Korean J Neurotrauma. 2019 Oct;15(2):88-94
https://doi.org/10.13004/kjnt.2019.15.e17
pISSN 2234-8999·eISSN 2288-2243

KJNT

Clinical Article

Application of Deep Learning System into the Development of Communication Device for Quadriplegic Patient

Jung Hwan Lee [1], **Taewoo Kang** [2], **Byung Kwan Choi** [1], **In Ho Han** [1], **Byung Chul Kim** [1], **and Jung Hoon Ro** [3]

[1]Department of Neurosurgery, Pusan National University Hospital, Busan, Korea
[2]Busan Cancer Center, Pusan National University Hospital, Busan, Korea
[3]Department of Biomedical Engineering, Pusan National University Hospital, Busan, Korea

Received: May 8, 2019
Revised: Jun 19, 2019
Accepted: Jul 12, 2019

Address for correspondence:
Taewoo Kang
Busan Cancer Center, Pusan National University Hospital, 179 Gudeok-ro, Seo-gu, Busan 49241, Korea.
E-mail: taewoo.d.kang@gmail.com

ORCID iDs
Jung Hwan Lee
https://orcid.org/0000-0002-1393-7105
Taewoo Kang
https://orcid.org/0000-0002-6279-0904
Byung Kwan Choi
https://orcid.org/0000-0001-5244-4592
In Ho Han
https://orcid.org/0000-0001-7193-6533
Byung Chul Kim
https://orcid.org/0000-0002-0439-1382
Jung Hoon Ro
https://orcid.org/0000-0002-5203-5956

Conflict of Interest
The authors have no financial conflicts of interest.

ABSTRACT

Objective: In general, quadriplegic patients use their voices to call the caregiver. However, severe quadriplegic patients are in a state of tracheostomy, and cannot generate a voice. These patients require other communication tools to call caregivers. Recently, monitoring of eye status using artificial intelligence (AI) has been widely used in various fields. We made eye status monitoring system using deep learning, and developed a communication system for quadriplegic patients can call the caregiver.
Methods: The communication system consists of 3 programs. The first program was developed for automatic capturing of eye images from the face using a webcam. It continuously captured and stored 15 eye images per second. Secondly, the captured eye images were evaluated for open or closed status by deep learning, which is a type of AI. Google TensorFlow was used as a machine learning tool or library for convolutional neural network. A total of 18,000 images were used to train deep learning system. Finally, the program was developed to utter a sound when the left eye was closed for 3 seconds.
Results: The test accuracy of eye status was 98.7%. In practice, when the quadriplegic patient looked at the webcam and closed his left eye for 3 seconds, the sound for calling a caregiver was generated.
Conclusion: Our eye status detection software using AI is very accurate, and the calling system for the quadriplegic patient was satisfactory.

Keywords: Eye; Artificial intelligence; Unsupervised machine learning; Quadriplegia; Caregiver

INTRODUCTION

Detection of the open or closed status of the eye is very important in many areas, such as human-device interface or driver drowsiness detection system.[13)] Monitoring of the patient's eye opening is also important in the medical field. Eye blinking can create an artifact on electroencephalogram (EEG) or ocular magnetic resonance imaging (MRI).[1,15)] As well as, blinking is a useful communication channel for severely disabled patients. In general, quadriplegic patients use their voices to find the caregiver. However, aggressive quadriplegic patients are mostly in a state of tracheostomy, because of prolonged ventilator care or difficult excretion of sputum. The patient in tracheostomy state cannot generate speech. These patients require other communication methods to call caregivers.

Recently, there has been a remarkable advance in artificial intelligence (AI).[12] Especially, image recognition using AI with deep learning algorithms (DLAs) has been increasingly applied in the various medical fields.[7] These fields include radiographic images such as X-ray or MRI, histological classifications, and endoscopic findings.[4,5,7,12] To monitor the patient's eye status, some studies have been reported to use these DLA.[8] We created a new eye status monitoring system using a DLA. Further, we applied this algorithm to develop a communication system for quadriplegic patients to call the caregiver.

MATERIALS AND METHODS

Software configuration

Eye detection algorithm and collection of eye images

We used open source computer vision (CV) platform to detect the face and eyes using a Haar cascade. For .NET environment, Emgu CV was adopted (version 3.0.0). When the program was run, it recognizes the face and extracts both eye images (**FIGURE 1**). It continuously captures and stores 15 eye images per second through the webcam. The size of the images we collected was 28×28 pixels, for a total of 784 pixels (**FIGURE 2**). Eye images from 30 normal people were used in this study. People wearing glasses usually removed their glasses. Then, they looked at the webcam in front. While looking into a webcam, they were advised to open

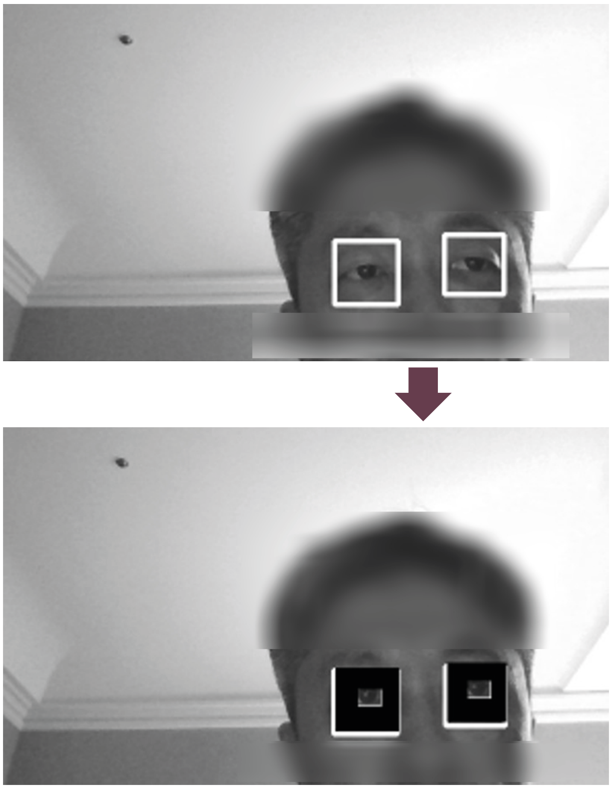


FIGURE 1. Both eyes are captured by the webcam.

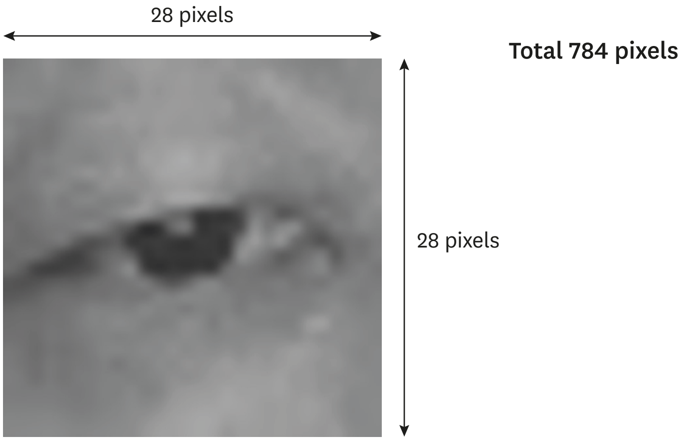


FIGURE 2. Captured eye image. The size is 28×28 pixels, a total of 784 pixels.

and close their eyes. Their images were captured for 20 seconds. The closed eye images and the open eye images stored separately. A total of 18,000 images were collected.

DLA: training eye images with the convolutional neural network (CNN)

Google Tensorflow (version 0.12) was used as the machine learning library for CNN. The CNN system was used to training images (**FIGURE 3**). We used 3 optimizers as follows: Gradient Descent optimizer, RMSprop optimizer, and Adam optimizer. The parameters used in this study were as follows: the batch size=128, learning rate=0.001, and training session=1,000. A total of 18,000 images from 30 volunteers were used to train DLA, and 1,000 images were given as a test set.

Implementation of patient communication

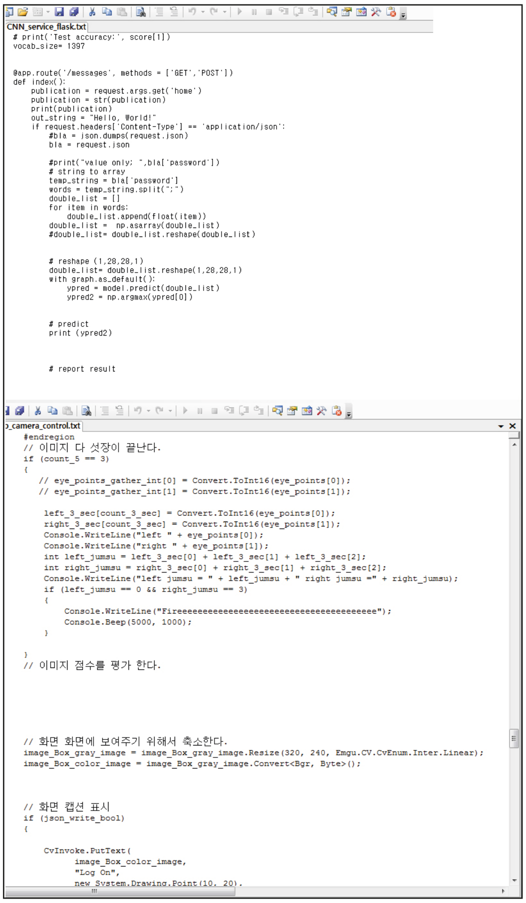
We used C# platform to implement patient communicator program in the .NET environment. Five eye images of both eyes were obtained in a second. Each eye image was sent to the AI server. The AI server is a flask server with a trained model that runs on the same computer. Eye status was reviewed via our DLA. We made a program to determine that the eye was closed for a second if 3 or more eye images were classified as closing. The program generated a sound when it was confirmed that the left eye was closed continuously for 3 seconds once the right eye was open.

Specification of computer

In this study, a computer system with the following specifications was used: A 2.5 GHz Intel® (Santa Clara, CA, USA) Core™ i7-6500U CPU, 8 GB RAM, and an Intel® (Santa Clara) HD 520 graphics card. The computer used in this study was a common office laptop. The programs were executed on a 64-bit Windows 10 operating system.

Practical applications in quadriplegic patients

Under the Institutional Review Board (IRB) approval of Pusan National University Hospital, the software was applied to 5 quadriplegic patients. IRB approval number was 1802-009-063. After receiving informed consents, patients were asked to use the calling system 30 times each. Then, it was confirmed whether it did not work or malfunctioned.



CNN_service_flask.txt

```
# print('Test accuracy:', score[1])
vocab_size= 1397


@app.route('/messages', methods = ['GET','POST'])
def index():
    publication = request.args.get('home')
    publication = str(publication)
    print(publication)
    out_string = "Hello, World!"
    if request.headers['Content-Type'] == 'application/json':
        #bla = json.dumps(request.json)
        bla = request.json

        #print("value only: ",bla['password'])
        # string to array
        temp_string = bla['password']
        words = temp_string.split(";")
        double_list = []
        for item in words:
            double_list.append(float(item))
        double_list =  np.asarray(double_list)
        #double_list= double_list.reshape(double_list)


        # reshape (1,28,28,1)
        double_list= double_list.reshape(1,28,28,1)
        with graph.as_default():
            ypred = model.predict(double_list)
            ypred2 = np.argmax(ypred[0])


        # predict
        print (ypred2)



        # report result
```

camera_control.txt

```
#endregion
// 이미지 다 섯장이 끝난다.
if (count_5 == 3)
{
   // eye_points_gather_int[0] = Convert.ToInt16(eye_points[0]);
   // eye_points_gather_int[1] = Convert.ToInt16(eye_points[1]);

    left_3_sec[count_3_sec] = Convert.ToInt16(eye_points[0]);
    right_3_sec[count_3_sec] = Convert.ToInt16(eye_points[1]);
    Console.WriteLine("left " + eye_points[0]);
    Console.WriteLine("right " + eye_points[1]);
    int left_jumsu = left_3_sec[0] + left_3_sec[1] + left_3_sec[2];
    int right_jumsu = right_3_sec[0] + right_3_sec[1] + right_3_sec[2];
    Console.WriteLine("left jumsu = " + left_jumsu + " right jumsu =" + right_jumsu);
    if (left_jumsu == 0 && right_jumsu == 3)
    {
        Console.WriteLine("Fireeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee");
        Console.Beep(5000, 1000);
    }

}
// 이미지 점수를 평가 한다.




// 화면 화면에 보여주기 위해서 축소한다.
image_Box_gray_image = image_Box_gray_image.Resize(320, 240, Emgu.CV.CvEnum.Inter.Linear);
image_Box_color_image = image_Box_gray_image.Convert<Bgr, Byte>();



// 화면 캡션 표시
if (json_write_bool)
{

    CvInvoke.PutText(
          image_Box_color_image,
          "Log On",
          new System.Drawing.Point(10, 20),
```

FIGURE 3. Software coded using Google Tensorflow. The upper image is the algorithm related to sending and receiving eye images to the artificial intelligence server. The lower image is the algorithm related to generating a sound when the left eye was closed continuously for 3 seconds once the right eye was open.

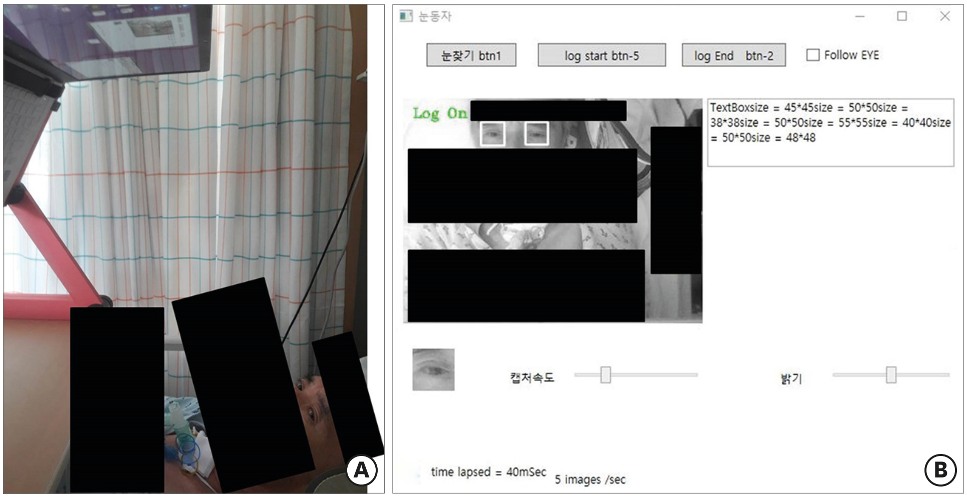
RESULTS

The DLA of Google Tensorflow, trained with 18,000 images, distinguished the open and close status of the test eye images with a probability of by 96.1–98.7%. The results of each of the Tensorflow optimizers are as follows: Gradient Descent optimizer=96.05, RMSProp optimizer=98.1, and Adam optimizer=98.7.

In the absence of a caregiver, the patient viewed the webcam and closed the left eye for 3 seconds, and a sound was generated to call the caregiver (**FIGURE 4**). In a total of 150 attempts, there was no malfunction, thus the success rate was 100%.

DISCUSSION

Methods determining the closed status of an eye are largely divided into non-image-based and image-based technique.[8)] A typical non-image-based technique is EEG.[1)] The advantage of this technique is fast data collection. However, a few disadvantages of this technique include discomfort following attachment of sensors to the body or low accuracy due to noise



TextBoxsize = 45*45size = 50*50size = 38*38size = 50*50size = 55*55size = 40*40size = 50*50size = 48*48

time lapsed = 40mSec 5 images /sec

FIGURE 4. Practical applications in a quadriplegic patient. (A) Actual setting for the quadriplegic patient. (B) Window of the software. Click 'btn 1' button to recognize both eyes. Click 'log start' button to recognize eye status, and call the caregiver. When the left eye was closed for 3 seconds, a sound was generated to call the caregiver. The 'log end' button was used to stop distinguishing eye status.

generated by patient's motion. Thus, many studies have been performed using an image-based technique to overcome these drawbacks.[8)]

Image-based techniques to determine eye opening status were divided into a video-based method and a single-image-based methods. Various video-based methods have yielded high accuracy. However, they were time-consuming and required excessive computation.[8)]

Single-image-based methods were divided into non-training and training techniques. One of the most popular applications of single-image-based methods (non-training techniques) was iris detection.[2)] The main advantage of these techniques was that additional training process was not required. However, the disadvantage was that the performance of these techniques might be negatively influenced if appropriate extraction failed to elicit the requisite information to distinguish eye status from a single image, which might limit the optimization of image-acquisition when extracting features.[8)] On the other hand, the advantage of single-image-based methods (training techniques) was shorter processing time compared with video-based methods and was more accurate than non-training techniques. However, training techniques were difficult to find an optimal feature extraction method.[8)] To resolve this problem, we considered the CNN-based methods. CNN is a type of deep learning that is widely used in the field of image recognition, which facilitated the automatic acquisition of optimal features from training data by both filters and classifiers.[8,9)] Extraction was automatically acquired from training data.[8)] We used TensorFlow in several types of CNN models. TensorFlow is an open source deep learning platform made by Google. It is widely used for both research and commercial applications industrially.[3)]

Our DLA determines whether the eye is open or closed fairly accurately, but it is not perfect. The error rate of the method to confirm eye status using CNN is reported to be 0.24–0.91%.[8)] Therefore, an additional process is required depending on the application. In our study, 5 eye images were captured in one second, and in case 3 or more images were identical, it was decided whether or not to open for 1 second. If 3 or more pictures are recognized incorrectly, the single-second eye status is recognized incorrectly. Since the error rate of eye status detection from a picture was 1.3% in this study, the error rate of eye status detection per second was 0.002% ($0.987^2 \times 0.013^3 \times 10 + 0.987 \times 0.013^4 \times 5 + 0.013^5$). If this process was performed continuously for 3 seconds, the success rate of sound generation was 99.99% $\{(1-0.00002)^3\}$. Thus, it was almost 100% likely to determine that a single eye was closed continuously for 3 seconds.

It would be unnecessary to develop devices simply to determine whether the eyes opened or closed once or twice. However, it is important to confirm continuous eye status in specific areas. For example, monitoring the eye status may be essential for a device that detects drowsiness while driving or when engaged in dangerous work.[13] In the medical area, whether the eyes status may affect the result of the examination. A typical example is EEG.[1] Waves can vary depending on the state of the eye, when the EEG test is performed. Therefore, a few studies were conducted on a device for confirmation of the eye status during the EEG before AI was widely known.[1] Using our methods, the eye status can be determined via the webcam by the time even if the patient lacked a special monitoring device. In severe patients who can move only with the eyes, it is very important to confirm whether the eyes are open or closed. Diseases that cause this condition include complete cervical spinal cord injury, amyotrophic lateral sclerosis, and locked-in-syndrome.[6,10,11] These patients need constant caregivers. However, since caregivers are humans, they cannot constantly watch the patients. Thus, both the patient and caregiver benefit from a device that generated sound to call the caregiver when needed. Therefore, we have designed this application. In our study, the application only produced a sound to call the caregiver.

Recently, commercial eye-tracking computing systems (ETCSs) have been developed and applied to quadriplegic patients.[14] In addition to calling the caregiver, ETCSs can help these patients through various other functions. However, such devices and software are expensive because the number of patients needed is few. The method used in this study cannot perform the eye-tracking, but can be implemented in a common office laptop with the webcam.

This system has some limitations. At first, this software is only implemented in Windows and is not compatible with Android or iOS for iPhone, which are widely used in mobile devices. This software cannot run concurrently with other software programs. Secondly, excessive head movement while running the software prevents this system from recognizing eyes on the face. Further, if the distance between the webcam and the face is too far or close, the software may not be able to recognize the face. And, the state of lighting affects facial and eye recognition. Finally, the number of patients who applied this system was small because the incidence of patients with these conditions is generally very low. Thus, further studies for more patients are needed.

CONCLUSION

Our eye status detection software using AI is very accurate and yields objective assessment. We developed calling system using AI for quadriplegic patients, which are difficult to call a caregiver, and achieved satisfactory results. The system has the potential for key applications in other clinical practice and research areas. Thus, additional studies using AI are needed to improve patient convenience.

ACKNOWLEDGMENTS

This work was supported by Convergence Medicine and Technology Center of Pusan National University Hospital.

REFERENCES

1. Chang WD, Lim JH, Im CH. An unsupervised eye blink artifact detection method for real-time electroencephalogram processing. **Physiol Meas 37**:401-417, 2016
PUBMED | CROSSREF
2. Choi JS, Bang JW, Heo H, Park KR. Evaluation of fear using nonintrusive measurement of multimodal sensors. **Sensors (Basel) 15**:17507-17533, 2015
PUBMED | CROSSREF
3. Dominguez Veiga JJ, O'Reilly M, Whelan D, Caulfield B, Ward TE. Feature-free activity classification of inertial sensor data with machine vision techniques: method, development, and evaluation. **JMIR Mhealth Uhealth 5**:e115, 2017
PUBMED | CROSSREF
4. Esteva A, Kuprel B, Novoa RA, Ko J, Swetter SM, Blau HM, et al. Dermatologist-level classification of skin cancer with deep neural networks. **Nature 542**:115-118, 2017
PUBMED | CROSSREF
5. Feng R, Badgeley M, Mocco J, Oermann EK. Deep learning guided stroke management: a review of clinical applications. **J Neurointerv Surg 10**:358-362, 2018
PUBMED
6. Harrop JS, Sharan AD, Scheid EH Jr, Vaccaro AR, Przybylski GJ. Tracheostomy placement in patients with complete cervical spinal cord injuries: American Spinal Injury Association grade A. **J Neurosurg 100**:20-23, 2004.
PUBMED
7. Hirasawa T, Aoyama K, Tanimoto T, Ishihara S, Shichijo S, Ozawa T, et al. Application of artificial intelligence using a convolutional neural network for detecting gastric cancer in endoscopic images. **Gastric Cancer 21**:653-660, 2018
PUBMED | CROSSREF
8. Kim KW, Hong HG, Nam GP, Park KR. A study of deep CNN-based classification of open and closed eyes using a visible light camera sensor. **Sensors (Basel) 17**:E1534, 2017
PUBMED
9. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. **Adv Neural Inf Process Syst 25**:1097-1105, 2012
10. Lugo Z, Pellas F, Blandin V, Laureys S, Gosseries O. Assessment of needs, psychological impact and quality of life in families of patients with locked-in syndrome. **Brain Inj 31**:1590-1596, 2017
PUBMED | CROSSREF
11. Nakayama Y, Shimizu T, Mochizuki Y, Hayashi K, Matsuda C, Nagao M, et al. Predictors of impaired communication in amyotrophic lateral sclerosis patients with tracheostomy-invasive ventilation. **Amyotroph Lateral Scler Frontotemporal Degener 17**:38-46, 2015
PUBMED | CROSSREF
12. Olczak J, Fahlberg N, Maki A, Razavian AS, Jilert A, Stark A, et al. Artificial intelligence for analyzing orthopedic trauma radiographs. **Acta Orthop 88**:581-586, 2017
PUBMED | CROSSREF
13. Tafreshi M, Fotouhi AM. A fast and accurate algorithm for eye opening or closing detection based on local maximum vertical derivative pattern. **Turk J Electr Eng Comput Sci 24**:5124-5134, 2016
CROSSREF
14. van Middendorp JJ, Watkins F, Park C, Landymore H. Eye-tracking computer systems for inpatients with tetraplegia: findings from a feasibility study. **Spinal Cord 53**:221-225, 2015
PUBMED | CROSSREF
15. Wezel J, Garpebring A, Webb AG, van Osch MJ, Beenakker JM. Automated eye blink detection and correction method for clinical MR eye imaging. **Magn Reson Med 78**:165-171, 2017
PUBMED | CROSSREF