An effective approach for detecting and identifying human hand gestures using convolutional neural network

Sunil G. Deshmukh[1]*, Shekhar M. Jagade[2]

[1]Department of Electronics and Computer Engineering, Maharashtra Institute of Technology, Aurangabad, Maharshtra-431010, India

[2]Department of Electronics and Telecommunication, N B Navale Sinhgad College of Engineering, Solapur, Maharashtra-413255, India

*__Corresponding author:__ Sunil G. Deshmukh, __Email:__ sunildeshmukh7472@gmail.com

Abstract

Human gestures are a non-verbal method of communication that is essential in interactions between humans and robots. In order to recognize hand movements and facilitate such interactions, vision-based gesture recognition techniques are crucial. A simple and useful interface between gadgets and people is made possible by hand gesture recognition. Hand gestures may be employed in many different contexts, making them useful for communication and other purposes. People with hearing loss or disabilities, as well as those who have had strokes, might benefit from hand gesture recognition because they need to be able to interact with others by employing gestures that are universally understood, such as the signs for food, drink, family, and more. This study suggests a method for identifying hand motions using convolutional neural networks (CNN). Based on a number of criteria, including execution time, accuracy, sensitivity, specificity, positive and negative predictive value, probability, and root mean square, the proposed approach is assessed and contrasted between training and testing modes. The results demonstrate that CNN is a successful method for identifying distinctive characteristics and categorizing data, with testing accuracy of 100%.

Keywords: Computer vision, Machine learning, Deep learning, Pattern recognition, Hand gesture, Convolutional neural network

DOI Number: 10.14704/nq.2022.20.13.NQ88128 **Neuro Quantology 2022; 20(13):1006-1013**

1. Introduction

Recently, direct touch has become the most common method of user-machine communication. Devices like a mouse, keyboard, remote control, touch screen, and other direct contact techniques are the foundation of the communication channel [1]. Natural and intuitive non-contact approaches, such as sound and bodily motions, are used to communicate between people. Many researchers are considering using these non-contact communication techniques to promote human-computer interaction because of their adaptability and effectiveness [2] [3]. A significant component of human language, gestures is a vital non-contact communication tool. In the past, wearable data gloves were often employed to record the angles and locations of each user's joint during a motion [4]. The general use of such a technique has been constrained by the complexity and expense of a worn sensor. Gesture recognition is the capacity of a computer to comprehend gestures and execute specific instructions in response to such motions. The basic objective of gesture recognition is to create a system that can recognize, comprehend, and transmit information from certain gestures [5].

There is now a trend toward gesture recognition techniques based on non-contact visual examination. This is as a result of their affordability and user-friendliness. In addition to helping users with special needs and the elderly, hand gestures are an expressive communication technique utilized in the healthcare, entertainment, and educational sectors of the economy. Hand tracking, which incorporates several computer vision operations such as hand

segmentation, detection, and tracking, is essential for performing hand gesture identification [6]. To communicate with people who have hearing loss, sign language makes use of hand movements to express emotions or information. The key issue is that the message might be readily misunderstood by the average individual. To identify and understand sign language, AI and computer vision advancements may be used [7]. An average individual may learn to identify and comprehend sign language with the aid of current technology. In this article, a deep learning-based technique for hand gesture detection is presented.

In this sense, gestures are crucial for using the system in remote mode. Human gestures are captured by the machines, which then identify them as being used to operate them. Different sorts of modes, including static and dynamic, are used for the motions. While the machine is being controlled, the static motions remain in place, while the dynamic gestures move into different locations [8][9]. Consequently, it is crucial to identify or recognize dynamic motions rather than static gestures. At first, the camera that is attached to the device records the motions that individuals make. The gesture's foreground is collected after the backdrop of any detected motions is eliminated. Filtering methods are used to identify and get rid of the sounds in the foreground gesture [10]. To confirm the meaning of the gestures, these noise-removed gestures are compared to pre-stored and taught motions [11].

Without requiring any human input, the gesture-based machine operating system is used in the automotive and consumer electronics industries. Human gestures may also be divided into static and dynamic gestures, as well as online and offline motions [12]. The offline gestures control the machine's icons, but they are unable to change where the objects are shown in the system or menu. The machine's symbols are moved or inclined in various ways by the internet motions [13]. In real-time machine operating systems, online gestures are far more helpful than offline gestures. These techniques needed a lot of training samples and did not support huge training datasets. By recommending CNN classifier in this work, this fault is solved. The complexity of this approach is not large, and it doesn't need a lot of data to train. The innovative aspect of the proposed work is the integration of a deep learning algorithm and a cutting-edge segmentation method into a hand gesture detection system [14] [15].

Therefore, one of the aspects of hand gestures that has to be addressed with is non-linearity. The photos' metadata and content data may be used to do this. Images of hand movements may be recognized using their meta-information. Feature extraction and classification are two activities that are integrated throughout the process [16]. The characteristics of a picture must be retrieved before any gesture can be recognized. You should use any classification approach after obtaining those characteristics. The major issue is thus how to extract and infer such characteristics for categorization [17].

1.1 Motivation for Opting CNN

It is essential to have large features for categorization and identification. Natural data in its raw form cannot be processed by conventional models for pattern recognition [18]. Therefore, it takes a lot of work and is not automated to extract characteristics from raw data. CNNs, a subset of deep learning neural networks, can classify data using fully connected layers and extract features on-the-fly [19]. CNN combines these two processes to improve performance by lowering memory needs and computational complexity. It is also capable of comprehending the intricate and illogical connections between the visuals. In order to tackle the issue, a CNN-based technique will be applied [20] [21].

So, employing a convolutional neural network, this article offered a unique framework for classifying gestures. A multi-layer feed-forward neural network with biological inspiration, the convolutional neural network has several layers [22]. The convolutional neural network may automatically learn successive stages of invariant characteristics for the particular job, in contrast to conventional approaches that use hand-crafted features. We employed two picture bases comprises 24 gestures in this study, along with certain segmentation methods and convolutional neural networks (CNNs) for categorization. Thus, using the suggested technique, we showed that it is feasible to

classify static gestures with great results using simple convolutional neural network designs [23].

2. Materials and Methods

The methods utilized in this study for data categorization and picture processing are covered in this section. A classifier must be trained, which requires the extraction of data. In addition, while working with photos, it is critical to selectively extract features from the areas of interest. To improve the relevant information for a certain application, segmentation methods, filters, and morphological processes are performed. Since convolutional neural networks take an image as its input, it is not essential to remove feature vectors from the pictures before using them. With these factors in mind, a successful pre-processing step should effectively isolate noise from key picture elements.

2.1. Proposed Convolutional Neural Networks (CNNs) Methodology

CNNs, also known as convolutional neural networks, are often used for object identification, recognition, and categorization in images. CNN is a potent image processing method. Right now, these are the finest algorithms available for automatically processing photos. These algorithms are widely used by businesses to do tasks like item identification in images [24].

CNN is a multi-layer neural network with a distinctive design used for machine learning, according to its definition. CNN is often used for detection process, retrieval, and classification as well as object and scene recognition. Due to the following three factors, CNN has been heavily utilized in recent years: Since CNN can learn the picture data directly, (1) feature extraction by means of image processing tools is no longer necessary, (2) it performs very well for results recognition and (3) it can be constructed on an entire network [25].

2.1.1. Three Layers of CNN

There are three types of layers in Convolutional Neural Networks [26]:

1. Convolutional Layer: Each input neurons in a typical neural network is linked to the subsequent hidden layer. Only a tiny portion of the input layer neurons in CNN are connected to the hidden layer of neurons.

2. Pooling Layer: The characteristic map's complexity is decreased using the pooling layer. Inside the CNN's hidden layer, there will be several activation and pooling layers. The representations will be resistant to both geometrical distortions and slight shifts thanks to the pooling and sub-sampling layers. On the convolution layer's output, the pooling layer applies average pooling. By calculating the mean of all the values in an area, average pooling takes portions of a picture and converts them to a numerical result [27].

3. Fully-Connected layer: The final levels of the network are known as Fully Connected Layers. The information from the last pooling or convolutional layer is passed into the completely connected layer, where it is flattened before being applied.

2.2.2. Working methodology of CNN

A picture characterized by an arbitrary color model serves as the input for a CNN when it comes to image categorization. Every neuron in the convolution layer is connected to a kernel frame that, during CNN training and classification, convolves with the input picture. The corresponding neuron weights make up this convolution kernel. A series of N pictures, one for each of the N neurons, is the result of this convolution process. These new pictures may have negative values due to convolution. A rectified linear unit (ReLU) is used to replace negative numbers with zero in order to get around this problem. Feature maps are the name given to this layer's outputs [28].

It is typical to put a pooling layer after a convolution layer. This is significant because pooling decreases the dimensionality of feature maps, which in turn decreases the amount of time needed to train the network. A multilayer perceptron neural network conducts categorization utilizing the feature maps produced by the preceding layers after the convolution and pooling designs. CNNs are one of the most popular deep learning algorithms due to their many layers and effective applications. Various visual elements including edges, circles, lines, and texture are

automatically extracted by its architecture. In subsequent layers, the extracted characteristics are more optimized. It is crucial to stress that backpropagation during CNN training is what determines the values of the kernel filters used in the convolution layers [29].

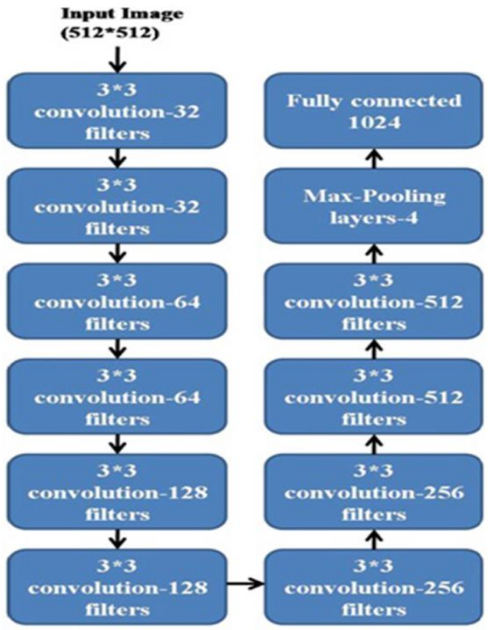

Figure 1. Developed CNN architecture

The developed CNN architecture employed in this study's hand motion picture categorization is shown in Figure 1. It includes fully connected layers, pooling, and convolutional filters that multiply the input image's kernel by (7*7), as illustrated in Figure 1. Convolution layers and fully linked layers are shown in Figure 2's internal architecture, which is a comprehensive internal representation of the proposed CNN classifier for hand gesture identification. N output classes are produced by the completely linked layers.

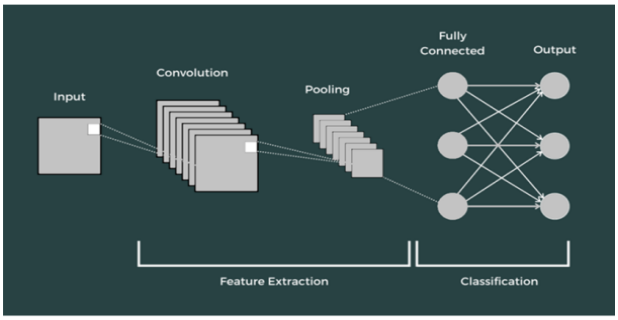

Figure 2. Detailed internal architecture of proposed CNN

2.2.3. Dataset

Among the most crucial aspects of any investigation is data collection. It is crucial to gather information that is pertinent to the study and complies with its standards. The dataset for the proposed research project has been designed to closely resemble the MNIST classic dataset. A well-known benchmark for graphics machine learning algorithms is the iconic MNIST image dataset of handwritten characters. Researchers have nonetheless redoubled their attempts to modernize it and create drop-in substitutes that are more difficult for computer vision and unique for practical applications [30].

27,455 training images and 7,172 test images make up the dataset. Each picture has a grayscale structure and a 28 × 28-pixel size. The label for the letter in the alphabet that each picture represents is included. This dataset contains 24 letters; the letters "j" and "z," which both entail motion, are not included. The dataset format roughly resembles the traditional MNIST. There are no examples for 9=J or 25=Z due to gesture movements; each training and test case represents a label (0–25) as a one-to-one mapping for each alphabetical letter A–Z. A single 28x28 pixel picture with grayscale values ranging from 0-255 is represented by each of the training data (27,455 instances) and test data (7172 cases), which are about half the size of the regular MNIST dataset but otherwise identical. Multiple users repeating the move across various backdrops were represented by the original hand gesture picture data. The limited number (1704) of color photos provided that were not trimmed from around hand area of interest was considerably increased by the Sign Language MNIST data. After studying the dataset, it is seen that the data is equally distributed over the values, as illustrated in Figure 3. The picture shows that the large range of training sample values are spread equally [31].

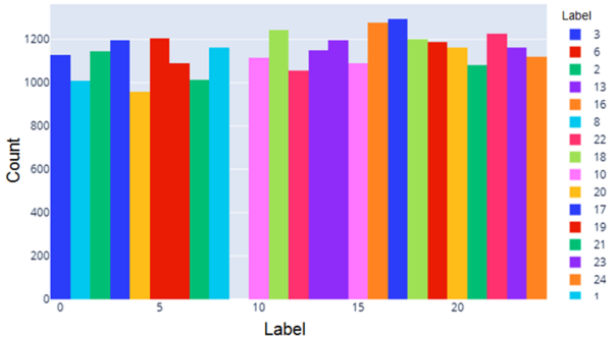

Figure 3. Distribution of labels in the training set.

Figure 4. Set of training images

3. Processing the Images

3.1. Creating Our X and Our Y

The labels are listed in the dataset's first section. The label and pixels data will be separated first.

3.2. Scaling Our Images

The data must then be scaled from 0-255 to 0-1. As a result, the neural network will converge more quickly, making it simpler to work with.

3.3. Reshaping Our Images

To input into our model, the photos will need to be altered. Since we'll be utilizing grayscale, our pictures will be 28x28, with a color channel of 1.

3.4. Distribution of Labels in the Training Set

The training images are dispersed quite evenly. Since they signify j and z, which aren't in the database, the labeled 9 and 25 columns are vacant.

3.5 Viewing the Images/Training Images

It is permitted to choose the photos in the Train set and divide individuals into the training and testing sets. It is advised to utilize the validation set in conjunction with the new train set while training the model to ensure accuracy. The pixel values from the test set are chosen once the model has been programmed to anticipate the labeling for each test picture, as illustrated in Figure 4. The effectiveness of the model is then assessed by reviewing a categorization report [32].

3.6. Training of models

Once the dataset has been created, it may be verified by showing a gesture along with the numbers 0 to 9 for each letter of the alphabet, from A to Z. These photos are 50x50 in size. The model is then trained over many epochs to see where the best outcomes may be obtained. The software is performed in 4 separate epochs for this purpose [33].

4. Results and discussion

Figure 5 below displays the findings for the training accuracy, validation accuracy, and validation loss.

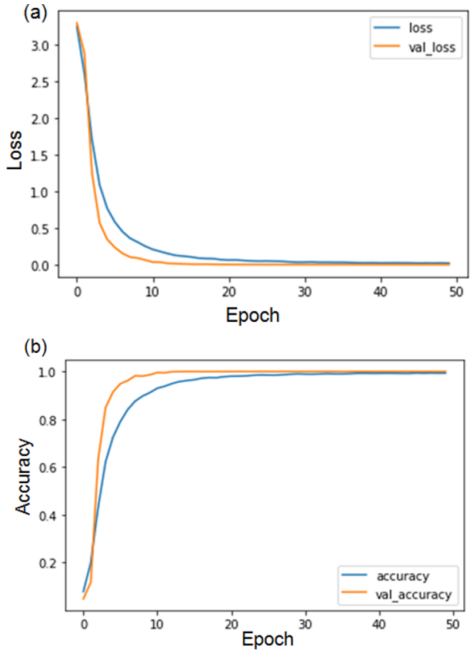

Figure 5. Training and accuracy test (a) Loss VS Epoch (b) Accuracy VS Epoch

Up to a certain amount, it has been discovered that accuracy grows as the number of epochs rises. After then, it falls because the model has been over-trained. Additionally, as the number of epochs rises, so does the computational time. Table 1 shows the accuracy and a sample of validation losses for each training period of the model. At each of the first four epochs, loss dropped and accuracy rose. The loss began to rise after the fourth epoch, and the model began to over fit.

Table 1. Sample values of Validations loss and accuracy gained

Epoch	Validation loss	Validation accuracy
17	0.0055	0.9995
18	0.0057	0.9998
21	0.0016	0.9999
50	7.7203e-05	1.0000

4.1 Classification Report

The classification reports, which include accuracy, recall, and an F1-score for each class from 0 to 24, provides the information. Recall indicates the number of favorable samples that are properly identified as positive, while precision demonstrates the classifier's capacity to identify just the true positive observations as beneficial. The accuracy, recall, and F1 score % for each class are shown in Table 2. The lowest accuracy and recall for both 6 and 7 were noted to be 95% and 99%. The lowest and maximum F1-score percentages for classes 6 and 100%, respectively, were 97% and 100%.

Table 2. Precision, recall and F1 score for each 24 digits

	precision	recall	f1-Score
0	1.00	1.00	1.00
1	1.00	1.00	1.00
2	1.00	1.00	1.00
3	1.00	1.00	1.00
4	1.00	1.00	1.00
5	1.00	1.00	1.00
6	0.95	0.99	0.97
7	1.00	0.96	0.98
8	1.00	1.00	1.00
9	1.00	1.00	1.00
10	1.00	1.00	1.00
11	1.00	1.00	1.00
12	1.00	1.00	1.00
13	1.00	1.00	1.00
14	1.00	1.00	1.00
15	1.00	1.00	1.00
16	1.00	1.00	1.00
17	0.98	1.00	0.99
18	1.00	1.00	1.00
19	1.00	1.00	1.00
20	1.00	0.99	0.99
21	1.00	1.00	1.00
22	0.99	1.00	1.00
23	1.00	1.00	1.00
24	1.00	1.00	1.00
		Accuracy :	**1.00**

The categorization report states that 100% accuracy is preferred. The batch normalization and dropout layers definitely helped with improving the accuracy, which was the main takeaway. The training results also seemed to be rounded off by the Dropout layers. Mathematically, the report can be represented in a similar manner as given in equation 1, 2 and 3;

$$precision = \frac{True\ positive}{True\ positive + False\ positive} \quad (1)$$

$$recall = \frac{True\ positive}{True\ positive + Negative} \quad (2)$$

$$f1 - Score = 2 * \frac{precision * \mathrm{Re}call}{precision + \mathrm{Re}call} \quad (3)$$

According to the report, the average number is quite near to 1, which indicates that the model is producing accurate findings.

5. Conclusion

The ability to recognize gestures and sign languages is a highly sought-after skill in the contemporary world. For the future generation of robotic assistants as well as a tool for the deaf, we think it is crucial. This study demonstrates the enormous potential of neural networks in this application. On top of that, using our own dataset of sign language number motions, we get encouraging recognition results.

In this study, hand gesture recognition using a convolution neural network was the focus. The identification of sign language is one of the key uses of hand motion recognition. One of the means of communication for those who are physically disabled, deaf, or dumb is sign language. The distance between normal, deaf, and stupid individuals will be reduced with the use of this instrument. It is inferred from the data above that Convolutional Neural Networks significantly improve sign language character identification. My suggested approach attained a validation accuracy of 100% for the dataset. This research may be further upon by creating a real-time program that can distinguish phrases and sentences in addition to merely characters when identifying sign language.

References

1. Kelly SD, Manning SM, Rodak S, “Gesture gives a hand to language and learning: Perspectives from cognitive neuroscience, developmental psychology and education”, Language and Linguistics Compass, Vol. 2, No. 4, (2008), 569-588. DOI: 10.1111/j.1749-818X.2008.00067.x
2. Cohen MW, Zikri NB, Velkovich A, “Recognition of continuous sign language alphabet using leap motion controller”, in 2018 11th international conference on human system interaction (HSI), IEEE.(2018),193-199.
3. Shahbakhsh MB, Hassanpour H, “Empowering Face Recognition Methods Using a GAN-based Single Image Super-Resolution Network”, International Journal of Engineering, Vol. 35, No. 10, (2022), 1858-1866. DOI: 10.5829/ije.2022.35.10a.05.
4. Betancourt, A., P. Morerio, C. Regazzoni, and M. R. Auterberg. "A structure for deoxyribose nucleic acid." Circuits and Systems for Video Technology, Vol. 25, No. 5, (2015), 744-760.
5. Riofrío S, Pozo D, Rosero J, Vásquez J, “Gesture recognition using dynamic time warping and kinect: A practical approach”, in 2017 International Conference on Information Systems and Computer Science (INCISCOS), IEEE. (2017), 302-308.
6. Mitra S, Acharya T, “Gesture recognition: A survey”, IEEE Transactions on Systems, Man, and Cybernetics, Part C, Vol. 37, No. 3, (2007), 311-324. DOI: 10.1109/TSMCC.2007.893280.
7. Pisharady PK, Saerbeck M, “Recent methods and databases in vision-based hand gesture recognition: A review”, Computer Vision and Image Understanding. Vol. 141, (2015), 152-165. DOI: 10.1016/j.cviu.2015.08.004.
8. Yang MH, Ahuja N, Tabb M, “Extraction of 2d motion trajectories and its application to hand gesture recognition”, IEEE Transactions on pattern analysis and machine intelligence, Vol.24, No. 8, (2002),1061-1074. DOI: 10.1109/TPAMI.2002.1023803
9. Elsoud AS, Elnaser OA, “LVQ for hand gesture recognition based on DCT and projection features”, Journal of Electrical Engineering, Vol. 60, No. 4, (2009), 204-208.
10. Oyedotun OK, Khashman A, “Deep learning in vision-based static hand gesture recognition”, Neural Computing and Applications, Vol. 2, No.12, (2017), 3941-3951. DOI: 10.1007/s00521-016-2294-8
11. Huang DY, Hu WC, Chang SH, “Gabor filter-based hand-pose angle estimation for hand gesture recognition under varying illumination”, Expert Systems with Applications, Vol. 38, No.5, (2011), 6031-6042. DOI: 10.1016/j.eswa.2010.11.016.
12. Otiniano-Rodrıguez KC, Cámara-Chávez G, Menotti D, “Hu and Zernike moments for sign language recognition”, in Proceedings of international conference on image processing, computer vision, and pattern recognition, (2012),1-5.
13. Van den Bergh M, Carton D, De Nijs R, Mitsou N, Landsiedel C, Kuehnlenz K, Wollherr D, Van Gool L, Buss M, “Real-time 3D hand gesture interaction with a robot for understanding directions from humans”, in 2011 Ro-Man, IEEE.(2011), 357-362.

14. Triesch J, Von Der Malsburg C, "A system for person-independent hand posture recognition against complex backgrounds", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 23, No. 12, (2001), 1449-53. DOI: 10.1109/34.977568
15. Ge SS, Yang Y, Lee TH, "Hand gesture recognition and tracking based on distributed locally linear embedding", Image and Vision Computing, Vol. 26, No. 12, (2008), 1607-1620. DOI: 10.1016/j.imavis.2008.03.004.
16. Tompson J, Stein M, Lecun Y, Perlin K, "Real-time continuous pose recovery of human hands using convolutional networks", ACM Transactions on Graphics (ToG), Vol. 33, No. 5, (2014), 1-10. DOI: 10.1145/2629500.
17. Chevtchenko SF, Vale RF, Macario V, Cordeiro FR, "A convolutional neural network with feature fusion for real-time hand posture recognition", Applied Soft Computing, Vol. 73, (2018),748-766. DOI: 10.1016/j.asoc.2018.09.010
18. Ranga V, Yadav N, Garg P, "American sign language fingerspelling using hybrid discrete wavelet transform-gabor filter and convolutional neural network", Journal of Engineering Science and Technology, Vol.13, No. 9, (2018), 2655-2669.
19. Suzuki S, "Topological structural analysis of digitized binary images by border following, Computer Vision, Graphics, and Image Processing, Vol. 30, No. 1, (1985), 32-46. DOI: 10.1016/0734-189X(85)90016-7
20. Ramer U, "An iterative procedure for the polygonal approximation of plane curves", Computer Graphics and Image Processing, Vol. 1, No. 3, (1972), 244-256. DOI: 10.1016/S0146-664X(72)80017-0
21. Erol A, Bebis G, Nicolescu M, Boyle RD, Twombly X, "Vision-based hand pose estimation: A review", Computer Vision and Image Understanding, Vol. 108, No. 1-2, (2007), 52-73. DOI: 10.1016/j.cviu.2006.10.012.
22. LeCun Y, Bottou L, Bengio Y, Haffner P, "Gradient-based learning applied to document recognition", Proceedings of the IEEE, Vol. 86, No. 11, (1998), 2278-2324. DOI: 10.1109/5.726791
23. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z, "Rethinking the inception architecture for computer vision", in Proceedings of the IEEE conference on computer vision and pattern recognition, (2016), 2818-2826.
24. Simonyan K, Zisserman A, "Very deep convolutional networks for large-scale image recognition", in Proceedings of ICLR 2015, (2015), 1-14. DOI: 10.48550/arXiv.1409.1556
25. He K, Zhang X, Ren S, Sun J, "Deep residual learning for image recognition", in Proceedings of the IEEE conference on computer vision and pattern recognition, (2016), 770-778.
26. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ, "Densely connected convolutional networks", in proceedings of the IEEE conference on computer vision and pattern recognition, (2017), 4700-4708.
27. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A, "Going deeper with convolutions", in proceedings of the IEEE conference on computer vision and pattern recognition, (2015), 1-9.
28. Gonzalez RC, Digital image processing. India: Pearson Education, 2009.
29. Szeliski R, Computer vision: algorithms and applications. USA: Springer Nature, 2022.
30. Barczak AL, Reyes NH, Abastillas M, Piccio A, Susnjak T, "A new 2D static hand gesture colour image dataset for ASL gestures", Research Letters in the Information and Mathematical Sciences, Vol. 15, (2011), 12-20.
31. Lei Y, Yuan W, Wang H, Wenhu Y, Bo W, "A skin segmentation algorithm based on stacked autoencoders", IEEE Transactions on Multimedia, Vol. 19, No. 4, (2016), 740-749. DOI: 10.1109/TMM.2016.2638204.
32. Zuo H, Fan H, Blasch E, Ling H, "Combining convolutional and recurrent neural networks for human skin detection", IEEE Signal Processing Letters, Vol. 24, No. 3, (2017), 289-293. DOI: 10.1109/LSP.2017.2654803
33. Nie S, Zheng M, Ji Q, "The deep regression bayesian network and its applications: Probabilistic deep learning for computer vision", IEEE Signal Processing Magazine, Vol. 35, No. 1, (2018), 101-11. DOI: 10.1109/MSP.2017.2763440