analysis," Electronics, vol. 11, no. 9, p. 1468, May. 2022.

[6] M. Zimba and S. Xingming, "DWT-PCA (EVD) Based Copy-move Image Forgery Detection," International Journal of Digital Content Technology and its Applications, vol. 5, no. 1, pp. 251-258, Jan. 2011.

[7] H. Cai, "Luminance gradient for evaluating lighting," Lighting Research & Technology, vol. 48, no. 2, pp. 155-175, Nov. 2013.

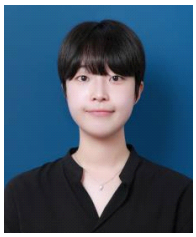
[8] Y. Rao and J. Ni, "A deep learning approach to detection of splicing and copy-move forgeries in images," 2016 IEEE International Workshop on Information Forensics and Security (WIFS), pp. 1-6, Dec. 2016.

[9] T. Wang, M. Liu, W. Cao, and K. P. Chow, "Deepfake noise investigation and detection," Forensic Science International: Digital Investigation, vol. 42, Jul. 2022.

[10] A. Rossler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Niessner, "FaceForensics++: Learning to Detect Manipulated Facial Images," Proceedings of the IEEE/CVF International Conference on Computer

Vision (ICCV), pp. 1-11, Oct. 2019.
〔11〕 J. H. Bappy, A. K. Roy-Chowdhury, J. Bunk, L. Nataraj, and B. S. Manjunath, "Exploiting Spatial Structure for Localizing Manipulated Image Regions," Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 4970-4979, Oct. 2017.
〔12〕 Minsik Kim, Doowon Jeong, and Sangjin Lee, "Building a Database of DQT Information to Identify a Source of the SmartPhone JPEG Image File," Journal of the Korea Institute of Information Security & Cryptology, 26(2), pp. 359 - 367, Apr. 2016.
〔13〕 Uk Hur, Soram Kim, Eunhu Park, Sumin Shin, and Jongsung Kim, "Study on image distribution device identification using JPEG image compression information," Journal of Digital Forensics, 14(1), pp.33-44, Mar. 2020.
〔14〕 J. D. Kornblum, "Using JPEG quantization tables to identify imagery processed by software," Digital Investigation, vol. 5. pp. S21 - S25, Sep. 2008.
〔15〕 Dohyun Kim, Yunho Lee, and Sangjin Lee, "Mobile forensic reference set (MFReS) and mobile forensic investigation for android devices," The Journal of Supercomputing, vol. 74, no. 12, pp. 6618 - 6632, Dec. 2017.

〈저자소개〉



이 지 원 (Jiwon Lee) 학생회원
2022년 8월: 동국대학교 경찰사법대학 경찰행정학부 졸업
2022년 9월~현재: 동국대학교 일반대학원 경찰행정학과 석사과정
〈관심분야〉 디지털 포렌식, 정보보호, 인공지능 등

전 승 제 (Seungjae Jeon) 학생회원
2023년 2월: 동국대학교 경찰사법대학 경찰행정학부 졸업
2023년 3월~현재: 동국대학교 일반대학원 경찰행정학과 석사과정
〈관심분야〉 디지털 포렌식, 정보보호, 스마트폰 포렌식 등

박 윤 지 (Yunji Park) 정회원
2022년 8월: 동국대학교 경찰사법대학 경찰행정학부 졸업
2022년 9월~현재: 동국대학교 일반대학원 경찰행정학과 석사과정
〈관심분야〉 디지털 포렌식, 메타버스, 정보보호 등

정 재 현 (Jaehyun Chung) 학생회원
2023년 2월: 동국대학교 경찰사법대학 경찰행정학부 졸업
2023년 3월~현재: 동국대학교 일반대학원 경찰행정학과 석사과정
〈관심분야〉 디지털 포렌식, 정보보호, 가상현실, 사이버보안 등

정 두 원 (Doowon Jeong) 정회원
2019년 2월: 고려대학교 정보보호대학원 공학박사
2020년 9월~현재: 동국대학교 경찰사법대학 조교수
2022년 1월~현재: 동국대학교 융합안전학술원 사이버안전연구센터 센터장
〈관심분야〉 디지털 포렌식, 정보보호 등