

고해상도로 찍은 이미지에서의 손가락 지문 채취 방지에 관한 연구

윤원석¹, 김상근^{2*}

¹성결대학교 컴퓨터공학과 학생, ²성결대학교 컴퓨터공학과 교수

A study on Prevent fingerprints Collection in High resolution Image

Won-Seok Yoon¹, Sang-Geun Kim^{2*}

¹Student, Department of Computer Engineering, Sungkyul University

²Professor, Department of Computer Engineering, Sungkyul University

요약 본 연구에서는 나날이 발전하는 카메라의 해상도 기술과 SNS의 이미지 공유를 통해서 고해상도로 찍은 이미지를 손쉽게 구할 수 있고, 이미지를 통해서 사람의 손가락 지문을 손쉽게 채취하여 이를 악용할 수 있다는 가능성을 고려해 이를 방지하는 기술을 제시한다. 이 기술을 개발하기 위해서는 Python 언어를 이용한 Opencv와 opencv안의 Blur 처리를 해주는 라이브러리 등을 사용한다. 우선 이미지에서 손을 찾아주기 위해서 딥러닝 기반의 학습된 Hand Key point Detection 알고리즘을 사용한다. 이 알고리즘을 이용해 손가락 마디를 찾아 이 마디의 좌표를 이용해 이미지에서의 손가락 지문 부위만을 따로 blur 처리를 해줌으로써 원본 이미지에서의 손상을 최소화하면서 손가락 지문을 보호할 수 있다. 향후 정확한 손가락 추적 알고리즘의 개발로 스마트폰 카메라 app의 내부 옵션으로 사용하여 고해상도의 이미지에서의 지문을 보호할 수 있을 것이다.

주제어 : 손 관절 인식, 지문 해킹 방지, 이미지 보안, 고해상도 기술 문제점, 딥러닝

Abstract In this study, Developing high resolution camera and Social Network Service sharing image can be easily getting images, it cause about taking fingerprints to easy from images. So I present solution about prevent to taking fingerprints. this technology is develop python using to opencv, blur libraries. First of all 'Hand Key point Detection' algorithm is used to locate the hand in the image. Using this algorithm can be find finger joints that can be protected while minimizing damage in the original image by using the coordinates of separate blurring the area of fingerprints in the image. from now on the development of accurate finger tracking algorithms, fingerprints will be protected by using technology as an internal option for smartphone camera apps from high resolution images.

Key Words : Detecting Finger Joints, Fingerprinting Prevent, Images Security, High Resolution Image Problems, Deep Learning

1. 서론

오늘날 우리는 인터넷과 스마트폰의 발전을 통해 시공간의 제약을 받지 않고 언제 어디서나 손쉽게 SNS(Social Network Service)를 통해서 자신의 생활 혹은 개인의 일상 등을 사진을 통해서 다른 사람들과 공유할 수 있다. 그러나 문제점은 이 공유되는 사진 속에 있다. 사진을 찍을 때 한 번쯤은 손가락 모양을 'V'

자로 하고 사진을 찍은 경험이 있을 것이다. 이러한 디지털 이미지 속의 지문이 드러난 이미지 데이터를 이용해 지문 부위를 채취하고 채취한 지문을 통해 지문보안 프로그램의 보안을 통과한 사례가 있다[1]. 이렇듯 카메라의 기술이나 디지털카메라의 노이즈 감소[2] 같은 기능들이 발전할수록 해상도는 더욱 높아질 것이고 디지털 이미지 속에서 지문의 채취는 더욱 쉬워져 해킹의

*Corresponding Author : Sang-Geun Kim(sgkim@sungkyul.ac.kr)

가능성이 커지고 있다. 따라서 본 연구는 카메라 기술의 발달로 고해상도 이미지 속의 지문을 채취를 방지하는 시스템을 설계하고 개발하였다. 시스템 개발을 위해서는 이미지 속에서 손과 손의 골격을 인식해주는 기술을 이용해 이미지 데이터에서 손의 골격을 좌뿔값으로 나타내 주는 기술이 필요하다. 이 좌뿔값을 이용하여 이미지 속에서 손가락의 지문 부위만을 구하는 알고리즘을 개발해 원본 이미지에서 최소한의 손상 없이 손가락 지문 부분만 해킹 불가능할 정도로 수정하는 것을 목표로 시스템을 개발하고 연구를 진행하였다.

본 논문의 구조는 다음과 같다. 2장에서는 본 논문에 관련된 기술 및 연구에 관해 설명하고, 3장에서는 지문 채취 방지 연구에 필요한 알고리즘, 4장에서는 지문 채취 방지 시스템을 구현한 결과를 기술한다. 5장에서 결론을 맺고 향후 연구에 관해 설명한다.

2. 관련 연구

본 연구 주제와 관련된 연구를 살펴보면은 기본적으로 컴퓨터 비전기술을 사용한다.

2.1 컴퓨터 비전

컴퓨터 비전은 인공지능 분야인 딥러닝기술을 활용해 특정 객체에 속한 디지털 이미지 데이터를 속한 객체에 맞게 식별하고 분류하는 학습을 통해 딥러닝 모델을 만들어 컴퓨터가 학습 데이터 외의 이미지 데이터를 식별할 수 있게 사용하는 기술이다. 기본 원리는 Fig. 1과 같이 이미지 데이터의 특징들을 추출하여 학습된 모델에 있는 특징들과 같은 특징들을 찾아가며 마지막의 결괏값을 확률적으로 계산하여 특정 객체임을 식별하는 원리이다.

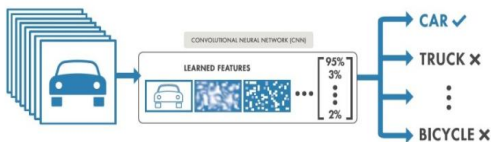


Fig. 1. Convolution neural network

컴퓨터 비전의 구조적인 측면은 영상 혹은 이미지에서의 인식하고자 하는 목표의 특징을 추출하기 위해 물체의 경계를 찾는다. 이미지 데이터에서는 특정 물체의 경계 외의 부분을 0 혹은 255 두 가지 값으로 나누어 이미지에서 물체를 구분하기 쉽게 한다. 이미지의 경계

를 이용해 그 물체의 둘레, 면적, 밝기, 경계선의 굴곡 등 여러 가지 특징들을 추출하고 물체가 회전했을 때의 특징들도 모두 추출한다. 이를 바탕으로 물체의 여러 이미지의 특징 데이터를 학습 데이터로 사용하여 그 물체를 인식할 수 있게 모델링을 진행하는 방식의 구조이다. 컴퓨터 비전의 구조적인 흐름도는 Fig. 2와 같다.

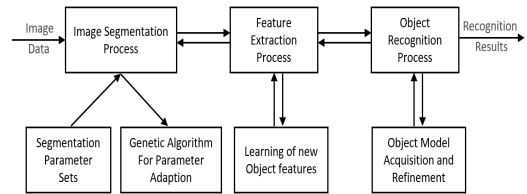


Fig. 2. Computer Vision Processing

2.2 기존 인식 기술

기존의 인식 기술에는 Kinect를 사용해 모션을 인식하여 모션에 따라 기기를 조작하는 연구가 진행되었다. J. H Park[3]은 이 Kincet 기술을 이용하여 골격의 위치 값을 가지고 환풍구나 창문 등의 개방과 폐쇄를 요구하는 연구를 진행하였다. T. Simon[4]은 Kinect 기기를 이용하지 않고 영상 데이터에서 사람의 골격 모형을 학습시켜 신체의 골격 모형을 영상 데이터를 통해 인식하는 기술연구를 진행하였다. S. H. Kim[5]의 경우에는 미니 드론을 조종하면서 드론에 부착된 카메라가 촬영하는 영상을 실시간으로 받아들여 특정인의 얼굴을 인식하여 확인시켜주는 연구를 하였다.

본 연구에서 진행한 손가락의 지문 부위를 찾기 위해서는 손 인식 기술을 활용한 연구의 동향을 살펴볼 필요가 있다. 현재까지 손 인식 기술을 연구한 J. C. Han[6]의 연구처럼 CNN을 이용해 손 모양의 제스처를 인식해 사용자의 의사결정에 도움을 주는 연구가 활발히 진행 중이다. C. Y. Jang[7]은 디지털카메라의 영상과 Kinect의 스킴레톤 정보를 이용하여 사용자의 의사 결정을 판단하고 손 제스처를 인식하는 연구를 하였다. 하지만 이 방법은 손의 전체적인 뼈대만을 나타내어 뼈 마디마디의 좌뿔값을 알아내기는 어렵다. T. Simon[8]은 손의 전체적인 인식이 아니라 손의 마디의 Key Point를 통해 대략적인 추정치를 얻어 손가락 마디의 위치를 검출해내고 이미지에서 그 위치의 좌뿔값을 검출해내는 연구를 진행하였다.

손 인식의 연구 뿐만 아니라 지문 인식에 관한 연구에는 스마트폰에 사용되는 지문 인식 기술의 원리는 정

전 센서를 이용해 Fig. 3과 같이 지문 방향의 특징점인 에지(Edge)를 검출해 방향 지도를 구축한다.

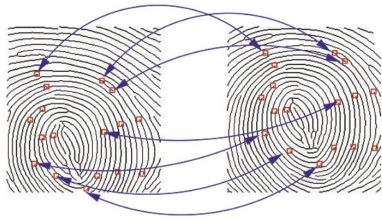


Fig. 3. Fingerprint matching based on Edge

이 기술은 극소의 지문 부위를 채취하기 때문에 여러 번의 방향을 채취하여 이를 이어 붙여 지문을 인식하는 기술이다[9]. 현재 이 지문 인식이 최신 기술의 발전으로 위변조 및 검출에 대한 위험성을 알리고 있다[10]. 인식 기술에 대한 선행연구의 특성과 본 연구가 요구하는 특성에 대한 비교는 Table 1에 제시하였다.

본 연구는 급속도로 발전하는 기술에서 지문 채취가 가능하다는 문제점을 분석하여 지문 채취를 방지하기 위해 이미지에서 지문 부위만의 해상도를 강제로 뭉개는 방법을 제시한다.

Table 1. Comparison with previous recognition research

Research	Factors			
	1	2	3	4
A. Jain, L. Hong & R. Bolle. (1997)	Finger prints	Technological	Fingerprints recognition system	Capacitive Sensor
J. H. Park & H. J. Park. (2017)	Body Skeleton	Comfortability	Body Motion recognition system	Using Kinect
C. Zhe, T. Simon, S. E. Wei & Y. Sheikh. (2017)	Body Skeleton	Technological	Detecting Body skeleton system	(R, G, B) Cam
S. H. Kim. (2019)	Face	Technological	Using Drone, Flying drone recognize face	(R, G, B) Cam
J. C. Han & K. J. Cheoi. (2019)	Hand	Comfortability	Using CNN Model, Hand gesture recognition	(R, G, B) Cam
C. Y. Jang & T. Y. Kim. (2020)	Hand	Comfortability	Using Kinect, Hand gesture Recognition	Using Kinect
This research	Hand Skeleton	Security	(R, G, B) Cam, Hand Skeleton Detection, Erase fingerprints	(R, G, B) Cam

1 : Detection, 2 : Purpose Of Use, 3 : Features, 4 : Recognition Device

3. 지문 채취 방식을 위한 설계 및 시스템 구성

지문 채취 방식을 위한 시스템은 사람의 손가락이 드러난 이미지 데이터를 입력 값으로 넣어 이미지 속에서 손가락의 골격을 인식하는 Hand Skeleton 딥러닝 모델[11]을 이용해 이미지에서 손가락 골격의 점들의 좌표값을 추출한다.

손가락의 지문 그려진 끝 마디의 좌표값을 이용해 좌표값을 연결한 길이 만큼을 빗변으로 두어 직각삼각형을 만들어 직각삼각형의 세 변의 길이, 빗변의 기울기와 끝 마디를 연결한 직선의 중심의 좌표값, 중심의 좌표로부터 지문 부위를 그릴 장축과 단축의 반지름값을 알고리즘을 통해 추출한다.

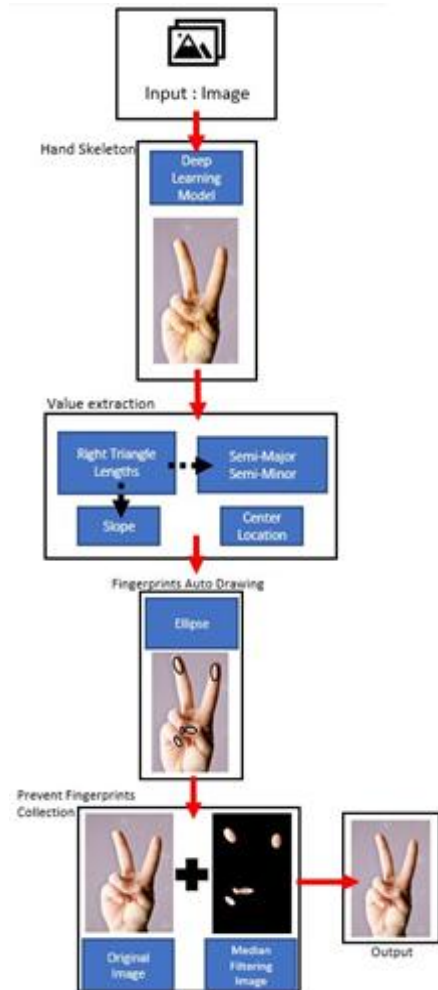


Fig. 4. System Flow

추출한 값들을 이용해 지문 부위를 타원 형태로 그리고 이미지 blurring을 진행을 위해 타원을 그린 이미지에서 타원을 그린 부분만 데이터값을 따로 뽑아 median filtering을 진행하여 원본의 이미지와 blur처리를 한 이미지를 XOR 연산을 통해 합성하여 결괏값으로 지문 채취를 방지한 이미지를 얻는다. 시스템 흐름의 구성은 Fig. 4와 같다.

3.1 Hand Skeleton

본 연구에서는 이미지에서 손가락의 뼈마디를 인식해서 표현한 Hand Key Point Detection using Deep Learning and Opencv[10] 연구를 기반으로 진행하였다. 연구에 쓰인 학습된 모델은 손가락의 각 마디 끝점에 위치를 학습시켜 이미지 데이터에서 손의 위치에 따른 총 21개의 뼈마디 점을 이용하여 손의 형태를 표현하였다. 이 21개의 점은 이미지에서 X, Y 좌표값으로 나타내기 때문에 본 연구에서는 이 손가락 마디 점의 좌표값을 이용하였다. Fig. 5는 21개의 점을 통해 손가락 형태를 표현한 이미지이다.

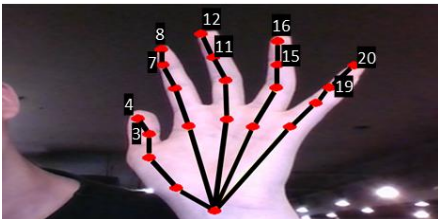


Fig. 5. Hand Key Point Detection

3.2 Value Extraction

이미지를 통해 인식한 21개 점의 좌표값 데이터를 이용해서 지문에 필요한 데이터를 따로 추출해줄 필요가 있다. 지문 부위만을 따로 추출하기 위해 총 21개 점의 Point 중 Fig. 6에 보이는 손가락 첫 마디 부분인 10개 점의 좌표값을 추출한다.

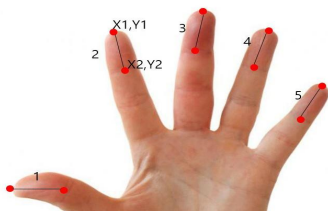


Fig. 6. Fingerprints Data Location:

3.2.1 Value Extraction - Right Triangle Lengths

손가락 첫 마디의 10개 점에서 값 추출을 위하여 Fig. 5의 Point F1[4, 8, 12, 16, 20]과 Point F2 [3, 7, 11, 15, 19]을 구분 지어 저장한다. 각 Point에는 이미지에서의 (x, y) 좌표값을 가지고 있다. 이 Point를 활용하여 먼저 끝 마디의 길이를 구하기 위해 값을 추출한다.

Input : Point Finger_top F1[4, 8, 12, 16, 20], Point Finger_second F2[3, 7, 11, 15, 19]

Output : A(Adjacent), O(Opposite), H(Hypotenuse)

Start

```
for int i =0; i < size(F1); i++; :
```

$$A = F1_x[i] - F2_x[i] \tag{1}$$

$$O = F1_y[i] - F2_y[i] \tag{2}$$

$$H = \sqrt{A^2 + O^2} \tag{3}$$

```
return H
```

End

디지털 이미지 데이터에서 기준점은 가장 왼쪽 위의 가장자리를 (0, 0)으로 정의한다. 수식1과 수식2에서는 손가락 점의 끝 마디를 f1과 f2를 직선으로 이은 빗변으로 둔 직각삼각형을 그렸을 때 수식 1 같은 경우에는 밑변을 구할 수 있다. 수식 2 같은 경우에는 f1과 f2를 빗변으로 둔 직각삼각형의 높이를 구할 수 있게 된다. 이제 끝 마디의 길이를 구하기 위해서 수식 3을 이용해서 빗변의 길이를 구해 손가락의 끝 마디의 길이를 구할 수 있다.

3.2.2 Value extraction - Center Location

지문 부위를 표현하기 위해서는 이미지에서의 지문 부위의 중앙 좌표값이 필요하다. 중앙 좌표값은 F1, F2 Point의 중간에 위치한 값으로 이미지에서 이 지문의 중앙의 좌표를 구하기 위한 알고리즘은 아래와 같다.

Input : Point Finger_top F1[4, 8, 12, 16, 20], Point Finger_second F2[3, 7, 11, 15, 19]

Output : Center Location(X, Y)

Start

```
for int i = 0; i < size(F1); i++; :
```

$$C(X, Y) = \frac{F1_x + F2_x}{2}, \frac{F1_y + F2_y}{2} \quad (4)$$

return C
End

지문의 중앙을 구하기 위해서는 각 손가락 끝 마디의 점들을 이용한다. Fig. 6의 이미지에서 찍힌 끝 마디의 두 점의 좌표에서 X축의 좌표의 합과 Y축의 좌표의 합을 구한 후 이 각 합에 2를 나누어 주어 구해진 X, Y의 뜻값이 끝 마디 손가락의 지문의 중앙값이다. 이를 손가락마다 반복해준 결과 모든 손가락의 중앙값을 알 수 있다.

3.2.3 Value extraction - Slope

이미지에서는 손가락의 방향이 매번 틀리기 때문에 이를 위해 손가락의 방향에 따른 기울기 값을 구해줄 필요가 있다. 각 손가락의 기울기를 구하기 위해선 수식 1, 3을 이용해 얻은 각 손가락의 빗변의 길이와 밑변의 길이가 필요하다.

Input : A[5](Adjacent), H[5](Hypotenuse)

Output : S(Slope)

Start

for int i = 0; i < size(A); i++; :

$$\angle S = \text{Acos}\left(\frac{A[i]}{H[i]}\right) \quad (5)$$

$$\text{if } F1_y[i] \geq F2_y[i] \quad (6)$$

$$\angle S = \angle S - 180$$

Else

$$\angle S = 180 - \angle S$$

return $\angle S$

End

수식 5에서 사용한 식은 역삼각함수를 이용하여 밑변을 빗변으로 나누어진 값에 역코사인 함수를 활용해서 각 손가락의 방향을 나타내는 기울기 값을 구할 수 있었다. 수식 6에서는 $F1_y$ 와 $F2_y$ 는 손가락의 방향에 따른 기울기의 정규화가 필요하다. 기울기에 대한 값은 Fig. 7에서 보이는 것처럼 수평선을 기준으로 위쪽 방향을 가리킬 때와 아래쪽 방향을 가리킬 때 두 가지 조건으로 나뉘게 된다.

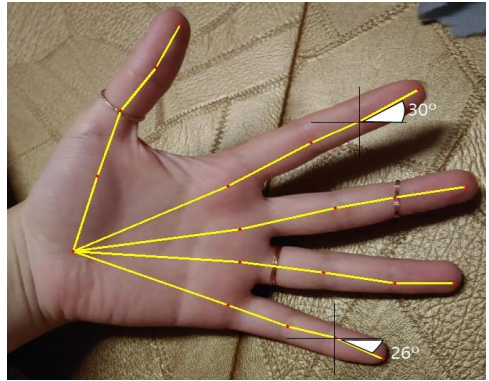


Fig. 7. Finger Slope Condition

현재 기울기의 값은 1사분면의 X축을 기준으로 위아래의 방향으로 기울기 값이 구해져 있다. 그러나 손가락 방향에 따른 타원의 지문 형태를 나타내기 위한 함수의 기울기는 제 2 사분면의 X축을 기준으로 양의 기울기 값이면 시계방향으로 기울기만큼 꺾고 음의 기울기이면 반시계방향으로 기울기만큼 꺾기 때문에 이에 맞춰서 구한 기울기 값을 수정해야 한다. 손가락이 아래쪽을 가리킬 때 이미지의 좌푯값 왼쪽 위(0, 0)를 기준으로 $F1_y$ 의 값이 더 큰 값을 가지기 때문에 $\angle S$ 에서 180도를 빼서 사용한다. 반대로 아래쪽일 때는 $F2_y$ 의 값이 더 큰 값이기 때문에 180도에서 $\angle S$ 를 빼서 기울기 값을 수정하고 함수에 사용한다.

3.3 Fingerprints Auto Drawing

본 연구에서 지문 부위를 그리기 위해 Python 언어를 사용한다. Opencv[12] 라이브러리의 Ellipse 함수를 사용해 타원의 형태로 지문을 표현한다. 이때 필요한 Input 값에는 지문의 중심점 좌푯값, Semi-Major와 Semi-Minor의 길이, 수식 5를 통해 구한 손가락 방향의 Slope 값이 필요하다. Semi-Major는 빗변의 길이가 타원의 지름이기 때문에 지름의 반을 구한다. Semi-Minor 같은 경우에는 지문 형태를 타원으로 생각해 타원의 긴 지름과 짧은 지름의 비율을 약 2:1로 두어 Semi-Minor 값을 구한다.

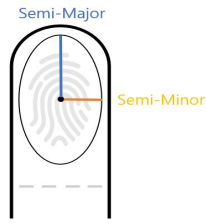


Fig. 8. Semi-Major, Semi-Minor at finger

지문 부위를 표현하기 위해 추출한 Input 값들을 이용하여 이미지 속의 지문에 맞는 타원을 그리기 위해서는 Ellipse 함수 내의 값을 조건에 맞게 넣어줄 필요가 있다.

```
Input : Image, C(X, Y), Semi-Major, Semi-Minor, ∠S
Output : Fingerprints Expression Image
Start
for int i = 0; i < size(Finger); i++;
    cv2.Ellipse(Image, C, (Semi-Major, Semi-Minor),
    ∠S, StartAngle(0), EndAngle(360), Color(R, G, B))
End
```

타원을 그리는 함수를 이용하여 이미지에서의 손가락이 존재할 시 지문 모양의 형태를 나타낼 수 있게 된다. 알고리즘을 통해서 얻은 이미지는 Fig. 9와 같다.



Fig. 9. Fingerprints Expression Image

4. 지문 채취 방지 및 연구결과

본 연구에서는 이미지에서 지문 부위를 찾아 지문 채취를 방지한다. 원본의 이미지에서 지문 부위의 지문을 뭉개기 위해서 이미지 Blur[13]를 이용하여 지문 부위의 해상도를 낮추어 지문 채취를 방지하도록 한다.

4.1 Prevent Fingerprints - Dark Frame Image

이미지에서 지문 부위에 대한 색만을 가져오기 위해

서 가장 먼저 Fig. 9에서 얻은 지문 형태를 나타내는 이미지에서 타원 형태의 도형 내부를 Opencv의 Ellipse 함수의 속성을 수정해 도형 내부를 모두 검은색으로 채워 Fig. 10과 같은 이미지를 얻는다.



Fig. 10. Filled Fingerprints by Black Color

4.2 Prevent Fingerprints - Bitwise XOR Image

Fig. 10의 이미지와 원본의 이미지의 XOR 비트 연산[14]을 통해서 원본의 이미지에서 지문 부분의 색만을 가져올 수 있다. XOR 연산은 두 이미지를 비교하여 같은 색의 이미지는 모두 RGB 색상(0, 0, 0)으로 바꾸어 지문을 제외한 나머지 이미지의 색상이 모두 검은색으로 바뀐 Fig. 11과 같은 이미지를 얻는다.



Fig. 11. Bitwise_XOR Processing Image

4.3 Prevent Fingerprints - Median Filtering Image

Fig. 11의 이미지는 원본 이미지에서의 손가락 지문 부위만을 채취하였다. 이제 이 이미지에서 Blur 처리를 통해 지문 부위의 데이터를 뭉개줄 필요가 있다. 이를 위해 여러 가지 Blur 함수 중에 Median Filtering[15]을 사용한다. 본래 Median filtering은 지정된 픽셀 범위에서 중간 평균값을 이용하여 이미지의 노이즈를 제거하여 이미지를 더욱 깔끔하게 보여 주는 데 사용되지만, 본 연구에서는 반대로 노이즈와 관련 없는 이미지

에서 지문 부위를 뭉개주기 위해 사용하여 Median Filtering 처리를 진행하게 된다. 픽셀의 범위에는 보통 3*3에서 5*5 사이의 값으로 주고 카메라의 화소가 더욱더 발전할수록 픽셀의 범위의 값을 증가시키는 방향으로 진행한다.

4.4 Prevent Fingerprints - Bitwise_OR Image

원본 이미지에서 지문 부위만을 뭉개 이미지를 얻기 위해 Median Filtering으로 지문을 뭉개 이미지와 원본 이미지의 비트 연산을 진행한다. 두 이미지의 합성을 위해 OR 비트 연산을 사용하여 이미지를 합성한다. 원본의 이미지에서 최소한의 손실을 통한 지문 부위만을 뭉개 이미지 Fig. 12를 얻는다.



Fig. 12. Bitwise_OR Processing Image

4.5 연구결과

본 연구결과 이미지 데이터를 입력 값으로 넣었을 때 손가락의 지문을 찾아 원본의 이미지에서 지문 부분만 해상도 흐림 처리를 통해 지문 채취를 방지하는 시스템의 전체적인 알고리즘을 나타낸다.

```

Input : Image
Output : Prevent_Fingerprints_Image
Start
For i = 0; i <= size(All_Point_21); i++;
    Points[i] = Find_Points_caffemodel
Finger_top = Points[[20], [16], [12], [8], [4]]
Finger_second = Points[[19], [15], [11], [7], [3]]
For j = 0; j <= size(Finger1); j++;
    Value Extraction Output - (Adjacent, Opposite, Center
    Location(X,Y), Hypotenuse, Semi-Major, Semi-Minor,
    Slope)
    cv2.Ellipse(Image, (X,Y), (Semi-Major, Semi-Minor),
    
```

```

    Slope, 0, 360, RGB(0,0,0), -1)
return Dark_Frame
Image_XOR = Bitwise_XOR(Dark_Frame, Image)
Image_Blur = MedianBlur(Image_XOR, 3)
Prevent_Fingerprints_Image=add(Image_Blur,
Dark_Frame)
return Prevent_Fingerprints_Image
End
    
```

전체적인 알고리즘을 거친 이미지를 보면 원본의 이미지에서 지문 부위만 흐림 처리한 결과를 Fig. 13을 통해 볼 수 있다. Fig. 13은 SNS에 올린 이미지를 사용하여 실제로 일상의 이미지에서 지문 채취를 방지하는 결과를 도출하였다.



Fig. 13. Prevented Fingerprints Image

확실한 연구결과와 검증의 위해 결과로 추출한 이미지에서 손가락 부분만을 확대하여 결과를 도출해 보았다. Fig. 14의 이미지를 통해서 지문 부분이 흐려진 것을 확인할 수 있다.



Fig. 14. Result Image Validation-1

손가락의 방향이 아래쪽을 향할 때의 지문 또한 Fig. 15와 같이 지문 부위를 그려 채취 방지를 위한 흐림 처리를 진행한 것을 확인할 수 있다.



Fig. 15. Finger Down Result Image Validation-2

좀 더 확실한 연구결과를 위해 실제 화질이 좋은 스마트폰을 사용해 찍은 이미지에서 실행시켜 보았다. Fig. 16은 스마트폰으로 찍은 이미지를 확대하고 지문 채취 방지 시스템을 통한 이미지를 확대한 결과를 비교한 것이다.



Fig. 16. Result Image Validation-3

Fig. 16의 이미지는 스마트폰으로 직접 촬영한 크기 2268 x 4032 크기의 이미지이고 확대했을 때 손가락에 지문의 형태가 드러나 있는 것을 확인할 수 있다. 지문 채취 방지 시스템을 통한 결과를 보면 손가락의 지문 부분의 해상도가 뭉개져 있는 것을 확인할 수 있다. 따라서 고해상도의 이미지에서도 지문 채취 방지 시스템을 통해 지문 채취를 방지할 수 있다는 결과를 얻을 수 있음을 확인할 수 있다.

5. 결론 및 향후 연구

본 연구는 앞으로의 스마트폰 카메라 기술의 발전으로 인해 이미지의 해상도가 증가함으로 사진 내의 손가락에서 지문을 채취할 수 있고 이를 악용하는 사례가 일어나는 것을 방지하고자 이미지에서 지문 부위만을

채취해 그 부분만을 흐리게 처리해주는 시스템을 연구하고 개발하였다. 손가락의 지문 부위만을 구하기 위해 손가락의 좌푯값들을 이용해서 지문 부위만을 구하는 알고리즘을 설계하였고, 이에 필요한 값들을 구하는 식들을 제시하였다. 그 결과 이미지에 따라 손가락의 방향에 상관없이 손을 인식한다면 지문 부위 형태의 타원형을 그리는 것을 구현하였고, Opencv의 비트 연산과 Blur 처리를 이용해 지문 부위만을 흐리게 해서 원본 이미지에서 최소한의 손실로 지문 채취를 방지할 수 있게 된다. 이를 통해 이미지를 통한 개인의 생체데이터를 보호될 것으로 판단된다. 본 연구에서 사용한 Hand Skeleton Detection 알고리즘은 이미지에서 모든 손에 대한 인식이 적절치 못할 때가 있다. 향후 연구에서는 이를 보완하여 좀 더 높은 인식률을 가진 손의 뼈대 인식 알고리즘을 사용하여 좀 더 다양한 환경의 이미지에 적용하여 시스템을 설계하여 개선하고 보완할 필요가 있다.

REFERENCES

- [1] P. Monckton. (2014). *Hacker Clones Fingerprint From Politician's Photograph*(Online). <https://www.forbes.com/sites/paulmonckton/2014/12/30/hacker-clones-fingerprint-from-photograph/#5b9873d46896>
- [2] D. H. Har & H. J. Park. (2009). Effect of Noise Reduction Function in Digital Camera on the Resolution of Digital Image. *The Journal of the Korea Contents Association*, 9(5), 91-98
DOI : 10.5392/JKCA.2009.9.5.091
- [3] J. H. Park, & K. J. Lee. (2017). Realization of user-centered smart factory system using motion recognition. *Journal of Convergence for Information Technology*, 7(6), 153-158.
- [4] C. Zhe, T. Simon, S. E. Wei & Y. Sheikh. (2017). Realtime multi-person 2d pose estimation using part affinity fields. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 7291-7299).
- [5] S. H. Kim. (2019). Development of Face Recognition System based on Real-time Mini Drone Camera Images. *Journal of Convergence for Information Technology*, 9(12), 17-23.
- [6] J. C. Han & K. J. Cheoi. (2017). Design and Implementation CNN for Hand Shape Recognition. *Journal of the Research Institute for*

Computer and Information Communication, 25(1), 43-45.

- [7] C. Y. Jang & T. Y. Kim. (2020). Hand Feature Enhancement and User Decision Making for CNN Hand Gesture Recognition Algorithm. *Journal of The Institute of Electronics and Information Engineers*, 57(2), 60-70.
DOI : 10.5573/ieie.2020.57.2.60
- [8] T. Simon, H. B. Joo, I. Matthews, & Y. Sheikh. (2017). Hand keypoint detection in single images using multiview bootstrapping. *In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. (pp. 1145-1153).
- [9] A. Jain, L. Hong & R. Bolle. (1997). On-line fingerprint verification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4), 302-314.
- [10] W. J. Kim. (2019). Recent Technology Trends on Fingerprint Recognition and Forgery. *The Institute of Electronics and Information Engineers*, 46(8), 31-38.
- [11] V. Gupta. (2018). *Hand Keypoint Detection using Deep Learning and OpenCV* (Online).
<https://www.learnopencv.com/hand-keypoint-detection-using-deep-learning-and-opencv>
- [12] A. Mordvintsev & K. Abid. (2014). *Opencv-python tutorials documentation* (Online).
<https://media.readthedocs.org/pdf/opencv-python-tutorials/latest/opencv-python-tutorials.pdf>
- [13] D. Y. Kang. (2020). *Image data blur processing through object detection*. Master dissertation. HangYang University, Seoul.
- [14] S. Mallick, (2013). *Bitwise AND, OR, XOR and NOT* (Online).
<http://opencvexamples.blogspot.com/2013/10/bitwise-and-or-xor-and-not.html>
- [15] B. Y. Lee. (2002). *Design and analysis of Fuzzy Weighted Median(FWM) filter eliminating image noises*. Master dissertation. KonKuk University, Seoul

윤 원 석(Won-Seok Yoon)

[정회원]



- 2021년 2월 : 성결대학교 컴퓨터공학과 졸업예정(공학사)
- 관심분야 : 컴퓨터비전, 인공지능
- E-Mail : wonseok3629@gmail.com

김 상 근(Sang-Geun Kim)

[정회원]



- 1987년 2월 : 중앙대학교 전자계산학과(이학사)
- 1989년 2월 : 중앙대학교 컴퓨터공학과 (공학석사)
- 1996년 2월 : 중앙대학교 컴퓨터공학과 (공학박사)
- 1996년 3월 ~ 현재 : 성결대학교 컴퓨터공학과 교수
- 관심분야 : 정보보안, 인공지능, 빅데이터, 소프트웨어공학
- E-Mail : sgkim@sungkyul.ac.kr