IMPLEMENTATION OF HAND RECOGNITION USING CONVOLUTIONAL NEURALNETWORK FOR SIGN LANGUAGE

Submitted as partial Fulfillment of the Requirements for Getting Bachelor Degree of Informatics Department Faculty of Communication and Informatics

By:

RIAN RAHMAN
L200164014

DEPARTEMEN OF INFORMATICS

FACULTY OF COMMUNICATION AND INFORMATICS

UNIVERSITAS MUHAMMADIYAH SURAKARTA

2023

HALAMAN PERSETUJUAN

IMPLEMENTATION OF HAND RECOGNITION USING CONVOLUTIONAL NEURAL NETWORK FOR SIGN LANGUAGE

PUBLIKASI ILMIAH

oleh:

<u>RIAN RAHMAN</u>

<u>L200164014</u>

Telah diperiksa dan disetujui untuk diuji
oleh:Dosen Pembimbing

acc siding pendadaran
12-01-2023

<u>Aris Rakhmadi, S.T., M.Eng.</u>

NIK 983

HALAMAN PENGESAHAN

IMPLEMENTATION OF HAND RECOGNITION USING CONVOLUTIONAL NEURAL NETWORK FOR SIGN LANGUAGE

OLEH
RIAN RAHMAN
L200164014

Telah dipertahankan di depan Dewan Penguji
Fakultas Komunikasi dan Informatika
Universitas Muhammadiyah Surakarta
Pada hari Sabtu, 14 Januari 2023
dan dinyatakan telah memenuhi syarat

Dewan Penguji:

1. **Aris Rakhmadi, S.T., M.Eng.** (……………….)
   (Ketua Dewan Penguji)
2. **Ir. Bana Handaga, M.T., Ph.D.** (……………….)
   (Anggota I Dewan Penguji)
3. **Heru Supriyono, S.T., M.Sc., Ph.D.** (……………….)
   (Anggota II Dewan Penguji)

Dekan Fakultas Komunikasi dan Informatika	Ketua Program Studi Informatika
Nurgiyatna, S.T., M.Sc., Ph.D. NIK 881	Dedi Gunawan, S.T., M.Sc., Ph.D. NIK 1305

ii

PERNYATAAN

Dengan ini saya menyatakan bahwa dalam naskah publikasi ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu perguruan tinggi dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan orang lain, kecuali secara tertulis diacu dalam naskah dan disebutkan dalam daftar pustaka.

Apabila kelak terbukti ada ketidak benaran dalam pernyataan saya di atas, maka akan saya pertanggungjawabkan sepenuhnya.

Surakarta, 13 februari 2023

Penulis

<u>RIAN RAHMAN</u>

L200164014

IMPLEMENTATION OF HAND RECOGNITION USING CONVOLUTIONAL NEURAL NETWORK FOR SIGN LANGUAGE

Abstrak

Tunarungu dan tunawicara sangat sulit berkomunikasi dengan cara berbicara. Mereka menggunakan bahasa isyarat dalam berkomunikasi terhadap sesama maupun masyarakat umum. Penggunaan bahasa isyarat yang digunakan adalah isyarat tangan sebagai media komunikasi. Indonesia memiliki dua Bahasa isyarat yaitu Bahasa Isyarat Indonesia (BISINDO) dan Sistem Isyarat Indonesia (SIBI). Perkembangan teknologi membuat kamus sebagai sarana menerjemahkan gesture tangan. Teknologi machine learning membuat komputer dapat membacaca gesture tangan hanya dari gambar. Untuk membaca isyarat tangan dengan memanfaatkan *hand gesture recognition*, yaitu teknologi yang dapat mengartikan gerakan tangan manusia menggunakan algoritma matematika. Besaran ukuran resolusi dari kumpulan data dapat mempengaruhi tingkat keberhasilan. *Convolution Neural Network* (CNN) adalah metode dari *Deep Learning* yang digunakan dalam penelitian ini. Maka, dengan menggunakan CNN kecepatan dan tingkat akurasi dalam mendeteksi isyarat tangan akan lebih tinggi. Penelitian ini mengimplementasikan metode CNN terhadap 24 huruf abjad dalam SIBI sebagai input yang berupa gambar. Data input sebanyak 3468 gambar yang akan dilatih. Data uji sebanyak 720 gambar yang masing-masing abjad terdiri dari 30 gambar. Pengujian dengan data yang telah dilatih menghasilkan akurasi sebesar 96,67%.

Kata Kunci: bahasa isyarat, CNN, *deep learning*, deteksi tangan, SIBI.

Abstract

The deaf and mute are hard to communication by talking. They using sign language to communication towards the other and the general public. Sign language using hand to communication. Indonesia has two sign languages, namely *Bahasa Isyarat Indonesia* (BISINDO) and *Sistem Isyarat Indonesia* (SIBI). Technological development made a dictionary to translating a hand gesture. Machine learning is the technology that make a computer can recognize a hand from the image. Using hand gesture recognition to read gesture of a hand, that is a technology that can define gesture of human hand using mathematical algorithms. Resolution of the dataset can affect a percentages of success rates. Convolution Neural Network (CNN) is a method of Deep Learning that used in this study. Using CNN to hand gesture recognition can make the peed and the accuracy become higher. This study aboutimplementation of CNN method to 24 alphabets of SIBI as the input in image. The input data is 3468 images to be trained. The test data consists of 720 images, each alphabet consisting of 30 images. Testing with data that has been trained produces an accuracy of 96,67%.

Keywords: CNN, deep learning, hand recognition, SIBI, sign language.

1. INTRODUCTION

Communication is connection of people. Through communication people can socialize with environment. Language is a mediator of communication for interaction and activity between once and the others. There is so many languages in this world, sign language is one of nonverbal communication used by deaf and mute but deaf as the main people who used this language. Deaf and mute are hard to say words, cause of that they using sign language to talk with the others. Hand gesture utilized by sign language to communicate without talk. Every movement of finger has different value. Public need to understand about sign language if they want to talk with the deaf or mute. Many people don’t understand the meaning of sign language because in daily life they don’t use it to communication, that make them difficult to communication with the deaf and mute (Kusumawati, 2016). Indonesia has two types of sign language Sistem Isyarat Bahasa Indonesia (SIBI) and Bahasa Isyarat Indonesia (BISINDO) (Isyarat & Bisindo, 2015; Study et al., 2013; Yusnita et al., 2017).

Technological developments have biometric system that can recognize fingerprint, voice, hand palm, and hand gesture. Along with the progress of the times digital image classification is needed in various fields such as informatics, medicine, marine, agriculture, and business (Kah et al., 2015). The purpose of Image Classification is to classify images into certain categories. Image classification is currently one of the problems that have long sought a solution in computer vision One successful approach is using an Artificial Neural Network (ANN). ANN is a part of artificial intelligence that has the ability to learn from data and does not take long to build models. ANN is part of Machine Learning (ML).

Machine Learning is an artificial intelligence that aims to optimize the performance of a system by studying sample data or historical data. The type of ANN model consisting of many layers is called Multi-Layer Perceptron (MLP) which functions to fully connect neurons (Ramchoun et al., 2016). The classification technique using MLP has a weakness when the input is in the form of an image. Images must be pre-processed, segmented, and extracted to obtain optimal performance. Another development of MLP that can overcome this problem is the Convolutional Neural Network (CNN).

Convolutional Neural Network (CNN) is a Deep Learning (DL) method that is used to detect and recognize an object in a digital image. Deep Learning is one of the sub-fields of Machine Learning. Deep Learning is the implementation of the basic concepts of Machine Learning that implements the ANN algorithm with more layers. With the number of hidden layers used between the input layer and the output layer, then this network can be said to be a deep neural network Deep learning and has many library tensorflow is one of that developed by google (Bantupalli, 2018;

Wati, 2019; Li et al., 2021).

The last couple of years, Deep Learning has shown good performance. In recent years Deep Learning has shown remarkable performance with much of it due to more powerful computing factors, large data sets, andtechniques to train deeper networks. CNN has the same as the other deep learning models, but also has a weakness because the model training process is quite long. However, with the rapid development of hardware, this can be overcome by using Graphics Processing Unit (GPU) and PCs with high specifications. This study uses the deep learning method with the CNN model to help recognize and classify the SIBI alphabet. Hand gesture recognition is a part that used to detecting hand gesture of sign language. Deep learning is a part of machine learning that can help to hand gesture recognize with higher accuracy.

2. METHOD

2.1 Dataset

The research samples are hand gesture photos taken using a cellphone camera based on the official website of the Ministry of Education, Culture, Research, and Technology. The sample photo of the alphabet A-Y without J and Z because they are movements. The sample consists of 24 classes with a total of 3468 photos which will be divided into Train (80%) and Validation (20%). The SIBI alphabet is shown in Figure 1

Figure.1 Image SIBI Alphabet

2.2 Data Preprocessing

The data that has been collected is uploaded to Google Drive and entered into Google Colaboratory. The data is divided into 2 folders, namely train of 2768 (80%) and validation of 700 (20%) using the splitfolders library, each containing 24 classes. The augmentation data uses Image Data

Generator from the keras library with parameters on the train data:

rescale= 1./255,

shear_range= 0.2,

horizontal_flip= True. Validation data rescale= 1./255.

After augmentation, labeling is carried out where each class with the same image will be given the same label. Labeling uses *flow_from_directory()*, the system will take the directory path to read images from numpy array according to the labels we have created with parameters *batch_size=16*, *shuffle=True*, and *input_image=(64x64)*. Data augmentation is carried out namely, the images will be grouped as many as 16 images per group, the images will be randomized and the images will be resized to 64x64 pixels.

2.3 CNN Model

The Convolutional Neural Network model used has 2 layers which can be seen in Figure.2. The input shape is 64x64 which has 3 color layers or 'RGB'. Pooling layers are used to reduce the dimensions of a feature map, specifically width, height, and depth. Max pooling returns the maximum value of elements in the image portion covered by the filter. Max pooling is more precise at extracting dominant features and is therefore considered more performant.

Layer (type)	Output Shape	Param #
conv2d_7 (Conv2D)	(None, 62, 62, 32)	896
max_pooling2d_7 (MaxPooling2D)	(None, 31, 31, 32)	0
conv2d_8 (Conv2D)	(None, 29, 29, 64)	18496
max_pooling2d_8 (MaxPooling2D)	(None, 14, 14, 64)	0
flatten_2 (Flatten)	(None, 12544)	0
dense_4 (Dense)	(None, 128)	1605760
dense_5 (Dense)	(None, 24)	3096

Total params: 1,628,248
Trainable params: 1,628,248
Non-trainable params: 0

Figure.2 CNN model

Each block contains a max pooling and 128 filters in the last fully connected layer activated by the ReLu activation function. To avoid overfitting the model using Dropout. Dropout is a regularization technique in CNN that breaks the interdependence between neurons. This interdependence can lead to overfitting of the data. Overfitting can lead to poor predictions in data sets.

The Flatten layer used in the CNN model is used to transform the final feature into a single vector or one-dimension. The flattening step is needed so fully connected layers can be used after some convolutional/max pool layers. In this last layer, we use the softmax activation function. Softmax is an activation function that scales numbers/logits into probabilities. The output of a Softmax is a vector with probabilities of each possible outcome. The probabilities in vector sums to one for all possible outcomes or classes. CNN are trained for 173 steps per epoch, using a size of 16 batches, and 20 epochs. An epoch is when the entire data set actually goes through the training data set. The number of epochs and steps per epoch has an influence on the level of accuracy and loss. So, a greater number of epochs and steps per epoch can produce a good level of accuracy and alow level of loss but requires a longer time (Yuliani et al., 2019).

The training will be held twice. The first uses the CPU and the second uses the GPU. The results of accuracy, loss, and training time will be compared. This study conducts training using Google Colaboratory with the hardware specifications used in this study:

CPU : Intel(R) Xeon(R) CPU @ 2.00GHz (1 Core, 2 Threat).

GPU : Nvidia Tesla T4, VRAM 16 GB GDDR6.

RAM : 12.6 GB.

Disk : 78.2 GB.

3. RESULT AND DISCUSSION

3.1 Training and Validation Accuracy

Training was carried out twice with data on 3468 images divided into 16 batches. The first training uses a CPU has 1 hour 39 minutes 55 seconds. The output of training uses a CPU generated produces accuracy and loss, a comparison of the two can be seen in Figure.3. In the first epoch, the data train produces training data with 99.13% accuracy and 0.24% loss and data validation with 96.00% accuracy and 20.43% loss.

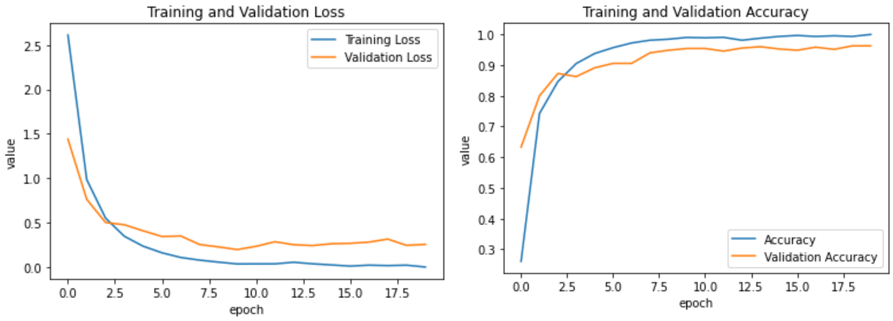

Figure.3 Graph of accuracy and loss from data train and validation using CPU

The second training uses a GPU has 1 hour 18 minutes 30 seconds. The output of training uses a GPU generated produces accuracy and loss, a comparison of the two can be seen in Figure 4. In the first epoch, the data train produces an accuracy of 36.71% and a loss of 22.59%, while data validation produces a loss of 58.00% and 14.87%. The process ends in the 20th epoch with the results of training data with 99.31% accuracy and 0.26% loss and data validation with 95.57% accuracy and 2.63% loss. The results of the CNN process are saved as model.h5 to be used in the testing process with new data.

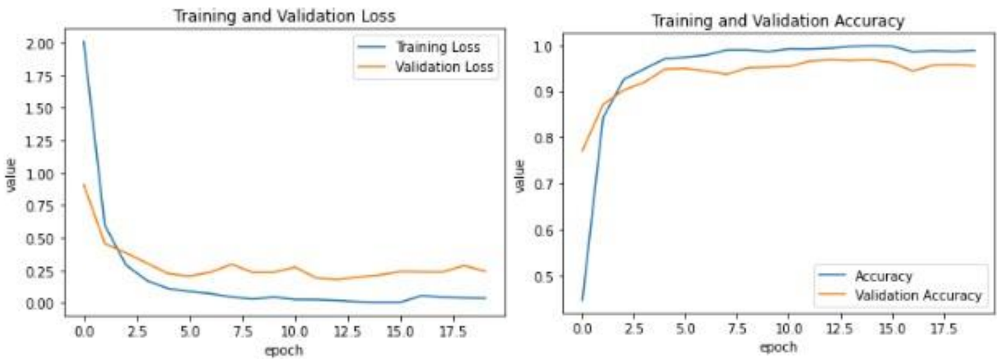

Figure.4 Graph of accuracy and loss from data train and validation using GPU.

3.2 Test Accuracy Model

The model accuracy test uses data that has never been used in training. Testing uses 720 images as the data, with 30 images for each class. The results can be seen in table.1.

Table.1 Prediction Result

Class	Correct Prediction	Incorrect Prediction	Accuracy, %
A	29	1	96,67%
B	30	0	100,00%
C	29	1	96,67%
D	29	1	96,67%
E	29	1	96,67%
F	28	2	93,33%
G	29	1	96,67%
H	28	2	93,33%
I	30	0	100,00%
K	27	3	90,00%
L	29	1	96,67%
M	28	2	93,33%
N	26	4	86,67%
O	30	0	100,00%
P	30	0	100,00%
Q	30	0	100,00%
R	30	0	100,00%
S	29	1	96,67%
T	30	0	100,00%
U	29	1	96,67%
V	29	1	96,67%
W	30	0	100,00%
X	30	0	100,00%
Y	28	2	93,33%
Total	696	24	

Prediction is done with input images of 64x64. Based on table.1 shows the precision results for each class. There are 3 results used in testing, namely correct prediction, wrong prediction, and accuracy. Correct prediction is the amount of data with prediction results that match the class name. Incorrect prediction is the amount of data with wrong results or not in accordance with the class. Accuracy is the percentage of success of the number of correct predictions. Accuracy can be determined by the formula:

$$Accracy\ (\%) = \frac{Correct\ Predicion}{Data\ in\ each\ Class} \times 100\% \tag{1}$$

Example

$$Accracy\ of\ Class\ F\ (\%) = \frac{28}{30} \times 100\% \tag{2}$$

$$Accracy\ of\ Class\ F\ (\%) = \ 93.33\%$$

The percentage of accuracy is obtained based on each class. The highest accuracy is 100% while the lowest is in class N which is 86.67%. In addition to the accuracy per class, the prediction accuracy of the entire data is also obtained. The overall accuracy is obtained by the formula:

$$Overall\ Accuracy\ (\%) = \frac{Total\ Correct\ Prediction}{Total\ Data} \times 100\% \qquad (3)$$

$$Overall\ Accuracy\ (\%) = \frac{696}{720} \times 100\%$$

$$Overall\ Accuracy\ (\%) = 96.67\%$$

The results obtained from the 64x64 model input with a test sample of 720 images have 696 correct predictions, 24 wrong predictions and get an accuracy value of 96.67%.

4. CLOSING

4.1 Conclusion

Based on the results of the research that has been done, some conclusions are obtained as follows.

1. Training is done using GPU and CPU have close accuracy results. but have different training time estimates. The GPU has a training time of 21 minutes 25 seconds which is faster than using the CPU.
2. The prediction uses new data with a total of 24 letters. The predictions made were 100% correct with the results of 24 data getting a value of more than 90% and 2 data getting a value below 89%.
3. The lowest accuracy results are in class N at 86.67%. There are 4 wrong predictions due to the similarity of N with M.

4.2 Suggestion

1. Future research is expected to use gestures in the form of movement.
2. This research can be read in the form of sentences and is carried out in real time.
3. Adding parameters such as comparison of input images with greater resolution and adding more layers to the CNN model.
4. This research can be developed in the form of an application combined with a smartphone.

REFERENCES

Bantupalli, K. (2018). American Sign Language Recognition using Deep Learning and Computer Vision. *2018 IEEE International Conference on Big Data (Big Data)*, 4896–4899. https://doi.org/10.1109/BigData.2018.8622141

Isyarat, B., & Bisindo, I. (2015). *RESPON TUNARUNGU TERHADAP PENGGUNAAN SISTEM BAHASA ISYARAT INDONESA (SIBI) DAN*. 221–232.

Kah, G., Michael, O., Connie, T., Teoh, A., & Ngo, D. (2015). *AUTOMATED HAND GEOMETRY VERIFICATION SYSTEM BASE ON SALIENT AUTOMATED HAND GEOMETRY VERIFICATION SYSTEM*. *July 2015*.

Kusumawati, T. R. I. I. (2016). *KOMUNIKASI VERBAL DAN NONVERBAL*. *6*(2).

Ramchoun, H., Amine, M., Idrissi, J., Ghanou, Y., & Ettaouil, M. (2016). Multilayer Perceptron: Architecture Optimization and Training. *International Journal of Interactive Multimedia and Artificial Intelligence*, *4*(1), 26. https://doi.org/10.9781/IJIMAI.2016.415

Study, C., Yrtrw, S., Rehabilitasi, Y., Tuna, A., & Wicara, R. (2013). *Learning Support Application of Gesture Language for Mute and Deaf*. *November*, 45–48.

Wati, V. (2019). Real Time Face Expression Classification Using Convolutional Neural Network Algorithm. *2019 International Conference on Information and Communications Technology (ICOIACT)*, *Clm*, 497–501.

Yuliani, E., Aini, A. N., & Khasanah, C. U. (2019). Perbandingan Jumlah Epoch Dan Steps Per Epoch Pada Convolutional Neural Network Untuk Meningkatkan Akurasi Dalam Klasifikasi Gambar. *Jurnal Informa: Jurnal Penelitian Dan Pengabdian Masyarakat*, *5*(3), 23–27. https://doi.org/10.46808/INFORMA.V5I3.140

Yusnita, L., Hadisukmana, N., Wahyu, R. B., Roestam, R., & Wahyu, Y. (2017). *Implementation of Real-Time Static Hand Gesture Recognition using Artificial Neural Network*.