International Journal of Advanced Culture Technology Vol.7 No.4 289-294 (2019)
DOI 10.17703/IJACT.2019.7.4.289

IJACT 19-12-35

An Automatic Face Hiding System based on the Deep Learning Technology

Hyeon-Dham Yoon[1], Seong-Yong Ohm[2*]

[1] Undergraduated Student, Major of Visual Communication Design, Seoul Women's University
[2]Professor, Department of Software Convergence, Seoul Women's University
E-Mail : dadi7259@swu.ac.kr, osy@swu.ac.kr (Corresponding Author)

Abstract

As social network service platforms grow and one-person media market expands, people upload their own photos and/or videos through multiple open platforms. However, it can be illegal to upload the digital contents containing the faces of others on the public sites without their permission. Therefore, many people are spending much time and effort in editing such digital contents so that the faces of others should not be exposed to the public.

In this paper, we propose an automatic face hiding system called 'autoblur', which detects all the unregistered faces and mosaic them automatically. The system has been implemented using the GitHub MIT open-source 'Face Recognition' which is based on deep learning technology. In this system, two dozens of face images of the user are taken from different angles to register his/her own face. Once the face of the user is learned and registered, the system detects all the other faces for the given photo or video and then blurs them out. Our experiments show that it produces quick and correct results for the sample photos.

Keywords: *Open-source, Deep Learning, Image Processing, Face Recognition*

1. INTRODUCTION

As the SNS platform grows and the one-person media market expands, non-professional broadcasters are sharing their daily lives or special experiences through various platforms such as YouTube, Facebook, and Instagram [1]. At this time, it often happens that some other persons are included in the pictures together. Exposing the digital contents containing other people's faces to the public without their permission is a portrait infringement [2], but most of them upload such the images as it is. We surveyed 100 people, who uploaded images to their SNS more than once a week, to identify the awareness of portrait rights. According to the survey, 88% of the respondents uploads pictures without editing the faces of others shoot. In addition, as shown in Figure 1, more than half of the respondents do not edit the faces of others due to difficulty of manual editing.

In this paper, we propose a new smartphone application called *autoblur* [3], which automates the existing method of manually editing only the exposed faces of others. This system has been implemented using *github* MIT open-source '*Face Recognition*' [4] based on deep learning technology [5]. The main steps of the system are <Face Registration> and <Auto Image Conversion>. In the <Face Registration> process, the user's face is photographed at various angles using a mobile phone camera, and the collected face images are sent to the server for use as training data. Once the <Face Registration> process is completed, the user can select an image (test data) to edit from the camera or the gallery. In the <Auto Image Conversion> process, all faces in the

Manuscript received: October 10, 2019/ revised: October 24, 2019 / Accepted: November 10, 2019
Corresponding Author: osy@swu.ac.kr
Tel: +82-10-6391-1116, Fax: +82-2-970-5981
Professor, Department of Software Convergence, Seoul Women's University, Korea

image are first detected and classified as learning data (i.e. registered face) and non-learning data (i.e. unregistered faces), and then only the non-learning data (i.e. unregistered faces) are hidden automatically.

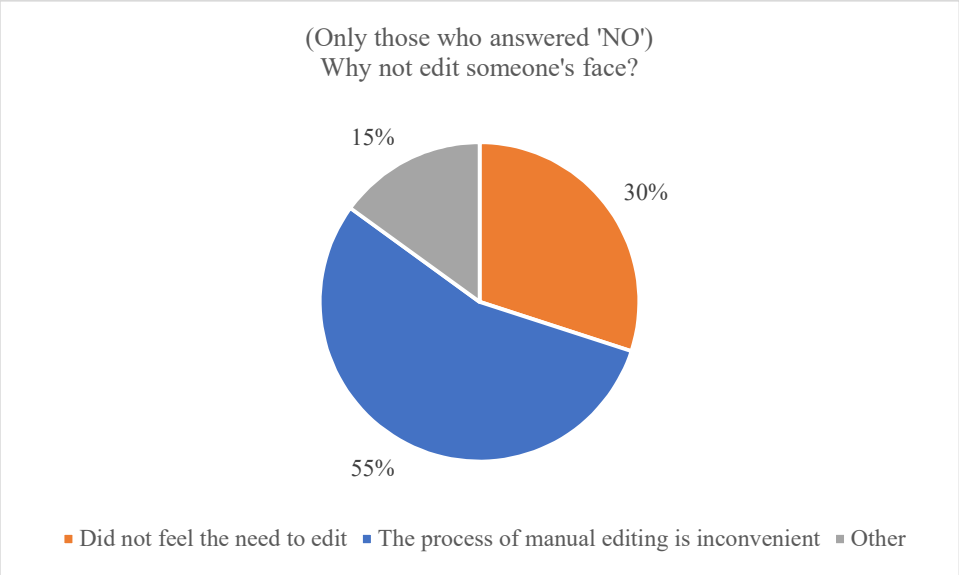


Figure 1. Survey results for people uploading images without editing their faces on SNS

2. RELATED WORK

2.1 Hello Mirror Project using *Openface*

'Openface'[6] is an open-source library based on deep learning that is the basis of 'Face Recognition' used in this study. *Openface* uses the *FaceNet* system [7] with high accuracy in face recognition. The Hello Mirror project [8] uses the *Openface* library to find Google API information and output mail or calendar information in real-time. When a new face appears in front of the mirror, the system extracts and compares the face with the registered one. Log in if the person is the registered one, and log out if not. In the face recognition library, faces in pictures / videos are also recognized. Thus, just having a photo of the registered person will give the user access to that information. Therefore, there are limitations to using these libraries for security purposes.

2.2 Application Zao using *DeepFake*

'*DeepFake*' is a compound word of deep learning and Fake. It refers to a video editing technology that synthesizes a human face or a specific part. *DeepFake* replaces the original eye, nose, and mouth with edit data. Zao [9], an application launched in China, uses this technology to provide a service for easily and naturally synthesizing images of characters in movies or dramas. When data compiled using *DeepFake* is uploaded to SNS or public site, copyright infringement and user information misuse can occur. It can also be used to create fake news by synthesizing people in the video with others. It is also easy to find examples of illegal dissemination of the artist's face and porn using this technology [10].

3. OUR METHOD

The main functions of this system are <Face Registration> and <Auto Image Conversion>. The <Face Registration> is a process of taking a plurality of photographs of a user and learning them, and the <Auto Image Synthesis> is a process of automatically blurring unregistered faces for the given photo. Figure 2 shows the process of the system.

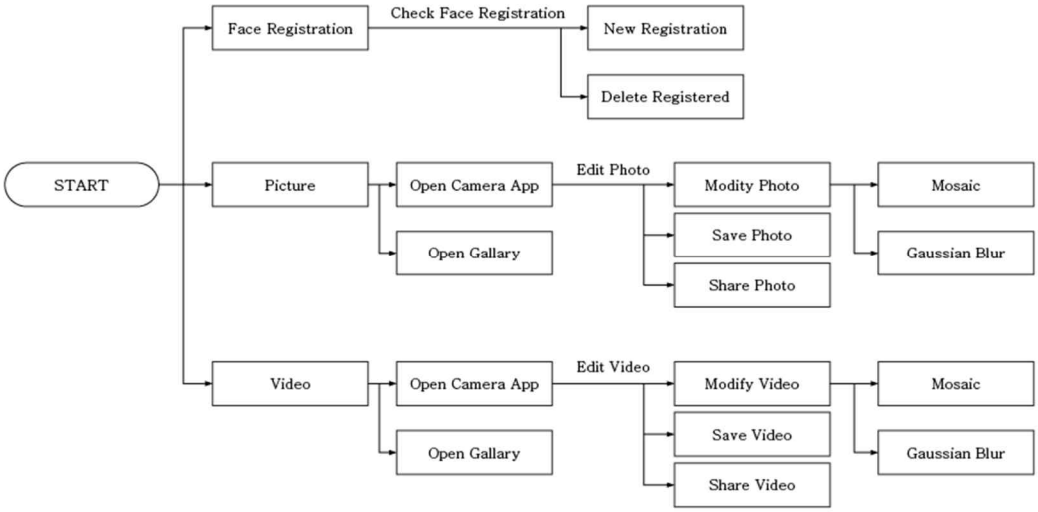


Figure 2. Process of '*autoblur*' system

3.1 Face Registration (Training Data Learning)

In the <Face Registration> process, the user's face images are photographed at various angles and then are uploaded to the server. Once the user pictures are uploaded to the server, the server starts to train about the user's face based on the pictures. Figure 3 shows the process of registering a user's picture. In this paper, when the user takes 20 face images of various angles in the <Face Registration> step, the data is automatically uploaded to the server for data learning.

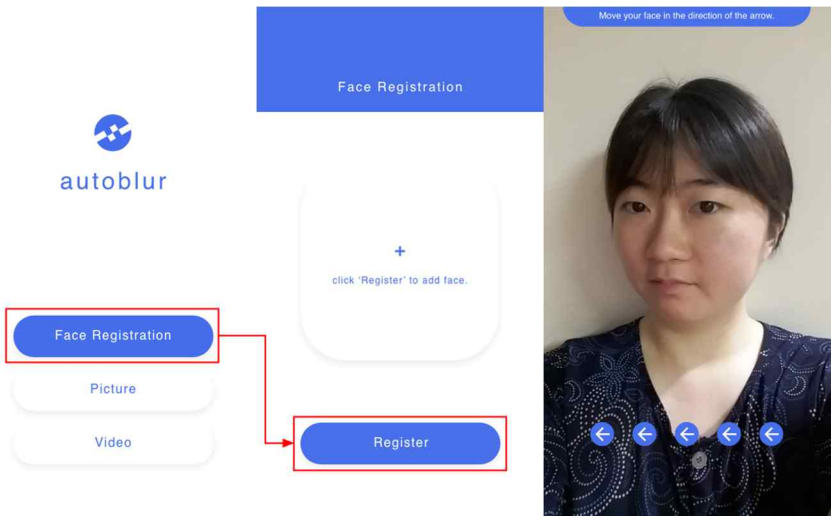


Figure 3. Face registration step for '*autoblur*' system

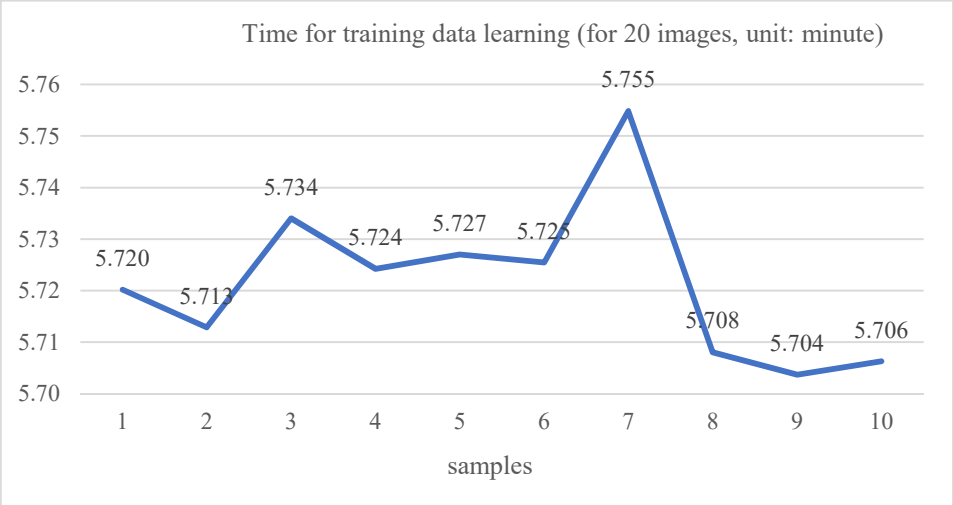


Figure 4. Time measurement result for training data learning

3.2 Automatic Image Conversion (Editing Test Data)

Once training data learning is completed, automatic transformation can be applied to any images. The user can take a photo from the smartphone's camera or select from the gallery to apply automatic conversion. This process first recognizes all the face areas in the image. Next, the recognized faces are classified into learned faces and unregistered faces. Unregistered face areas are automatically edited using a mosaic or Gaussian blur process. The Gaussian filter of *OpenCV* is used for image editing using Gaussian blur. Unlike Gaussian blur, the mosaic does not have a filter provided by *OpenCV*, so the photo is resized using a scaling method.

Figure 5 shows the original photo and the automatically converted photo. In the picture, the unregistered faces are automatically mosaic-treated.

Figure 5. Example of original and auto converted image

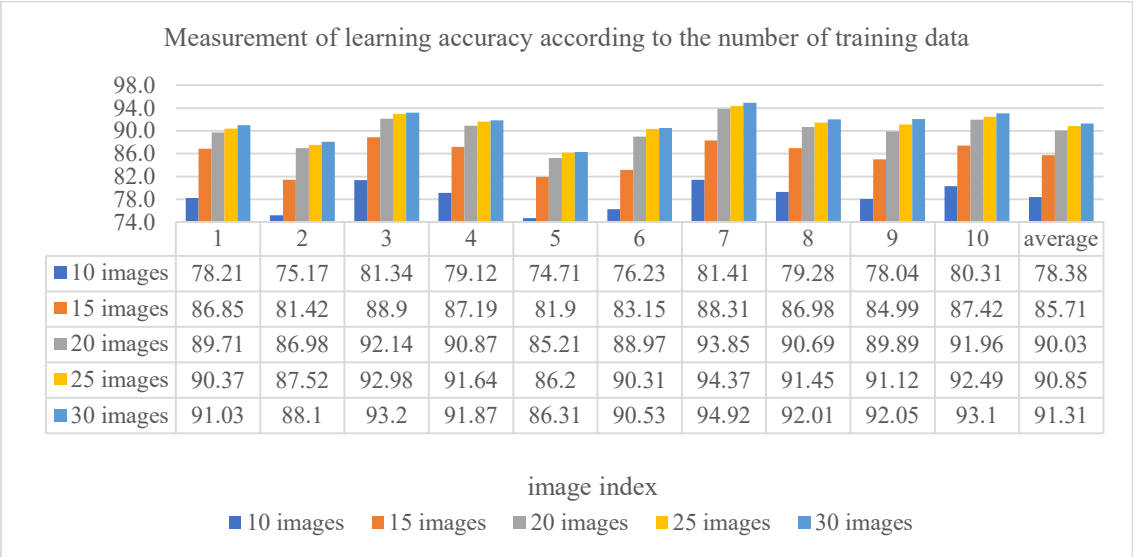
4. EXPERIMENTAL RESULTS

In order to evaluate the performance of this system, we experimented with 10 sample examples. The server used in this system is an *AWS* [11] *EC2* instance with an environment of *Ubuntu* [12] Server 18.04 LTS (HVM), a <t2.small> type with one vCPU and 2 GiB memory. Due to the performance and capacity limitations of the server, for high-quality photos, the original pictures are compressed to a width and height of 680px or less.

Figure 6 shows the result of measuring the accuracy of applying the trained face to 10 real sample pictures,

and Figure 7 shows the time and accuracy changes for training data learning as the number of training data increases. In Figure 6, when the number of training data is 20 or more, the accuracy is more than 90%. On the other hand, as the number of training data increases, the learning time increases constantly, but the increase in accuracy is not so large as shown in Figure 7.

.



	1	2	3	4	5	6	7	8	9	10	average
10 images	78.21	75.17	81.34	79.12	74.71	76.23	81.41	79.28	78.04	80.31	78.38
15 images	86.85	81.42	88.9	87.19	81.9	83.15	88.31	86.98	84.99	87.42	85.71
20 images	89.71	86.98	92.14	90.87	85.21	88.97	93.85	90.69	89.89	91.96	90.03
25 images	90.37	87.52	92.98	91.64	86.2	90.31	94.37	91.45	91.12	92.49	90.85
30 images	91.03	88.1	93.2	91.87	86.31	90.53	94.92	92.01	92.05	93.1	91.31

Figure 6. Accuracy measurement result of 10 sample images

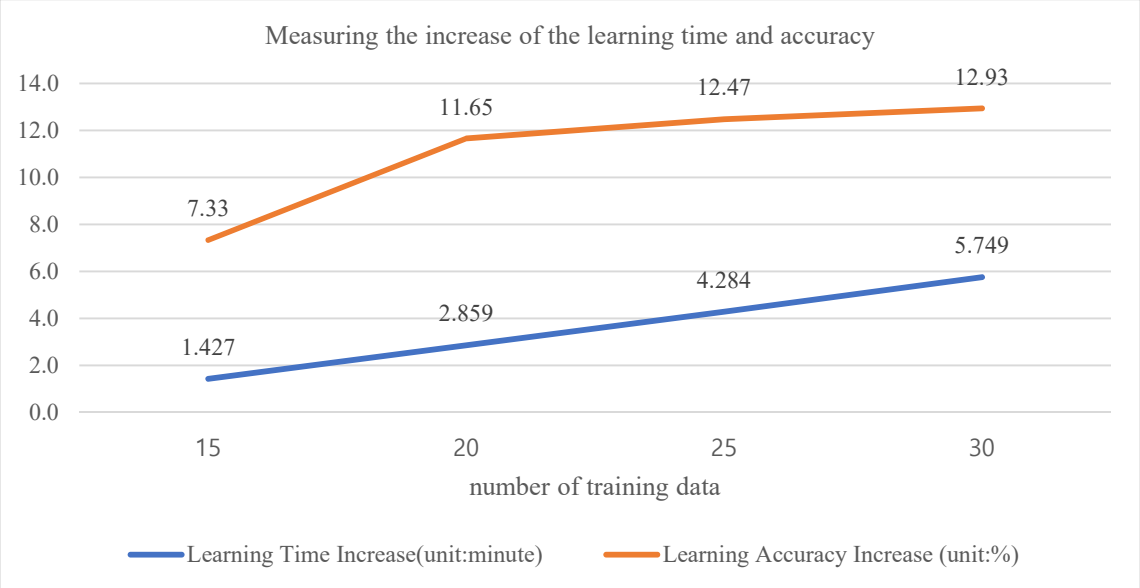


Figure 7. Increment of training data learning time and accuracy

As a result, at the <Face Registration> step, as the number of training data increases, time spent for taking user's photos, server data transfer time, and training data learning times increases at a constant rate, but the accuracy increase is not. Therefore, we comprehensively considered these points and set the number of pictures used for user face registration to 20.

5. CONCLUSION

In this paper, we propose a system to automate efficiently the image editing process to protect the portrait rights of others who do not have the consent of the shooting. Storing images and learning data inside the mobile phone is almost impossible, so we use a server linked with the Android system to improve the system speed.

The accuracy and processing speed considered in classifying registered faces and unregistered ones in test samples are inversely proportional to each other. We found, however, that the increase of the accuracy sometimes starts to decrease gradually, as the number of images used for learning increases. In this paper, we choose the most appropriate value by repeated experiments. In other words, in the environment of this paper, the best results in terms of speed and accuracy are determined when 20 training data has been learned.

In this system, among the types provided by AWS EC2 instance, <t2.small> type is used. It is the minimum environment where we can install *dlib* package for face detection [13]. To speed up learning while keeping the amount of training data fixed, we need to consider another instance type that provides better performance.

ACKNOWLEDGEMENT

This work was supported by a research grant from Seoul Women's University (2019).

REFERENCES

[1] Kim Yoon-Hwa, "SNS(Social Network Service) Usage Trends and Behavioral Analysis", *KISDI STAT Report 19-10,* p.7, May 2019.
[2] Supreme Court Decision 2004Da16280 of South Korea Sentencing, October 2006.
[3] Yoon Hyeon-Dham, Lee Ju-Min, Lee Na-Kyung, Ohm Seong-Yong, "Autoblur : An Image Editing Application Using Deep Learning-Based Face Recognition Technology", *KMMS(Korea Multimedia society)*, Vol.22, Issue 2, 2019.
[4] "Face recognition with OpenCV, Python, and deep learning", *https://www.pyimagesearch.com/2018/06/18/face-recognition-with-OpenCV-python-and-deep-learning*.
[5] "Deep learning", Wikipedia, *https://en.wikipedia.org/wiki/Deep_learning.*
[6] Kim Dae-Hwan, "*Various method for landmark detection and application to face recognition*", Degree dissertation – Graduate School of Hongik University, 2007.
[7] Florian Schroff, Dmitry Kalenichenko, James Philbin, "FaceNet: A Unified Embedding for Face Recognition and Clustering", *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 815-823, 2015.
[8] Yoo Kyung-Hwa, "*A Study on the upgrade and utilization of the Deep Learning based Face Recognition Library*", Graduate School of Welfare and Management, Namseoul University, 2018.
[9] "Change face" app blitzing China… It's an abuse in the U.S. presidential election?", *http://news.naver.com/main/read.nhn?mode=LSD&mid=sec&sid1=001&oid=469&aid=0000420634.*
[10] "Entertainer's Face Synthesis: porn deal using AI in China", *https://www.yna.co.kr/view/AKR20190719090200083.*
[11] Amazon Web Service General Reference*, https://docs.aws.amazon.com/ko_kr/general/latest/gr/Welcome.html.*
[12] The best Linux platform for modern cloud and IoT development, *https://ubuntu.com*.
[13] How to Install a Face Recognition Model at the Edge with AWS IoT Greengrass*, https://aws.amazon.com/ko/blogs/iot/how-to-install-a-face-recognition-model-at-the-edge-with-aws-iot-greengrass.*