https://doi.org/10.33778/kcsa.2022.22.4.141

DNN과 블러링을 활용한 홍채 마스킹 보안 강화 기술

백 승 민*, 최 영 해*, 홍 찬 우*, 박 원 형**

요 약

생체정보인 홍채는 지문과 같이 안전하고 유일하며 신뢰성이 다른 생체인증보다 오인식률을 크게 낮출 수 있는 개인정보이다. 그러나 생체인증 특성상 탈취 당하게 된다면 대체가 불가능하다. 실제 홍채 사진을 탈취 후 3d 프린팅 하여 눈이 카메라 앞에 있는 것처럼 작동하도록 한 사례가 있다. 이처럼 고화질의 영상과 사진을 통하여 홍채 유출 가능성이 존재하다. 본 논문에서는 기존의 블러링 기법을 기반으로 한 홍채 영역 마스킹 연구를 보완하여 홍채 마스킹 성능 향상을 제안한다. 본 연구에서 도출된 결과를 토대로 화상회의 프로그램 및 전자기기의 보안에 활용할 수 있을 것으로 기대된다.

Enhancement of Iris Masking Security using DNN and Blurring

Seungmin Baek*, Younghae Choi*, Chanwoo Hong*, Wonhyung Park**

ABSTRACT

The iris, a biometric information, is safe, unique, and reliable, such as fingerprints, and is personal information th at can significantly lower the misrecognition rate than other biometric authentication. However, due to the nature of biometric authentication, it is impossible to replace it if it is stolen. There is a case in which an actual iris photo is taken and 3d printed so that the eyes work as if they were in front of the camera. As such, there is a possibility of iris leakage through high-definition images and photos. In this paper, we propose to improve iris masking performa nce by supplementing iris region masking research based on existing blurring techniques. Based on the results deriv ed in this study, it is expected that it can be used for the security of video conference programs and electronic devi ces.

Key words : dnn, blurring, iris, masking, security

접수일(2022년 9월 29일), 게재확정일(2022년 10월 25일)

* 상명대학교 정보보안공학과 학부생(주저자, 공동저자)

** 상명대학교 정보보안공학과 부교수 (교신저자)

1. 서 론

기술이 발전하면서 날이 갈수록 스마트폰, 노트북 등 생활 속의 영상 촬영 장비의 해상도는 높아지고 있다. 이러한 장비의 성능 향상으로 코로나-19 팬데믹 현상에서도 생생한 화질로 업무의 차질을 줄여주고 있다. 하지만 고화질, 고해상도의 섬세하고 선명함의 이점을 이용해 생체 정보를 탈취하여 악용하는 사례가 생겨나고 있다. 생체 정보를 보안하는 시스템의 필요성이 대두됨에 따라 openCV의 Haar Cascades 모델을 활용해 홍채 영역에 자동으로 블러링 하는 방법을 제안한 연구가 진행되었다[1]. 하지만 이 연구는 정면 얼굴을 인식할 때는 홍채 블러링이 처리되는 성능을 보이지만 측면 얼굴을 인식에 있어서는 인식이 불가하여 홍채 블러링 처리가 되지 않는 문제점이 생긴다. 또한 성능에서도 평균 90%의 검출률을 보이며 이런 문제점으로 인해 측면에서나 정면 얼굴 미검출 시에 홍채가 탈취당할 경우에는 홍채 블러링은 의미가 없어지게 된다. 이외에도 CNN을 적용한 홍채 인식에 관한 연구도 존재한다[2]. 본 논문에서는 기존 연구에서의 홍채 블러링 처리의 성능을 개선한다.

2. 관련 연구

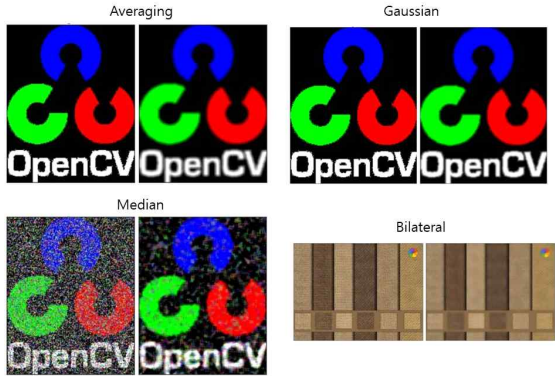
2.1 안면 정보 인식

Method category	Sample algorithms: year first appeared in the literature
Local, holistic, and hybrid	Principal component analysis (Eigenfaces): 1991 Modular Eigenfaces: 1994 Linear discriminant analysis (Fisherfaces): 1997 Independent component analysis (ICA): 2002 Local binary pattern (LBP): 2006 Scale-invariant feature transform (SIFT): 2006 Speeded-up robust features (SURF): 2009 Learning-based descriptor (LBD): 2010
Appearance- and model-based	3D morphable model: 1999 Active appearance model (AAM): 2000 Eigen light field: 2004 Associate–predict model (APM): 2011
Geometry- and template-based	Dynamic link architecture (DLA): 1993 Elastic bunch-graph matching (EBGM): 1997 Trace transform (TT): 2003 Kernel methods: 2002 Simulated annealing for 3D face recognition: 2009
Template-matching, statistical, and neural networks	Probabilistic decision-based neural network (PDBNN): 1997 Genetic algorithm–evolutionary pursuit (EP): 1998 Wavelet packet analysis (WPA): 2000 Sparse representation (SR): 2009 Partial least squares (PLS): 2013 Hybrid deep learning (HDL): 2013 Discriminant face descriptor: 2014 DeepFace deep neural network: 2014 Deep hidden identity features (DeepID): 2014 FaceNet embedding: 2015

(그림 1) Types of face recognition methods and sample algorithms

안면 인식 방법 및 알고리즘 유형은 (그림 1)과 같다. 안면 인식 모델은 대표적으로 Haar Cascade, DNN, MTCNN 등이 있다. 기존 연구에서 사용된 Haar Cascade는 간단한 아키텍처로 cpu에서 거의 실시간으로 동작한다는 장점이 있지만 잘못된 예측을 많이 제공하며 비 정면 이미지에서 작동을 안 한다는 한계점이 있다. MTCNN은 다양한 얼굴 방향에서 작동하며 매우 쉬운 교육 과정을 거치는 장점이 있지만 cpu에서 매우 느려서 영상처리에 적합하지 않다는 단점이 있다[3]. DNN은 안면 정보 인식에서의 방법 중에 가장 정확하다는 장점과 다양한 얼굴 방향에서 작동한다는 점에서 기존 연구의 한계점인 안면 인식 후 블러 처리율을 높이고 측면에서의 블러링 처리까지 보완할 수 있다고 판단되어 DNN 모델을 채택했다.

2.2 블러링 기법



(그림 2) Image Blurring

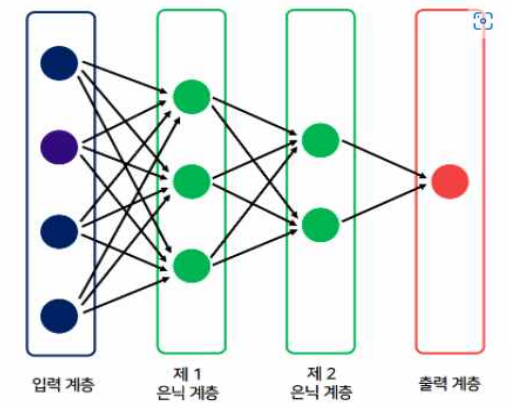
(그림 2)는 블러링 기법을 적용한 이미지의 모습이다. 블러링은 불분명한 사진처럼 사진이나 영상을 선명하지 않게 하는 필터링 기법으로 openCV에서는 대표적으로 Averaging, Median Blurring, Bilateral Filtering, Gaussian Blurring의 연산들을 지원한다. Averaging은 정규화된 상자 필터로 이미지를 합성곱 하는 방식으로 진행된다. 단순히 커널 부분 아래 전 픽셀의 평균을 취하여 중심 요소를 대체한다. Median Blurring은 커널 부분 아래 모든 픽셀의 중앙값을 취하며 중심 요소는 이 중앙값으로 대체되는 방식이다

[4]. Bilateral Filtering은 다른 블러링과 같이 노이즈를 제거하지만 이미지 내의 edge를 살리는 방식으로 진행된다. 이러한 부수적인 기능으로 인해 필터나 이미지의 크기가 커지면 처리가 조금 느려지는 단점이 있다. Gaussian Blurring은 중심 픽셀에서 멀어질수록 영향을 적게 받게 되어 자연스럽게 만들어 주는 효과가 있다. 기존의 연구와는 다르게 블러링 처리되는 영상에서의 이질감을 줄이기 위해 Gaussian Blurring 모델을 채택했다.

3. 제안하는 홍채 마스킹 방법

3.1 Deep Neural Network(DNN)

DNN은 딥러닝의 알고리즘의 하나로 입력 계층과 출력 계층 간에 다수의 은닉 계층들로 이루어진 인공 신경망이다[5]. DNN은 보편적인 인공 신경망과 매한가지로 난해한 비선형 관계들을 모각할 수 있다. 예시로 들어, 물체 분별 모델을 위한 심층 신경망 구성에서는 모든 물체가 영상의 근본적 요건들의 계층적 구조로 묘사될 수 있다[6]. 다른 인공 신경망과는 다르게 (그림 3)과 같이 다수의 은닉층을 활용하여 입력 신호를 더 정교하게 처리하는 게 가능한 알고리즘이다.



(그림 3) DNN 의 구조

openCV에는 얼굴 인식을 위한 라이브러리로 2500개 이상의 최적화된 알고리즘이 있으며 얼굴, 물체, 행동 등 여러 패턴을 인식하는데 최적화가 되어 있다[7]. openCV의 DNN 모듈은 학습된 모델을 활용해 실행한다. 지원하는 딥러닝 프레임워크는 Caffe, TensorFlow, torch, Darknet, ONNX 등이 있다. 본 논문에서는 Caffe 프레임워크의 Caffemodel과 신경망 구성 정보를 갖는 Prototxt 파일을 사용했다. openCV의 DNN 모듈 수행 방법은 <표 2>와 같다.

<표 2> openCV의 DNN 모듈 수행 방법

```
from imutils.video import VideoStream

// 디스크에 직렬화된 모델 로드
import cv2
net = cv2.dnn.readNetFromCaffe(args["prototxt"], args
["model"])

// 비디오 스트림 초기화
vs = VideoStream(src=0).start()
import time
time.sleep(2.0)

while True:
  // 비디오 스트림에서 프레임을 가져와 크기 조정
  frame = vs.read()
  import imutils
  frame = imutils.resize(frame, width=400)

  // 프레임 수치를 잡고 blob으로 변환
  (h, w) = frame.shape[:2]
  blob = cv2.dnn.blobFromImage(cv2.resize(frame, (30
0, 300)), 1.0, (300, 300), (104.0, 177.0, 123.0))

  // 네트워크를 통해 blob을 전달하고 탐지
  net.setInput(blob)
  dets = net.forward()

  for i in range(0, det.shape[2]):
    con = det[0, 0, i, 2]

    // 필터링 기준
    if con < args["con"]:
      continue

    // 경계 상자의 (x, y) - 좌표를 계산
    box = det[0, 0, i, 3:7] * numpy.array([w, h, w, h])
    (sp_x, sp_y, ep_x, ep_y) = box.astype("int")

    // 얼굴의 경계 상자 표시
    txt = "{:.3f}%".format(con * 100)
    y = sp_y - 10 if sp_y - 10 > 10 else sp_y + 10
    cv2.rectangle(frame, (sp_x, sp_y), (ep_x, ep_y),
```

```
(0, 0, 255), 2)
    cv2.putText(frame, txt, (sp_x, y), cv2.FONT_HE
RSHEY_COMPLEX, 0.46, (0, 0, 255), 2)

  // 출력 프레임
  cv2.imshow("Frame", frame)
  inp = cv2.waitKey(1) & 0xFF

  if inp == ord("q"):
    break

cv2.destroyAllWindows()
vs.stop()
```

3.2 Gaussian Blurring

Gaussian Blurring은 입력 영상과 가우시안 필터를 계산하여 출력 영상을 만들어 낸다. 가우시안 필터는 가우시안 분포를 영상 처리에 적용하여 확률분포, 정규분포를 통해 이미지에 생성된 잡음을 제거하기 위한 필터이다[8,9]. 가우시안 분포는 다음 식 (1)과 같다.

$$f(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \quad (1)$$

σ는 표준편차로서 파라미터로 사용한다. 표준편차의 값이 늘어날수록 함수가 더 크게 퍼지는 모습이 되며 더 큰 블러링 효과를 갖는다. 전 픽셀에 동일한 가중치를 부여했던 단순 블러링과 다르게 가우시안 블러링은 가운데에 있는 픽셀에 큰 가중치를 부여한다. 입력 영상의 모서리(Edge)에 있을 때, 단순 블러링은 모서리를 포함해서 전반적으로 블러링이 되지만 가우시안 블러링은 모서리에 남아있는 상태에서 출력 영상이 만들어진다. 따라서 가우시안 블러링은 모서리를 유지하면서 잡음을 제거하는 데 사용된다[9].

openCV의 GaussianBlur 함수를 사용 시 파라미터 설정을 해야 하는데, <표 3>과 같이 파라미터에는 src, dst, ksize, sigmaX가 포함된다. GaussianBlur 함수 수행 방법은 <표 4>와 같다.

<표 3> GaussianBlur 함수 파라미터 설명

파라미터	설명
src	소스(입력 이미지)를 나타내는 매트 개체
dst	대상(출력 이미지)를 나타내는 매트 개체
ksize	커널의 크기를 나타내는 크기 개체
sigmaX	X 방향의 가우스 커널 표준 편차를 나타내는 이중 유형의 변수

<표 4> GaussianBlur 함수 수행 방법

```
import org.opencv.imgproc.Imgproc;
import org.opencv.core.Core;
import org.opencv.core.Size;
import org.opencv.imgcodecs.Imgcodecs;
import org.opencv.core.Mat;

public class GaussianTest {
   public static void main(String args[]) {
      // OpenCV 코어 라이브러리 로드
      System.loadLibrary(Core.NATIVE_LIBRARY_N
AME);

      // 파일에서 이미지를 읽고 Matrix 개체에 저장
      String file ="C:/파일 경로";
      Mat src = Imgcodecs.imread(file);

      // 결과를 저장할 빈 행렬 만들기
      Mat dst = new Mat();

      // 이미지에 가우스 블러 적용
      Imgproc.GaussianBlur(src, dst, new Size(*, *),
*);

      // 이미지 쓰기
      Imgcodecs.imwrite("E:/파일 경로", dst);
      System.out.println("Image Processed");
   }
}
```

4. 실험 및 결과

기존에 진행되었던 블러링 기법 기반의 홍채 영역 마스킹 연구는 Haar Cascades 모델을 사용하여 얼굴의 정면에서는 안면의 높은 인식률을 보이지만 얼굴을 측면으로 돌리게 된다면 인식률이 저조해진다는 뚜렷한 단점을 지니고 있다. 또한 안면 인식률이 평균 89%로 나타난 모델을 사용하여 취약한 모습을 보

인다. 측면 인식이 되지 않는다면 홍채 인식도 같이 인식하지 못한다. 이러한 이유로 Haar Cascades보다 안면 인식률이 높은 openCV의 DNN 모듈을 이용하고 (그림 4)와 같이 openCV의 DNN 모듈의 측면 얼굴 인식도 이용하여 기존의 한계점인 안면 인식률과 측면 인식 불가능이라는 한계점을 개선했다.

(그림 4) DNN 모델을 이용한 측면 인식

5. 결 론

기술의 발전으로 고화질의 영상과 사진을 쉽게 구할 수 있다. 또한 코로나 19로 증가한 원격 수업 등이 많아지면서 생체 정보가 노출될 수 있다. 홍채는 동공 주위에 있는 조직으로 유아기에 완전한 모양을 갖춘다. 홍채 패턴은 사람마다 다 다르며 오랫동안 변하지 않는다는 특성을 갖고 있기 때문에 탈취를 당하게 되면 대체가 불가능하다[10]. 홍채의 노출을 최소화하기 위해 본 논문에서는 DNN 딥러닝 알고리즘을 활용한 안면 인식 방법을 제안하였다. 제안하는 방법은 정면에서뿐만 아니라 측면에서도 높은 인식률을 갖는다. 또한 Gaussian Blurring 기법을 적용해 이질감이 적은 블러링 결과를 얻을 수 있었다. 이는 영상과 사진에서의 개인 생체 보안 문제를 개선할 수 있을 것으로 기대된다.

참고문헌

[1] 이기성, 김수형, “블러링기법 기반의 홍채영역 마스킹 방법”, 스마트미디어저널, 제11권, 제2호, pp. 25-30, 2022.

[2] 이민범, 박강령, “딥러닝 기반 홍채 인식 알고리즘에 관한 연구”, 한국통신학회 학술대회논문집, pp. 1458-1460, 2018.

[3] https://velog.io/@easttwave/Deep-Learning-얼굴-인식-모델-비교-조사, velog.

[4] https://docs.opencv.org/4.x/d4/d13/tutorial_py_filtering.html, opencv.

[5] Y. Bengio, A. Courville, and P. Vincent., "Representation Learning: A Review and New Perspectives," IEEE Trans. PAMI, special issue Learning Deep Architectures, 2013.

[6] Szegedy, Christian, Alexander Toshev, and Dumitru Erhan. "Deep neural networks for object detection." Advances in Neural Information Processing Systems. 2013.

[7] 김수환, 공태민, 권여진, 임의연, 최동연, “얼굴 인식을 이용한 안전한 학습 도우미 프로그램”, 한국정보과학회 학술발표논문집, pp.1888-1890, 2022.

[8] 원유현, 김진성, 박병찬, 김영모, 김석윤 “360도 실감형 미디어에서의 고해상도에 대한 특징을 활용한 특징점 추출 방법”, 한국컴퓨터정보학회논문지, 제24권, 제1호, pp.85-92, 2019.

[9] 김강섭, 홍영기, 김현종, 김국환, 김경철, 이명훈, “영상처리를 이용한 토마토 생육량 측정 시스템”, 한국통신학회논문지, 제45권, 제8호, pp. 1460-1471, 2020.

[10] 이영원, 박강령, “홍채인식 시스템에 관한 기존 연구 분석”, 대한전자공학회 학술대회, pp. 610-612, 2017.

〔저자소개〕

백 승 민 (Seungmin Baek)

현재 상명대학교 정보보안공학과 재학

email : 201721359@sangmyung.kr

최 영 해 (Younghae Choi)

현재 상명대학교 정보보안공학과 재학

email : 201721382@sangmyung.kr

홍 찬 우 (Chanwoo Hong)

현재 상명대학교 정보보안공학과 재학

email : 201721384@sangmyung.kr

박 원 형 (WonHyung Park)
2002년 서울과학기술대 산업정보시스템 학사
2005년 서울과학기술대 정보산업공학과 석사
2009년 경기대학교 정보보호학 박사
2015년 성균관대학교 컴퓨터교육학 박사수료
2022년 상명대학교 정보보안공학과 부교수
email : whpark@smu.ac.kr