Regular paper | Journal of the KIECS. pp. 449-458, vol. 18, no. 3, Jun. 30. 2023, t. 119, pISSN 1975-8170 | eISSN 2288-2189
http://dx.doi.org/10.13067/JKIECS.2023.18.3.449

이중 동종 CNN 구조를 이용한 ASL 알파벳의 이미지 분류

어니요조브 쇼크루크* · 권만성* · 박성종*** · 김광준**

Classifying Images of The ASL Alphabet using Dual Homogeneous CNNs Structure

Erniyozov Shokhrukh* · Man-Sung Kwan* · Seong-Jong Park*** · Gwang-Jun Kim**

요 약

많은 사람들이 수화는 청각 장애가 있고 말을 할 수 없는 사람들을 위한 것이라고 생각하지만 물론 그들과 대화하고 싶은 사람들에게 필요하다. ASL(: American Sign Language) 알파벳 인식에서 가장 큰 문제 중 하나는 높은 클래스 간 유사성과 높은 클래스 내 분산이다.

본 논문에서는 이 두 가지 문제점을 극복할 수 있는 유사도 학습을 수행하여 이미지 간의 클래스 간 유사도와 클래스 내 분산을 줄이는 아키텍처를 제안하였다. 제안된 아키텍처는 매개변수(가중치 및 편향)를 공유하는 이중으로 구성된 동일한 컨벌루션 신경망으로 구성하고 또한 이 경로를 통해 유사도 학습과 분산을 줄이는 Keras API를 적용하였다. 이중 동종 CNN을 사용한 유사성 학습 결과는 두 클래스의 좋지 않은 결과를 포함하지 않음으로써 클래스 간 유사성과 변동성을 줄임으로서 정확도가 개선된 결과를 나타내고 있다.

ABSTRACT

Many people think that sign language is only for people who are deaf and cannot speak, but of course it is necessary for people who want to talk with them. One of the biggest challenges in ASL(American Sign Language) alphabet recognition is the high inter-class similarities and high intra-class variance.

In this paper, we proposed an architecture that can overcome these two problems, which performs similarity learning to reduces inter-class similarities and intra-class variance between images. The proposed architecture consists of the same convolutional neural network with a double configuration that shares parameters (weights and biases) and also applies the Keras API to reduce similarity learning and variance through this pathway. The similarity learning results the use of the dual CNN shows that the accuracy is improved by reducing the similarity and variability between classes by not including the poor results of the two classes.

키워드

ASL(American Sign Language) Alphabet Recognition, CNN(Convolutional Neural Network), Deep Learning, Keras Deep Neural Network, Similarity Learning
ASL 알파벳 인식, 컨벌루션 신경망, 딥 러닝, Keras 심층 신경망, 유사성 학습

* 전남대학교 컴퓨터공학과(shoxrux.erniyozov@gmail.com, ,everygreen_ms@hanmail.net)
*** 전남대학교 산업기술융합공학과
(msj7681@naver.com)
** 교신저자 : 전남대학교 컴퓨터공학과
• 접 수 일 : 2023. 05. 02
• 수정완료일 : 2023. 05. 22
• 게재확정일 : 2023. 06. 17
• Received : May. 02, 2023, Revised : May. 22, 2023, Accepted : Jun. 17, 2023
• Corresponding Author : Gwang-Jun Kim
Dept. Computer Engineering, Chonnam National University,
Email : kgj@jnu.ac.kr

Ⅰ. Inroduction

ASL(American Sign Language) is very important in this day and age, and through it many facilities are being created for the deaf and dumb. This is how the human brain processes linguistic information. This posture includes many aspects, such as the movement, position and shape of the hands, as well as facial expressions and body movements. ASL is not a universal language because each country has its own alphabet and numbers, and ASL is a program based only on the English alphabet, so this program can only be used by someone who knows the English alphabet. Because of their inability to communicate with people, it is very difficult for such people to integrate into educational institutions, various jobs, and personal environments. For twenty years, automatic gesture recognition has been carried out in various ways[1]. Currently, there are 3 types of automatic gesture recognition systems: sentences, words, and gestures [1].

Finger movements are one of the most important steps in learning sign language for people who are new to using sign language and can help you make signs for the names of other words without specific signs. Some authors have proposed systems and published work. That is, the two most important of these are the vision-based and sensor-based category methods. In a sensor-based process, the gesturing person wears a glove and a sensor to provide external and accurate information about the hand's position, rotation, and orientation. However, these methods are very difficult and cause discomfort for people [1]. Vision-based methods, on the other hand, are popular because they do not require human-attached sensors and can be implemented with inexpensive cameras. Vision-based methods use electronic digital imaging, image processing, and machine learning techniques.

This paper is organized as follows: Section Ⅱ presents the related work. In section Ⅲ, some methodologies and the system structure of dual homogeneous convolutional neural networks by reducing inter-class similarities and intra-class variance between images, and experiments and results analysis are shown in section Ⅳ. Finally, the conclusion is in section Ⅴ.

Ⅱ. Related Work

According to the recognition task, the ASL alphabet is built on two subtasks: distinguishing features and performing multiclass classifications. [2] authors used color and depth images with filters to extract features from Gabor images and achieved 49 percent accuracy over random forest. In [3], the authors proposed Superpixel Earth Mover Distance (SP-EMD) to extract texture, shape, depth information from images, and determine distance between image features. For character classification by applying the template matching method, a recognition result of 75.8% was achieved. Another similar work is in [4], where volumetric spatiogram of binary pattern (VS-LBP) was used for feature extraction and 83.6% accuracy was achieved using Support Vector Machine (SVM). From complex images [5], differences were extracted and 81.2 % accuracy was achieved. Authors in [6-8] used random forest and depth images to recognize 24 classes of ASL alphabet and achieved 87% and 90% accuracy. These techniques rely on the two sub-tasks of feature classification and feature extraction, as we noted above, where the features considered are well known as hand-crafted due to human intervention. This separation results in a "split event" where some information is missing to classify a feature in the process of separation.

There are CNN networks that have the advantages of feature extraction and classification. Convolutional layers are responsible for capturing non-linear images (feature extraction) and Fully Connected (FC) layers encode and classify these images. A CNN with two inputs was introduced in [9]. Image categorization of color and complexity images was previously combined in fully connected parts, resulting in 80.33% purity. A new visual zoom is proposed in [10]. In it, only one complex image 3D point cloud is obtained. Then more cameras are moved and pointed at the point cloud with different perspectives. Finally, additional sets of views are generated from those distributed virtual cameras. In Microsoft Kinnect sensor [1], the authors proposed to use complex images obtained and extract features from them using PCANet, and then classify these features using Support Vector Machine (SVM) with 84.5% accuracy.

III. Proposed Method

3.1 Convolution Neural Network

Clearly, CNNs have taken the lead in image-related tasks on DNNs. First of all, due to the availability of large-scale annotated data sets (i.e., ImageNet) and the recovery of deep convoluted neural networks, remarkable advances in image recognition have been made [11]. CNN is similar to traditional DNNs in that they are made up of neurons that optimize themselves through learning. The difference between CNN and traditional DNNs is that CNNs are primarily used in the field of image recognition within images. This allows us to encode image-specific features in the architecture, making CNN's more suitable for image-focused tasks which further reduces the parameters required to customize the model.

Standard neural networks cannot eliminate the shifts and distortions in images that often occur in a set of images. However, despite these shortcomings, DNNs can still explore robust properties. Therefore, LeCun developed an earlier version of CNN, which was designed for computers and visual abilities that provide visual variability Fig. 1 of CNN shows the convolution and subsampling layers, which are powerful tools in image extraction that provide the boundaries of edges and objects. The model consists of convolution and recursive application of bonding layers, followed by internal product layers at the end of the network.

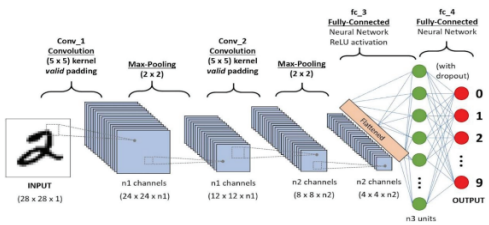

그림 1. 컨벌루션 신경망 구조
Fig. 1 Convolution neural networks architecture

In CNNs, convolution layers are constructed by limiting the receiving areas of hidden units to a local connection and adding shift-variability by amplifying spatially distributed weights. Simply put, S(i, j) is obtained by multiplying a set of filtered outputs f(m, n) (also called the core) to the input image I until the end of the pixel shift. The process of discrete convolution between the filter matrix of Fig 1 and f is defined as follow:

$$S(i,\ j) = (I * K)(i,\ j) = \sum m \sum n\ I(i - m,\ j - n)\ f(m,\ n)$$

where i, j are the row and column pointers of the feature map m and n are the filter sizes. After convolution, the convoluted output is activated with nonlinear activation functions. The

process is repeated with n filters, then the convolution layer with n filters creates n property maps, which will be the future entry for the next layer.

3.2 Proposed Architecture with Dual Homogeneous CNNs

One of the biggest challenges in ASL alphabet recognition is the high inter-class similarities and high intra-class variance. In this paper, we propose an architecture that can overcome these two problems, which performs similarity learning and thus reduces inter-class similarities and intra-class variance between images. For experiments, we initially used a small two networks architecture. For example, one architecture consisted of 4 convolutional layers and 1 fully connected layer, but this architecture was over clocked, and despite using a high dropout rate, the network did not merger. We concluded from this experiment that the final feature maps were too small and the network was difficult to learn well. So, we decided to increase the number of convolutional layers up to 6 and save the size of the feature maps using padding, and also increase the number of dense layers, since they are responsible for coding; this architecture achieved 91% confirmation accuracy. This accuracy value was too small, so we decided to add two more convolutional layers, as well as increase the number of neurons of the last dense layer. The proposed scheme was chosen because it showed better performance compared to other experimental architectures. The proposed architecture consists of dual identical convolutional neural networks that share their parameters (weights and biases). Each of these two CNNs consists of 8 convolutional and 3 fully connected (dense) layers, as shown in Figure 2.

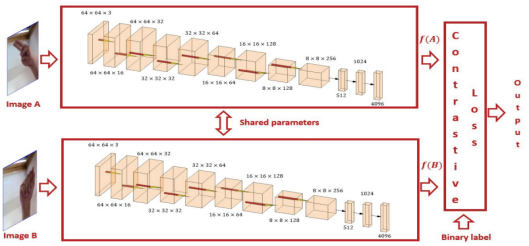

그림 2. 이중 동종 CNN으로 구성된 구조
Fig. 2 Architecture consists of dual homogeneous CNNs

A pair of images is provided as input, where this pair of images can be positive (identical images). class) or negative (images belonging to different classes). These images are fed to convolutional layers, which are responsible for extracting features such as color, texture, shape, edges, and directions. Unlike CNN-based systems for image classification, the dense layers of the proposed scheme only encode image features instead of encoding-classification. This encoding is fed by contrast loss, where similarity learning is performed. This similarity study uses the distances between each pair of feature vectors generated by the last dense layer to obtain a score that measures the similarity or dissimilarity between pairs of images (positive and negative, respectively).

3.3 Similarity Learning

As we mentioned above, a pair of images (A and B) are sent to networks; We proposed to use 64x64x3 images to reduce computational costs. Each network generates a 4096-dimensional feature vector (f(A) and f(B), respectively). Each CNN architecture for image classification consists of convolutional layers for feature extraction and dense layers for encoding and classification, and the last the number of neurons in the dense layer is equal to the number of classes.In this case, the last dense layer of the proposed architecture

consists of 4096 neurons because it is necessary to have a high-dimensional image representation to reduce the interclass similarities.

In order to perform a similarity learning, first, the distance between the encoding of image A (f(A)) and image B (f(B)) is obtained as follows:

$$D(A,B) = \sqrt{\sum_{i=1}^{n}(f(A)_i - f(B)_i)^2},$$

where D(.) is the distance between f(A) and f(B). If equation 1 is small, it means that A and B belong to the same class and vice versa. The contrastive loss is responsible for similarity learning and is defined as:

$$L = \frac{1}{2}lD^2 + \frac{1}{2}(max(0, m - D))^2,$$

where l is a binary label indicating if A and B belong to the same class (l = 1) or not (l = 0); m is a margin selected for dissimilarity images (m must be greater than zero). As can be observed from equation 2, the distance between two images of the same class.

Each network receives a representation of an input image and then feeds them with contrast loss to learn similarity. The output of Siamese architecture is that the score indicating the similarity of a pair of images should be small, and the distance should be large for images belonging to different classes. Thus, networks generate codes for each image so that those belonging to the same class have a small distance and vice versa. As a result, large cross-class overlap and large intra-class variation are reduced, which improves the classification rate of the ASL alphabet.

While deep neural networks are all the rage, the complexity of the major frameworks has been a barrier to their use for developers new to machine learning. There have been several proposals for improved and simplified high-level APIs for building neural network models, all of which tend to look similar from a distance but show differences on closer examination.

Keras is one of the leading high-level neural networks APIs. It is written in Python and supports multiple back-end neural network computation engines. Keras, on the other hand, is perfect for those that do not have a strong background in Deep Learning, but still want to work with neural networks. Using Keras, you can build a neural network model quickly and easily using minimal code, allowing for rapid prototyping.

Ⅳ. Experimental Results

4.1 Dataset Introduction

The dataset used was based on the American Finger Spelling format. Figure 3 shows the followed format.

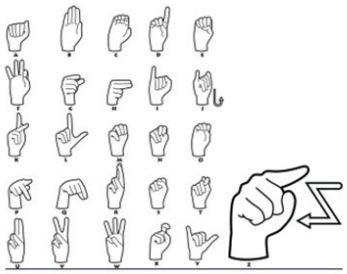

그림 3. 각 문자에 대한 데이터 세트를 구성할 때 기본적으로 사용되는 손 포즈
Fig. 3 Hand poses used as a basis in constructing the dataset for each letter.

The dataset used in training the network was composed of static hand poses of letters form the alphabet. This meant excluding the letters j and z due to the motion required to represent them. The work of [8] provided a set of images containing

more than 50,000 images of letters formed by the hands. Five sessions were conducted to collect five different sets of letters from different people. They conducted their data collection under similar lighting and background conditions. Although the dataset discussed contained many images for training, it was still insufficient to achieve the accuracy and robustness desired for this work. Its data collection was restricted to specific lighting conditions and backgrounds. As a result, we acquired a new set of images in more varied lighting conditions and using more variations in the hand letters. Some images were taken such that the hand does not necessarily cover the whole image, and others show the hand being tilted in different ways to take into account these variations. The background chosen for the images were also changed for the duration of the data collection.

4.2 Network Architecture

Given an image of a letter hand pose at test time, our goal is to construct a network that can properly classify the image to its corresponding letter using a deep neural network. In order to achieve this, we used the DenseNet architecture from. Figure 4 shows an implementation of a DenseNet block.

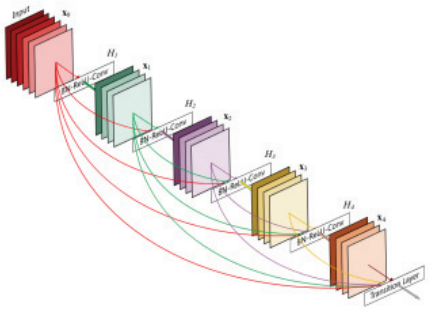

그림 4. 5계층을 이용한 dense 블록의 구조
Fig. 4 Structure of a dense block with 5 layers.

This was used to alleviate the problem of vanishing gradients despite using a deep network architecture. Furthermore, it encourages feature reuse across layers, while minimizing the number of parameters. The latter characteristic helps in applying the network for real-time prediction using a web camera. Fewer parameters would result to less training and prediction time, a desirable quality for the purposes of this work. In the implementation of the network, four such dense blocks in Figure 15 were used, in addition to three transition layers. Each dense block consisted of applying ten layers of batch normalization, relu activation, and a 3x3 convolutional layer. Transition layers on the other hand, use batch normalization, relu, a 1x1 convolutional layer, and a 2x2 average pooling layer. The transition layers are used to effectively reduce the feature map size after every dense block.

4.3 Real-time evaluation

The network was also tested on its real-time predictions on more varied environment. This proved the ability of the network to generalize on data it has not seen before using hands of people it has not yet encountered. It was easier to test the accuracy and reliability of the network with this method due to the ease at which the environmental factors can be changed.

We created an app for sign language assistant android phones in this diploma work. We can see this app in the figure 5 below. In this window we can see 3 stages of the application I created. That is, there are Recognition, Learning, Training buttons, which function as follows.

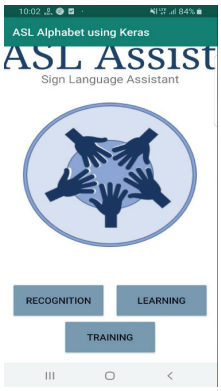

그림 5. 메인 윈도우
Fig. 5 Main window

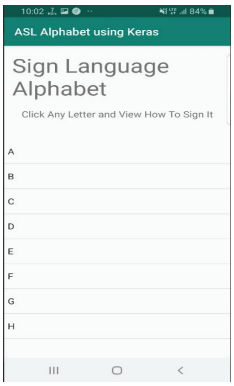

그림 6. 알파벳 테이블
Fig. 6 Table of alphabet

When we click the Learn button, the following window will appear and from this window. We can see what character each of the 26 letters has in figure 7 of the sign language. In this section, windows of the same appearance are created when we click on each letter.

그림 7. 학습된 손동작 윈도우
Fig. 7 Window for learning hand movements

In the pictures below which is figure 8 is the status of the application I created. In these windows, when I hold the shape of my hand on the camera, it shows what letter the symbol is at the top.

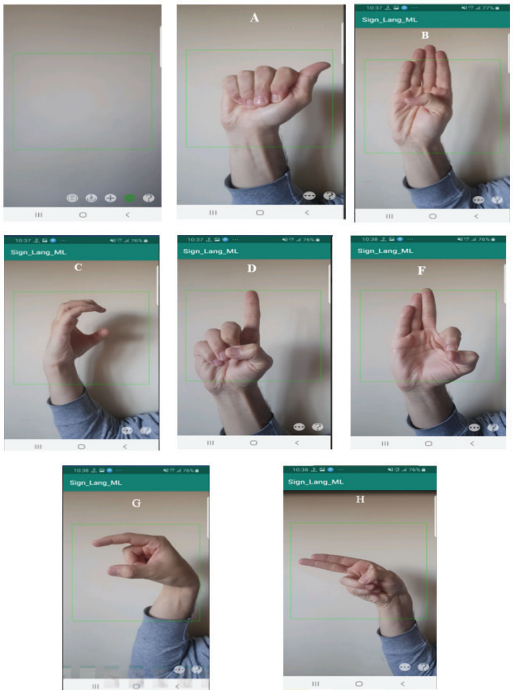

그림 8. 구현된 프로그램의 상태
Fig. 8 Status of the implemented program

In this section, you will be shown a letter at the top of the window, and you will manually indicate what character that character is in. If you show it correctly, the score below will show how many you have found.

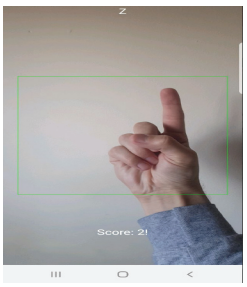

그림 9. 훈련 부분의 응용 프로세스
Fig. 9 Application process of the training part

4.4 Result

In order to evaluate the classification performance, we compute the confusion matrix

shown in Fig 10. The confusion matrix is a performance measurement for classification problems. It can be seen from Fig 10 that the proposed scheme is doing an excellent performance on classifying the 29 classes.

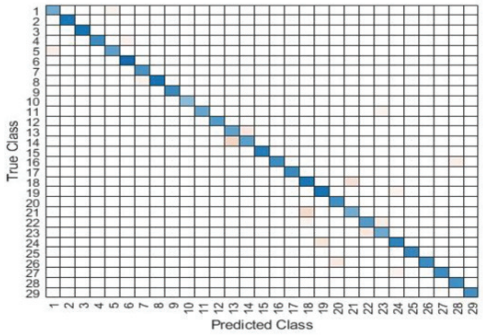

그림 10. Keras 구조를 이용한 오차 행렬
Fig. 10 The confusion matrix using the Keras scheme

We have used the accuracy, precision, and recall metrics to provide an evaluation in a quantitative manner. The results of these metrics are shown in Fig 23. Precision is the ratio of correctly predicted positive observations to the total predicted positive observations; recall, on the other hand, is the ratio of correctly predicted positive observations to all observations in the actual class.

From Fig 11, we can observe that for the sign “M” and “N”, the proposed scheme achieved 93% and 85% of accuracy, respectively, and for the pair “R” and “U” achieved 86% and 85%, respectively. These values of accuracy were lower compared to the rest of the alphabet. This is because the sign for these letters is very similar and despite of having used a Siamese architecture, it remains some level of interclass similarity. However, the average classification performance of the proposed method achieved an accuracy of 95%.

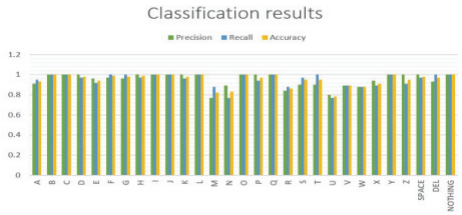

그림 11. ASL 알파벳 데이터셋의 클래스별 분류 결과
Fig. 11 Classification results per class of the ASL Alphabet dataset

The proposed scheme was compared to published works where authors propose some other techniques for the same purpose but using different types of images (RGB and depth images)

Despite having these errors, the network performed at an accuracy that is almost at par with existing systems. Table 2 shows the performance of our network in comparison to others.

표 2. 제안된 방법과 기존 연구의 비교
Table 2. Comparison of the proposed method to published works

Method	Accuracy[%]
Aly et al. [1]	84.5
Salem and Vadera. [4]	80.3
Dong et al. [12]	90
Kuznetsova et al. [10]	87
Maqueda et al. [3]	83.7
Nai et al. [6]	81.1
Pugeault and Bowden [7]	49
Tao et al. [5]	84.7
Wang et al. [11]	75.8
Proposed	96

Methods using depth images had better accuracies compared to methods using only color images. Our network achieved comparable results despite using only color images. Furthermore, the

difference in the testing accuracy can also come from the variation in the test and training dataset, since the networks used different sources of data. In this paper, we have proposed a system for ASL alphabet recognition which can help either hearing or no hearing people to learn sign language. The ASL language combines, as we mentioned above, hand movements and facial expressions.

Ⅴ. Conclusion

Many people think that sign language is only for people who are deaf and can't speak, but of course it is necessary for people who want to talk with them. Nowadays, such disabled people face various obstacles to find their place in society. In order to reduce these barriers, it is necessary to develop many such advanced technologies. In this article, we present an Android-based application for ASL alphabet recognition. Among the most difficult steps in this task are high interclass similarities and high variability. Our hypothesis was then to obtain an image encoding where those belonging to one class should be separated by a small distance (low variability) and simultaneously separated by a large distance (low similarity) from those belonging to another class. therefore, we proposed a Keras architecture that uses two identical CNNs through this pathway. the developed result shows that our hypothesis is correct. We believe this is because we have reduced the between-class similarity and variability by not including the poor results from the two classes. However, overall, we felt that the proposed scheme performed well in classification. The comparison presented in this paper shows that our neural architecture can outperform the work published in the literature

References

[1] W. Aly, S. Aly, and S. Almotairi, "User-independent american sign language alphabet recognition based on depth image and pcanet features," *IEEE Access*, vol. 7, 2019, pp. 123138-123150.

[2] A. Fierro, M. Nakano, K. Yanai, and H. Perez, "Siamese and triplet convolutional neural networks for the retrieval of images with similar contents," *Informacion Tecnologica*, vol. 30, no. 6, 2019, pp. 243-254.

[3] A. I. Maqueda, del Blanco, C. R., F. Jaureguizar, and N. Garc´ıa, "Human–computer interaction based on visual hand-gesture recognition using volumetric spatiograms of local binary patterns," *Computer Vision and Image Understanding*, vol. 141, 2015, pp. 126-137.

[4] A. Salem and S. Vadera, "A convolutional neural network to classify american sign language fingerspelling from depth and colour images," *Expert Systems*, vol. 34, no. 3, 2017, pp. 1-18.

[5] W. Tao, M. C. Leu, and Z. Yin, "American sign language alphabet recognition using convolutional neural networks with multiview augmentation and inference fusion," *Engineering Applications of Artificial Intelligence*, vol. 76, 2018, pp. 202-213.

[6] W. Nai, Y. Liu, D. Rempel, and Y. Wang, "Fast hand posture classification using depth features extracted from random line segments," *Pattern Recognition*, vol. 65, 2017, pp. 1-10.

[7] N. Pugeault and R. Bowden, "Spelling it out: Real-time asl fingerspelling recognition," *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, 2011, pp. 1114-1119.

[8] J. P. Sahoo, S. Ari, and D. K. Ghosh, "Hand gesture recognition using dwt and f-ratio based feature descriptor," *IET Image Processing*, vol. 12, no. 10, 2018, pp. 1780-1787.

[9] Kaggle homepage. [Online available]: https://www.kaggle.com/grassknoted/asl-alphabet. [Accessed: 20/06/2020].

[10] A. Kuznetsova, L. Leal-Taixe, and B. Rosenhahn, "Real-time sign language recognition using a consumer depth camera," *2013 IEEE International Conference on Computer Vision Workshops*, 2013, pp. 83-90.

[11] C. Wang, Z. Liu, and S. Chan, "Superpixelbased hand gesture recognition with kinect depth camera," *IEEE Transactions on Multimedia*, vol. 17, no. 1, 2015, pp. 29-39.

[12] Cao Dong, M. C. Leu, and Z. Yin, "American sign language alphabet recognition using microsoft kinect,", *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2015, pp. 44-52.

[13] C. Yeon and K. Seok, "Inter-module interworking evaluation of TDMA-based wireless IP video transmission system," *J. of the Korea Institute of Electronic Communication Sciences*, vol. 18, no. 1, 2023, pp. 1-10.

저자 소개

어니요조브 쇼크루크 (Erniyozov Shokhrukh)

2018년 Tashkent University of Information Technologies, Telecom munication Technology, 졸업(공학사)
2018년 ~현재 전남대학교 대학원 컴퓨터공학과 재학(석사과정)
※ 관심분야 : Machine Learning, Real-Time Communication, AI, IoT

권만성(Man-Sung Kwan)

2003년 우석대학교 유통통상학부 졸업(경영학사)
2020년~현재 ㈜씨엔에스컴퍼니 대표이사 경영 및 CNS AI 광학 연구소 소장
2022년~현재 전남대학교 대학원 컴퓨터공학과 재학 (석사과정)
※ 관심분야 : 스마트글라스, 인공지능, 빅데이터, 실시간 통신

박성종(Seong-Jong Park)

1989년 전남대학교 화학공학과 졸업(공학사)
2005년 전남대학교 대학원 경영학과 졸업(경영학박사)
2022년~현재 전남대학교 산업기술융합공학과 교수
※ 관심분야: 실시간 통신, 사물인터넷, 인공지능, 스마트 팩토링

김광준(Gwang-Jun Kim)

2000년 조선대학교 대학원 컴퓨터공학과 졸업(공학박사)
2019년~현재 한국융합기술연구학회 수석부회장
2003년~현재 전남대학교 공학대학 전기·컴퓨터공학부 교수
※ 관심분야 : 실시간 통신, 컴퓨터 네트워크, 클라우드 컴퓨팅, 인공지능, 사물인터넷