ASTES

Advances in Science, Technology and Engineering Systems Journal Vol. 5, No. 5, 282-287 (2020)

www.astesj.com

ASTESJ

ISSN: 2415-6698

BISINDO (*Bahasa Isyarat Indonesia*) Sign Language Recognition Using CNN and LSTM

Andi Aljabar*, Suharjito

Department of Computer Sciences, Bina Nusantara University, Jakarta, 11480, Indonesia

ARTICLE INFO

Article history:
Received: 16 May, 2020
Accepted: 07 July, 2020
Online: 17 September, 2020

Keywords:
Sign Language
BISINDO
CNN
RNN
LSTM

ABSTRACT

Sign language is one of the languages which are used to communicate with deaf people. By using it, they can communicate and understand each other. In Indonesia, there are two standards of sign language which are SIBI (Sistem Bahasa Isyarat) and BISINDO (Bahasa Isyarat Indonesia). Deep learning is a model that is used to apply to this topic. In this model, there are a lot of methods such as convolutional neural network, recurrent neural network, long-sort term memory, and each model has its characteristics. There are also some issues in deep learning by sign language recognition as the object such as data training, object position, pose, lighting, and the background of objects. This research will describe how to combine background subtraction and gaussian blur pre-processing, forwarding pre-processing background subtraction with CNN by using BISINDO, LSTM, and a combination between CNN and LSTM. In conclusion, this research shows that a combination between CNN and LSTM is the best model by explaining the accuracy and testing with sign language BISINDO as the object. The accuracy showed that for CNN 96%, LSTM 86%, and combination CNN and LSTM 96%, and the loss showed that for CNN 18%, LSTM 41%, and combination CNN and LSTM 17%.

1. Introduction

In recent years computer vision has been developed very rapidly, starting from its use in the robotic field, human interaction with computers, authentication of iris and fingerprints, face detection, and more. One popular topic at the moment is Sign Language Recognition (SLR). Sign language is a language that is used with deaf people communities to communicate with each other. In Indonesia, there are two standards of sign language, they are called SIBI (*Sistem Bahasa Isyarat Indonesia*) and BISINDO (*Bahasa Isayarat Indonesia*). There are many differences between SIBI and BISINDO, one of them was adopted from ASL (American Sign Language), this one calls SIBI [1]. However, both SIBI and BISINDO are still used in Indonesia. Nevertheless, SIBI has been approved by the government of Indonesia, and SIBI is used in schools and for studying, but most of the deaf people in Indonesia use BISINDO in their life activities more than SIBI [2].

Moreover, some research already studied this topic such as Leap Motion Controller (LMC), and HMM (Hidden Markov Model) vision base approach dan Microsoft Kinect dataset [3], [4]. For example, by using HMM and BISINDO object, the experiment got around 60% of accuracy [5]. It is because of how complex this system is. It is not only because of the method and model but also some aspects such as preprocessing. In preprocessing has some methods, one of them which will be experimented in this research is background subtraction and gaussian blur. One technique that will be used is how the system can distinguish the object's hand and the background.

Nevertheless, there are some issues about this topic in that data such as background image (data), lighting, and others [6]. As mentioned before, preprocessing is one of the important steps before the data entering the model. This research will use a black background. It will help the system to read the object easier than using a random background. Light and space between an object and camera are also important, which will influence the vectors matrix on the model and will affect the result.

In the earlier research about this machine learning and deep learning, the researcher using Generalized Learning Vector Quantization (GLVQ) and Kinnect as a dataset [2], [7]. As a rapid computer vision technology, especially for this topic, they will be applied with sign language. This topic has a lot increasing in its sectors. Such as Deep learning by using CNN (Convolutional Neural Network) and RNN (Recurrent Neural Network), they are

*Corresponding Author: Andi Aljabar, Bina Nusantara University, Jakarta, Tel: +62853-42622246 Email: andi.aljabar@binus.ac.id

used to manage image or video which is extracted in the frame than will translate the object to the text. The accuracy and loss of each own models are very dependent on some variables such as filter, pixel, and layers that are used. Filter, pixel, and layers are also a differentiator between the previous models such as Lipnet, Resnet, VGGnet, and others

In addition, All models have the same way to predict the sign [8]. Firstly, by preprocessing the data to be vectors matrix than doing processing by using the model, for the last the model giving the accuracy of recognition by doing training and validation to the data. In the pattern, the data must be divided into three parts, training, validation, and testing. The data training and validation will be used to make the model and data testing will be used to get the ability of the model that was created. The input from recognition sign language is an image or video (the combination of several images), the data processing requires a large bandwidth or low latency.

Furthermore, there are many types of researches that have each own positive and negative impact on this topic. According to some references by using deep learning in the neural network, this research will increase the accuracy and compare models CNN, LSTM, and a combination of CNN and LSTM.

2. Literature Review

2.1. Preprocessing

Preprocessing is the way to make data to be good to train. It means, preprocessing is a technique to improve the quality of image to remove the obstruction of the image and others [9]. *Preprocessing* is also used to smoothing the images for low frequencies. And also use to convert the image to the color that the model needed [10].

On the other hand, the mainframe of *preprocessing* is how the data can be normalized than that can be training as well to give the best result in accuracy [11]. Even it is text or image data. *Background subtraction* is one of the methods to subtract the image data in the *preprocessing* method. These three components in *background subtraction*, according to the color standard which is the RGB (Red, Green, Blue) [12]. By reducing and increasing the value of each own RGB, this can get the image result of *preprocessing*.

After *preprocessing*, data already normalize. Next, how the data will be matching with the models. In this research is using CNN and LSTM model to manage the data then get the result.

2.2. CNN (Convolutional Neural Network)

The convolutional neural network is a model that is used to process object recognition. In 2012, CNN is becoming a model that really important to support object recognition [13]. CNN also works well to do adaptive multi-modal and shows it's a great power on image recognition [13].

CNN is spreading the data to the layer frames. According to the upgrading, CNN has many changes such as VGG16, ResNet. Nowadays, 3D-CNN is a hot topic to study about. As mentioned before, 3D-CNN has become the model that can process 3D data

Figure 1 shows how CNN works [14]. The data will be converted to the layers which consist of *max pooling* and *fully connected* layers. It is also using high-resolution layers and low-resolution layers, it is depending on the dataset.

2.3. LSTM (Long Short-term Memory)

LSTM is a method that is used to process data sequentially and was developed by RNN. In LSTM, there is a new module which is called gates. That components are the input gate, neuron recurrent connection, forget gate, and output gate [15].

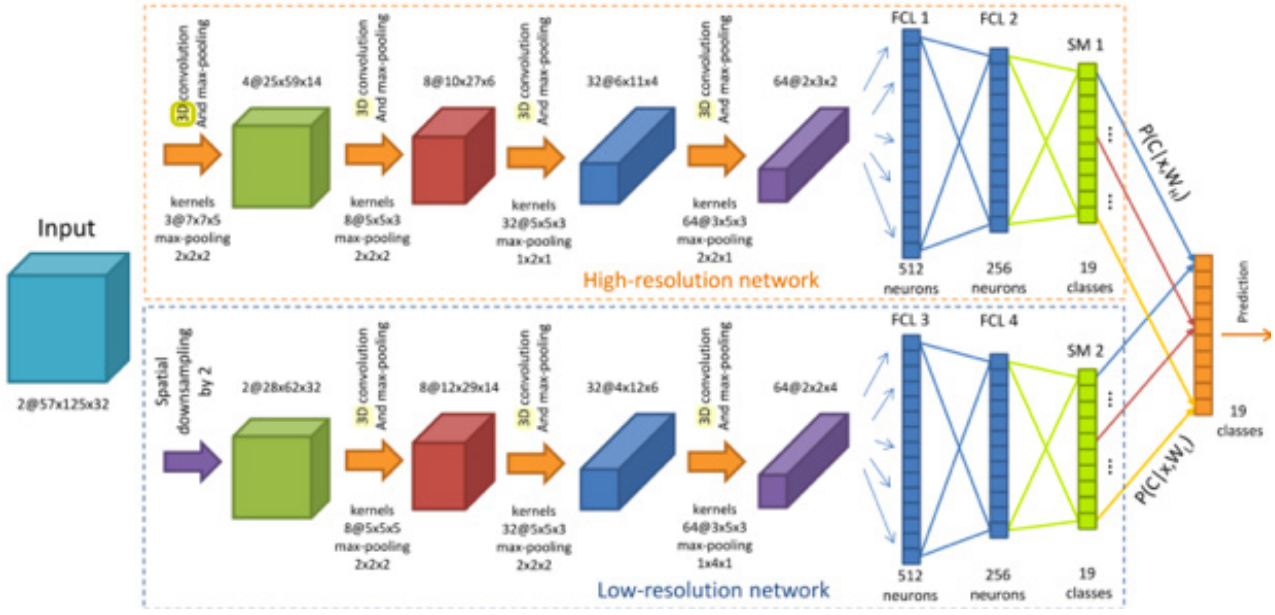


Figure 1: CNN Structure

RNN is used to the real sequential data which depends on time series data [7]. Both RNN and LSTM are parts of DNN [16]. It means DNN is increasing itself. Even the data is already learned, by using LSTM, the data will also be trained because of the sequential feature that it has. Therefore, LSTM is powerful to manage continuous recognition tasks. Figure 2 will show you how the LSTM works and the gate structure of it [17]

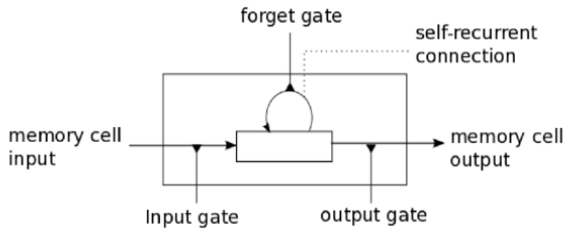


Figure 2: LSTM Structure

As shown in figure 2, the data will be stored in the memory cell. The data which is indicated as noise will be sent to the forget gate and the other data will continue to the self-recurrent. Furthermore, the forget gate will process the data again to the self recurrent connection than process the other data and the data from the forget gate to mark as the real data than send to the memory cell output to do the next steps.

In addition, it means to process image data there are three steps which are *preprocessing*, modeling, and testing model.

3. Methodology

Firstly, doing study literature about sign language recognition, then determine the background study and point of topic research. The next step was collecting the dataset which became the data training. The next step was doing *preprocessing* the data that was collected. To get a new result, the model was doing some interflow of layers in CNN, LSTM, and some combinations between CNN and LSTM. The last step was writing the paper and the result of this research.

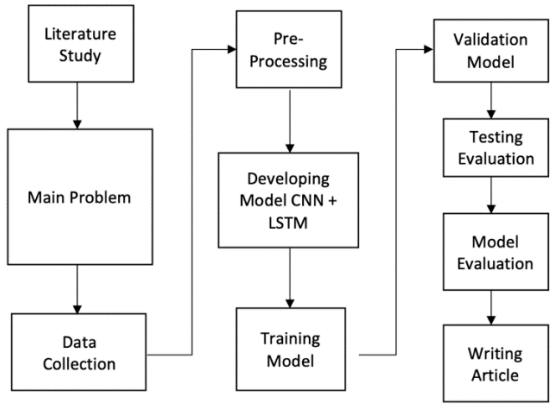


Figure 3: Research Method

As mentioned in [3] by using 9 frames in CNN than combine it with 1024 cells in LSTM showed the result accuracy and loss of CNN, LSTM, and combination CNN and LSTM, Especially for CNN used two types of model extraction, start from high to low and low to high. By using HMM (Hidden Markov Model), object BISINDO and both male and female as samples could get a 60-70% accuracy result [17]. Another research used four models of LipNet, 3 blocks of 3D-CNN. The first model was using 3 blocks of 3D-CNN, then the second was using one block of 3D-CNN, the third model was using eight blocks of CNN and last was using 2 blocks of B-RNN by using SIBI as objects. The result was by calculating the average of WER (Word Error Rate) equal to 88,79% and CER (Character Error Rate) equal to 65.33% [5].

Table 1: Research Comparison

Authors	Method & Dataset	Accuracy Result
[5]	HMM & BISINDO	accuracy 60-70%
[7]	CNN + B-RNN & SIBI	Average of word error rate 88.79%, average of character error rate 65.33%
[18]	C3D-CNN & EgoGesture	accuracy EgoGesture 91.04%, nVGesture 77.,39%
[19]	C3D-CNN & 3D dynamic skeletal data	accuracy 78%
[8]	CNN + LSTM & Chinese Sign Language	accuracy 96.52%
[20]	LSTM & Kinnect	accuracy 63.3%
[4]	CNN + LSTM & American Sign Language	Softmax Accuracy 91,5%, Pool Layer Accucary 56,5%

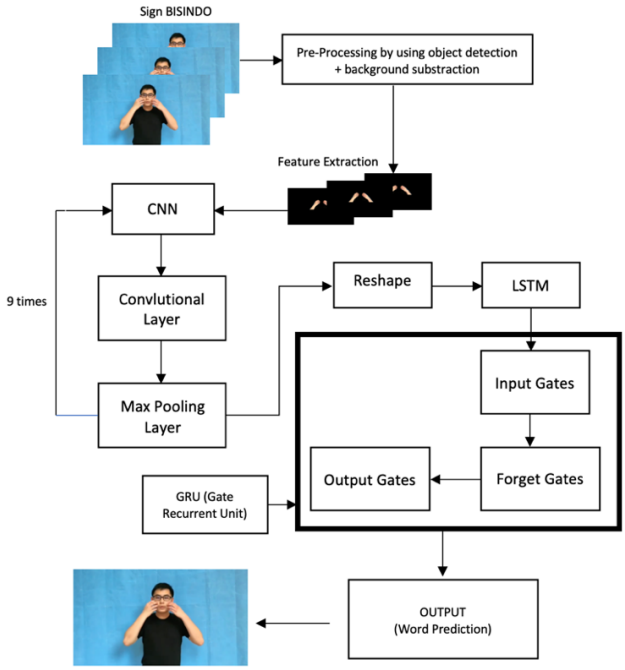
According to table 1, CNN, and LSTM were the best combinations to sign language by using such as kind of object sign language. Besides, by using HMM with BISINDO object had a high of error rating [17]. The result was used different dataset and still had a good accuracy by each own paper. By comparing the result of CNN, LSTM, and combination CNN and LSTM that this research showed the result of a good method for BISINDO object

3.1. Proposed Method

Before *preprocessing*, the data was created by 2 alphabets and 8 words of BISINDO (*A, Berapa (how many/much), Kamu (you), L, Nama (name), Sama-sama (your welcome), Saya (I), Sayang (love), Terimakasih (thank you), and Umur(age)*). After that normalizing the data. On the other hand, Inside the purpose model, the process was doing training, validation, and testing according to the dataset from *background subtraction*. The dataset became an input than did mapping from the real data. The data was converted to another kind of data. Pixel of data or images was transferred to the matrix number than gave the output data to recognize the gesture.

As imaged in fig. 4, it told how the rule from the model was used. The first, if the data is ready that will do *preprocessing* and

that *preprocessing* will produce all the images in black and white pictures. After that, the model was doing training data by using CNN and LSTM. On the other hand, after training, the model did the next process to produce hidden layers of CNN, refer to how many layers that were creating. Next steps, the output from CNN became processed in LSTM. According to LSTM, the data stored in the same gates of LSTM. The last step was doing validation data by doing *max pooling* and *fully connected* layer between both CNN and LSTM.



There are types of data, data training, and data validation. The model needs that because of the theory of validation. The model can validate the data if there is another data similar to datasets. Validation data is not an implementation mode, it is part of the model [21]. In the last, after getting the result of *preprocessing*, training and validation showed how the model worked together for implementation.

3.2. Dataset and Preprocessing

In preprocessing. The data was collected by using a camera around three feet and it has 720p HD. The camera recorded the object and direct it to save the object as the dataset by using the preprocessing method (*background subtraction and gaussian blur*). Data size was 100 x 89 pixels of each own image. Every single object had 1000 for training and 100 for validation. For the testing used a video for each object.

Figure 5 explains how the dataset was recorded and converted it to grayscale images by using *background subtraction.* However, preprocessing used *0.5 weight* and 7.0 gaussian blur. The first step is *preprocessing* using *background subtraction*. The videos or dataset converted to the gray object (hand object). After that made all the dataset in the same value, to do that, converting the image scale to the size 100x90 pixel. The dataset for training was 1000 images for each own object. And the data validation was 100 images for each own object. Figure 7 shows how the image became the vectors matrix.

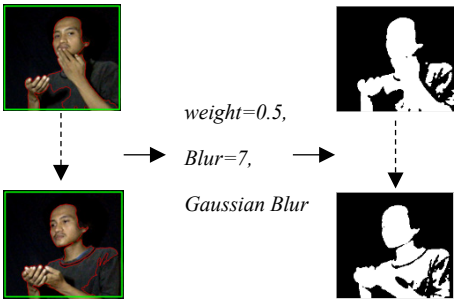


Figure 5: Dataset Recording

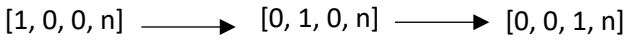


Figure 6: Image to vectors array

3.3. CNN (Convolutional Neural Network)

After *preprocessing*, the image had the same value capacity. After that converted again by translating the image gray to the array matrix. When the image pixel is found in white it will be given value 1 on the value matrix and if the image found black it will give value pixel as 0 and also the weight of data was 0.5. It showed in figure 6. Then, CNN processed the vectors matrix.

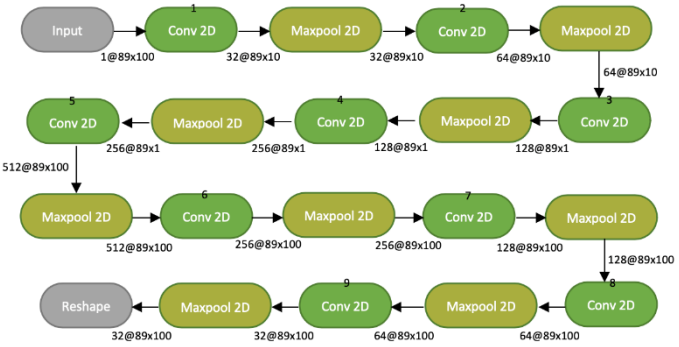


Figure 7: CNN Model

This research used the CNN model. Furthermore, this used 9 or more layers of CNN to training and validating the data. Figure 8 told how CNN worked with the matrix. The matrix processed in CNN using hidden layers. As mentioned before this used 9 hidden layers or more. Also, CNN 32 filters to 512 pixels filters of layers in the low to high then using 512 pixels to 32 in the high to low. CNN also used 89x100 pixels of each own layer including hidden layers. For each method was using 30 *epocs*.

In the CNN model after the last *maxpooling* 2D by 32@100x89 pixel, it used fully connected layers to produce the model of CNN. Therefore, in combination between CNN and LSTM using end to end training and validation, before the data entered the LSTM model, firstly it was reshaped then processed to the next step.

3.4. LSTM (Long Short-term Memory)

Figure 9 shown how LSTM worked with its gate. Next steps, the output from CNN processed more by using the LSTM model which had 1024 of cells. During the training process, LSTM or DNN model kept the data, then calculated all the data training. If the cache training did not clean the LSTM kept it in memory.

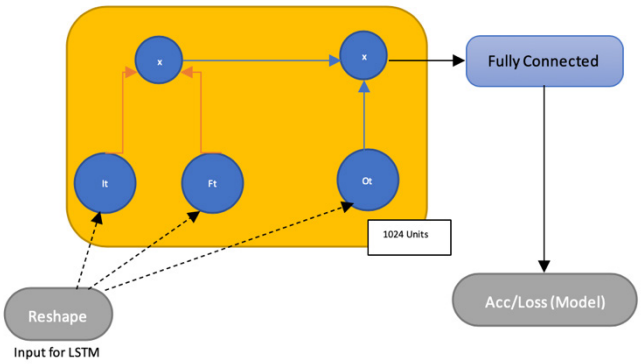


Figure 8: LSTM Model

LSTM worked with a sequential segment. The data sent to the gate. If the date had some noise than the data was sent to the forget gate. If the data was an input then the data stored to the cell. Before *max pooling* and *fully connected* layers, the LSTM system will call forget gate, afterward training the data more. After that give the result.

This research was used processor intel core i7 with 6 core of CPU, AMD Radeon R9 M370X of GPU, and 16GB for ram. The Training and validation took for CNN took more than 2 hours, LSTM took more than 3 hours, and the combination between CNN and LSTM took more than 3 hours.

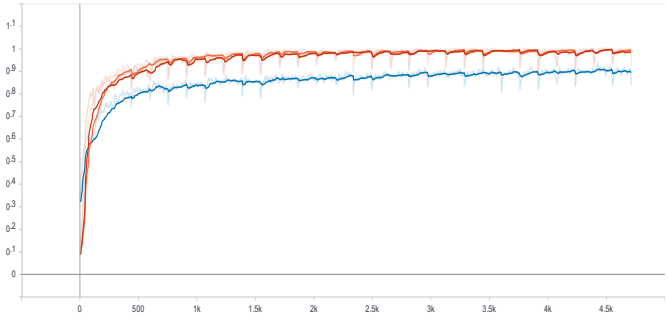


Figure 9: Training accuracy CNN, LSTM and CNNLSTM

4. Results

According to figure 10, orange for CNN, blue for LSTM, and red for CNNLSTM. The lowest score of accuracy training model is LSTM, and CNN also CNNLSTM almost the same.

Table 2: Accuracy and Loss Training

Medel	Acc	Loss
CNN	96%	18%
LSTM	86%	41%
CNN + LSTM	96%	17%

As shown in table 2, The lowest score of loss training model is CNNLSTM a point in the last result is 17% of loss, CNN has 18% of loss and LSTM has 41%. On the accuracy. CNN and LSTM have the same score which is 96%.

Table 3: Testing Result

Objek	CNN	LSTM	CNN LSTM
A	99%	98%	99%
Berapa (how much/many)	90%	75%	94%
Kamu (you)	99%	99%	99%
L	90%	94%	90%
Nama (name)	83%	66%	89%
Sama-sama (your welcome)	0%	75%	89%
Saya (I)	87%	73%	91%
Sayang (love)	88%	87%	88%
Terima Kasih (thank you)	92%	75%	90%
Umur (age)	0%	67%	71%
AVERAGE	73%	81%	90%

Table 3 shown the testing result of each own data object model CNN, LSTM, and CNN+LSTM. CNN got 73%, for LSTM got 81%, and combination CNN+LSTM got 90% of each own average.

5. Conclusion

With the rapid development in machine learning and deep learning, is expected to facilitate life. One of the benefits that develop a deep learning method to do hand predictions in sign language. By using the CNN, LSTM, and CNN + LSTM methods and focus on filtering, layers, and BISINDO object. According to the training, validation, and testing model, the best model to use in this object is CNN+LSTM. It got 96% for accuracy, 17% for loss, and 90% for testing.

Although this paper point explains the deferential between CNN, LSTM, and CNN+LSTM models, the dataset needs to add for all symbolic in sign language. For the future, it is also good to focus on the other part of sign languages such as expression detection, body gesture detection or mouth detection.

References

[1] M.A. Kumbhar, D.P. Bhaskar, "A Review on Motion Detection Techniques," International Journal of Trend in Scientific Research and Development, **-2**(1), 736–740, 2017, doi:10.31142/ijtsrd5928.

[2] R.A. Mursita, "Respon Tunarungu Terhadap Penggunaan Sistem Bahasa Isyarat Indonesa (SIBI) dan Bahasa Isyarat Indonesia (BISINDO) dalam Komunikasi," Inklusi, **2**, 221–232, 2015.

[3] R.K. Deepak, "Hand gesture recognition using Kinect," International Journal on Future Revolution in Computer Science & Communication Engineering, **4**(6), 59–62, 2012, doi:10.1109/icsess.2012.6269439.

[4] K. Bantupalli, Y. Xie, "American Sign Language Recognition using Deep Learning and Computer Vision," Proceedings - 2018 IEEE International Conference on Big Data, Big Data 2018, 4896–4899, 2018, doi:10.1109/BigData.2018.8622141.

[5] T. Handhika, R.I.M. Zen, D.P. Lestari, I. Sari, "Gesture recognition for Indonesian Sign Language," In Journal of Physics: Conf. Series, **1028**, 1–8, 2018, doi:10.1088/1742-6596/1028/1/012173.

[6] Y. Zhang, C. Cao, J. Cheng, H. Lu, "EgoGesture: A New Dataset and

Benchmark for Egocentric Hand Gesture Recognition," IEEE Transactions on Multimedia, **20**(5), 1038-1050_, 2018, doi:10.1109/TMM.2018.2808769.

[7] M.C. Ariesta, F. Wiryana, Suharjito, A. Zahra, "Sentence Level Indonesian Sign Language Recognition Using 3D Convolutional Neural Network and Bidirectional Recurrent Neural Network," 2018 Indonesian Association for Pattern Recognition International Conference (INAPR), 16–22, 2018, doi:10.1109/INAPR.2018.8627016.

[8] S. Yang, Q. Zhu, "Continuous Chinese sign language recognition with CNN-LSTM," Ninth International Conference on Digital Image Processing (ICDIP 2017), **10420**(100), 104200F, 2017, doi:10.1117/12.2281671.

[9] S. Perumal, T. Velmurugan, "Preprocessing by contrast enhancement techniques for medical images," International Journal of Pure and Applied Mathematics, **118**(18), 3681–3688, 2018.

[10] Y. Alginahi, "Preprocessing Techniques in Character Recognition," Intech, **10**, 1–21, 2010, doi:10.5772/9776.

[11] G. Hima Bindu, P.V.G.D. Prasad Reddy, M. Ramakrishna Murty, "Image preprocessing of abdominal CT scan to improve visibility of any lesions in kidneys," Journal of Theoretical and Applied Information Technology, **96**(8), 2298–2306, 2018.

[12] A. Solichin, A. Harjoko, "Metode Background Subtraction untuk Deteksi Obyek Pejalan Kaki pada Lingkungan Statis," Seminar Nasional Teknologi Informasi 2013, 1–6, 2013.

[13] A. Karambakhsh, A. Kamel, B. Sheng, P. Li, P. Yang, D.D. Feng, "Deep gesture interaction for augmented anatomy learning," International Journal of Information Management, **45**(October 2017), 328–336, 2019, doi:10.1016/j.ijinfomgt.2018.03.004.

[14] P. Molchanov, S. Gupta, K. Kim, J. Kautz, "Hand gesture recognition with 3D convolutional neural networks," In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 1–7, 2015, doi:10.1109/CVPRW.2015.7301342.

[15] T. Katte, "Recurrent Neural Network and its Various Architecture Types," International Journal of Research and Scientific Innovation, **V**(III), 124–129, 2018.

[16] D. Avola, M. Bernardi, L. Cinque, G.L. Foresti, C. Massaroni, "Exploiting Recurrent Neural Networks and Leap Motion Controller for the Recognition of Sign Language and Semaphoric Hand Gestures," IEEE Transactions on Multimedia, **21**(1), 234–245, 2019, doi:10.1109/TMM.2018.2856094.

[17] F. Beşer, M.A. Kizrak, B. Bolat, T. Yildirim, "Recognition of Sign Language using Capsule Networks," In 2018 26th Signal Processing and Communications Applications Conference (SIU), 1–4, doi:10.1109/SIU.2018.8404385.

[18] O. Köpüklü, A. Gunduz, N. Kose, G. Rigoll, "Real-time Hand Gesture Detection and Classification Using Convolutional Neural Networks," In 2019 14th IEEE International Conference on Automatic Face & Gesture Recognition, 1–8, 2019.

[19] C.R. Naguri, R.C. Bunescu, "Recognition of dynamic hand gestures from 3D motion data using LSTM and CNN architectures," Proceedings - 16th IEEE International Conference on Machine Learning and Applications, ICMLA 2017, 1130–1133, 2017, doi:10.1109/ICMLA.2017.00013.

[20] L. Tao, W. Zhou, L. Houqiang, "SIGN LANGUAGE RECOGNITION WITH LONG SHORT-TERM MEMORY," In 2016 IEEE International Conference on Image Processing (ICIP), 2871–2875, 2016, doi:10.1142/S0218127407017628.

[21] G. Devineau, W. Xi, F. Moutarde, J. Yang, G. Devineau, W. Xi, F. Moutarde, J. Yang, D. Learning, G. Devineau, W. Xi, F. Moutarde, J. Yang, "Deep Learning for Hand Gesture Recognition on Skeletal Data," In 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018), 106–113, 2018.