AMERICAN SIGN LANGUAGE DETECTION WITH CONVOLUTIONAL NEURAL NETWORK

Normalisa[1], Sunarto Usna[2], Sudjiran[3]

[1]Computer Science Engineering, International University Liaison Indonesia, BSD
[2,3]Sistem Informasi, STMIK Jakarta STI&K, Jakarta Selatan, Indonesia
Email: [1]normalisa@iuli.ac.id, [2]sunartousna@gmail.com, [3]ontosenosudjiran@gmail.com

ABSTRACT. Humans are social creatures, one of the ways humans socialize is by communicating with each other. Socializing is needed for humans to be able to adapt, recognize and be able to move towards their environment. However, this is an obstacle for the deaf community, even though the presence of sign language can help to socialize, but not everyone can understand it. This study will propose the detection of sign language so that it is easy to understand, the sign language that will be detected is American Sign Language (ASL). American Sign Language is used because it is quite well known for its use in terms of sign language research and will use the Machine Learning method, namely Convolutional Neural Network (CNN). The use of this method itself was chosen because it has a high level of accuracy, in this study the results of the resulting accuracy rate were 99.89%.
Keyword: Communication, ASL, CNN, Detection, Accuracy.

1. INTRODUCTION

Communication is a crucial thing to be able to live between humans, disturbed communication will be an obstacle and even misunderstanding. The deaf community in communicating has used sign language which is a bit difficult because not everyone understands it. Therefore, this study will detect sign language into ordinary human language so that it can be easily understood.

2. REVIEW

a. Related Work

Table 1 Review

No.	Author	Title	Method	Result
[1]	I Putu Iduar Perdana, I Ketut Gede Darma Putra, I Putu Arya Dharmaad	Classification of Sign Language Numbers Using the CNN Method	CNN	97%
[2]	Abul Abbas Barbhuiya & Ram Kumar Karsh & Rahul Jain	CNN based feature extraction and classification for sign language	CNN	99.82%
[3]	G.Ananth a Rao, K.Syamala, P.V.V.Kishore, A.S.C.S.Sastry	Deep Convolutional Neural Networks for Sign Language Recognition	CNN	92.88%
[4]	Basel Dabwan	CONVOLUTIONAL NEURAL NETWORK-BASED SIGN LANGUAGE TRANSLATION SYSTEM	CNN	96.68%
[5]	A R Syulistyo *, D S	SIBI (Sistem Isyarat Bahasa	CNN	100 %

	Hormansyah and P Y Saputra	Indonesia) translation using Convolutional Neural Network (CNN)		
[6]	Oscar Koller, Sepehr Zargaran, Hermann Ney, Richard Bowden	Deep Sign: Hybrid CNN-HMM for Continuous Sign Language Recognition	CNN - HMM	15% and 38% up to 13.3 %
[7]	Lionel Pigou, Sander Dieleman, Pieter-Jan Kindermans, Benjamin Schrauwen	Sign Language Recognition using Convolutional Neural Networks	CNN	91.7 %
[8]	Ahmed Adel Gomaa Elhagry, Rawan Gla Elrayes	Egyptian Sign Language Recognition Using CNN and LSTM	CNN & LSTM	90%
[9]	Kaustubh Jadhav, Abhishek Jaiswal, Abbas Munshi, Mayuresh Yerendekar	Sign Language Recognition Using Neural Network	CNN	
[10]	Adithya V.*a, Rajesh R.a	A Deep Convolutional Neural Network Approach for Static Hand Gesture Recognition	CNN	94.7 %
[11]	Sarfaraz Masood, Adhyan Srivastava, Harish Chandra Thuwal	Real-Time Sign Language Gesture (Word) Recognition	CNN and RNN	95.2 %
	and Musheer Ahmad	from Video Sequences Using CNN and RNN		
[12]	Vivek Bheda and N. Dianna Radpour	Using Deep Convolutional Networks for Gesture Recognition in American Sign Language	CNN	82.5 %

b. Theory

ASL

The data set used is a data set from kaggle.com where in the ASL dataset there is a total of 27,455 with 785 rows. However, the image shown is of course the same as in figure 1.

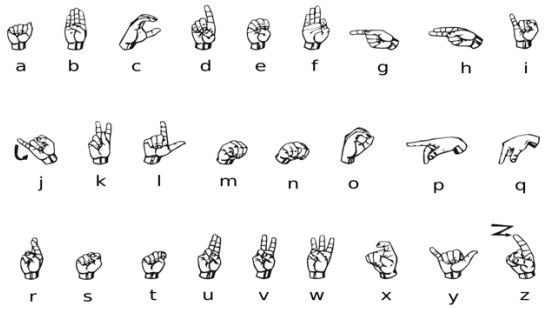


Figure 1 ASL

CNN

Machine learning is a subset of Artificial Intelligence, the difference is that Machine Learning will use even more data. One of the methods in machine learning is CNN (Convolutional Neural Network).

Convolutional Neural Network (CNN) is an architecture that can recognize information intended to predict an object. CNN's ability to recognize objects differs from the position of the input data. This ability makes Convolutional Neural Network (CNN) currently widely used in various fields [13].

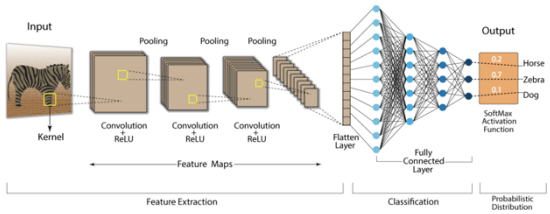


Figure 2 CNN

PYTHON

Python programming language will be used in processing data from the dataset and using the CNN method for detection. This programming language is very familiar to use among Machine Learning users.

3. METHODOLOGY

The data processing process starts from importing data, the data displayed will be like Figure 3. Then the next step is to process the signal image so that it can be read clearly.

label	pixel1	pixel2	pixel3	pixel4	pixel5	pixel6	pixel7	pixel8	pixel9	...	pixel775	pixel776	pixel777	pixel778	pixel779	pixel780	pixel781	pixel782	pixe
0	3	107	118	127	134	139	143	146	150	153	...	207	207	207	207	206	206	206	204
1	6	155	157	156	156	156	157	156	158	158	...	69	149	128	87	94	163	175	103
2	2	187	188	188	187	187	186	187	188	187	...	202	201	200	199	198	199	198	195
3	2	211	211	212	212	211	210	211	210	210	...	235	234	233	231	230	226	225	222
4	12	164	167	170	172	176	179	180	184	185	...	92	105	105	108	133	163	157	163

5 rows × 785 columns

Figure 3 head data of ASL

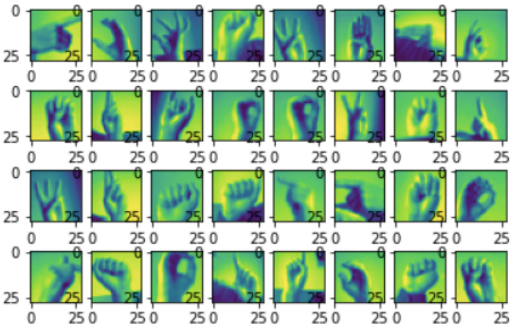


Figure 4 ASL for preprocessing

DETECTION

The following steps are to start the detection, the image displayed is an example of the signal letter b (figure 5) which will be changed to gray (figure 6) and then after detection, it will produce as shown in Figure 7.

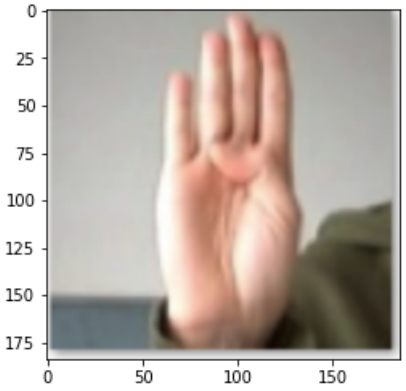


Figure 5 color detection

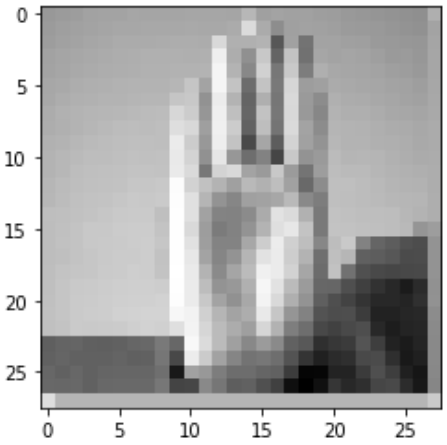


Figure 6 gray detection

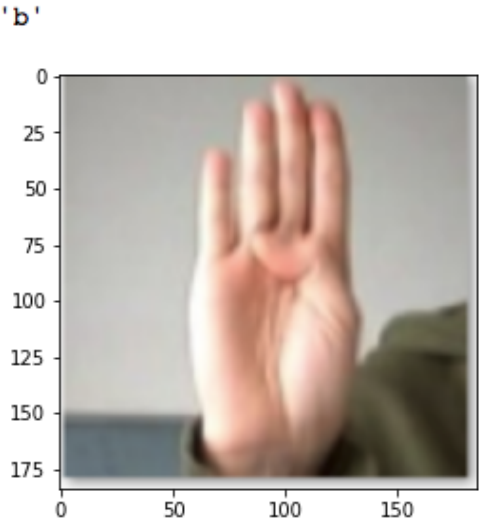
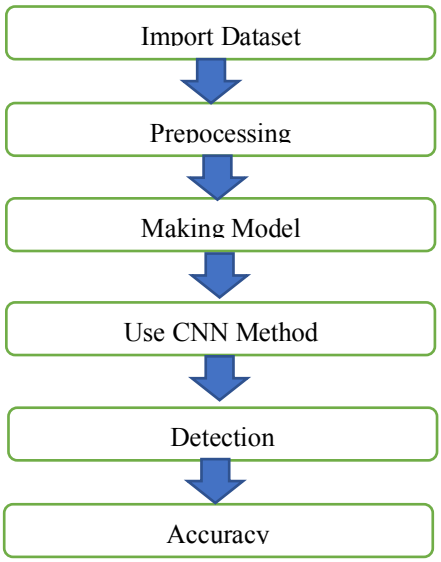


Figure 7 Result

in more detail the detection steps to be described as follows;



4. RESULT

From the steps that have been described previously, that there are steps for making a model as shown in the image below. This model is needed in the application of the CNN method to the dataset.

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 28, 28, 75)	750
batch_normalization (BatchNo	(None, 28, 28, 75)	300
max_pooling2d (MaxPooling2D)	(None, 14, 14, 75)	0
conv2d_1 (Conv2D)	(None, 14, 14, 50)	33800
dropout (Dropout)	(None, 14, 14, 50)	0
batch_normalization_1 (Batch	(None, 14, 14, 50)	200
max_pooling2d_1 (MaxPooling2	(None, 7, 7, 50)	0
conv2d_2 (Conv2D)	(None, 7, 7, 25)	11275
batch_normalization_2 (Batch	(None, 7, 7, 25)	100
max_pooling2d_2 (MaxPooling2	(None, 4, 4, 25)	0
flatten (Flatten)	(None, 400)	0
dense (Dense)	(None, 512)	205312
dropout_1 (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 24)	12312

Total params: 264,049
Trainable params: 263,749
Non-trainable params: 300

Figure 8 model

After the modeling step, when calculating the level of accuracy, the results are as shown below.

```
Epoch 1/20
858/858 [==============================] - 5s 6ms/step - loss: 0.2859 - accuracy: 0.9131 - val_loss: 0.1705 - val_acc
uracy: 0.9488
Epoch 2/20
858/858 [==============================] - 4s 5ms/step - loss: 0.0204 - accuracy: 0.9940 - val_loss: 0.3334 - val_acc
uracy: 0.9195
Epoch 3/20
858/858 [==============================] - 4s 4ms/step - loss: 0.0118 - accuracy: 0.9964 - val_loss: 0.7957 - val_acc
uracy: 0.8161
Epoch 4/20
858/858 [==============================] - 4s 5ms/step - loss: 0.0074 - accuracy: 0.9976 - val_loss: 0.1649 - val_acc
uracy: 0.9547
Epoch 5/20
858/858 [==============================] - 4s 5ms/step - loss: 0.0070 - accuracy: 0.9982 - val_loss: 0.3086 - val_acc
uracy: 0.9501
Epoch 6/20
858/858 [==============================] - 4s 4ms/step - loss: 0.0051 - accuracy: 0.9984 - val_loss: 0.3173 - val_acc
uracy: 0.9426
Epoch 7/20
858/858 [==============================] - 4s 4ms/step - loss: 0.0046 - accuracy: 0.9985 - val_loss: 0.1860 - val_acc
uracy: 0.9578
Epoch 8/20
858/858 [==============================] - 4s 4ms/step - loss: 0.0063 - accuracy: 0.9989 - val_loss: 0.1852 - val_acc
uracy: 0.9664
Epoch 9/20
858/858 [==============================] - 4s 4ms/step - loss: 0.0020 - accuracy: 0.9996 - val_loss: 0.2046 - val_acc
uracy: 0.9527
Epoch 10/20
858/858 [==============================] - 4s 4ms/step - loss: 0.0035 - accuracy: 0.9991 - val_loss: 0.1498 - val_acc
uracy: 0.9650
Epoch 11/20
858/858 [==============================] - 4s 4ms/step - loss: 0.0012 - accuracy: 0.9996 - val_loss: 0.2433 - val_acc
uracy: 0.9515
Epoch 12/20
858/858 [==============================] - 4s 4ms/step - loss: 0.0037 - accuracy: 0.9993 - val_loss: 0.1770 - val_acc
uracy: 0.9668
Epoch 13/20
858/858 [==============================] - 4s 4ms/step - loss: 0.0023 - accuracy: 0.9993 - val_loss: 0.1795 - val_acc
uracy: 0.9663
Epoch 14/20
858/858 [==============================] - 4s 4ms/step - loss: 0.0030 - accuracy: 0.9994 - val_loss: 0.4444 - val_acc
uracy: 0.9409
Epoch 15/20
858/858 [==============================] - 4s 4ms/step - loss: 0.0014 - accuracy: 0.9997 - val_loss: 0.8667 - val_acc
uracy: 0.8837
Epoch 16/20
858/858 [==============================] - 4s 4ms/step - loss: 0.0016 - accuracy: 0.9996 - val_loss: 0.3215 - val_acc
uracy: 0.9515
Epoch 17/20
 83/858 [=>............................] - ETA: 2s - loss: 0.0028 - accuracy: 0.9989
```

Figure 9 accuracy result

5. CONCLUSION

Based on the explanation above, that this study uses data taken from the American Sign Language (ASL) dataset, research using different datasets will of course produce different levels of accuracy. Likewise, the steps taken and the results obtained are still in the form of images, text or video captures. This research uses the ASL dataset and the steps are preprocessing and modeling then changing the color of the image and the detection results obtained are in the form of text and signal images, getting a very high level of accuracy, namely 99.89%.

REFERENCE

[1] I. P. I. Perdana, I. K. G. D. Putra and I. P. A. Dharmaadi, "Classification of Sign Language Numbers Using the CNN Method," *JITTER,* vol. 2, no. Vol. 2, No. 3 Desember 2021, 2021.

[2] A. A. Barbhuiya1, R. K. Karsh and R. Jain, "CNN based feature extraction and classification for sign language," *Springer,* 2020.

[3] G. Rao, K.Syamala, P.V.V.Kishore and A.S.C.S.Sastry, "Deep Convolutional Neural Networks for Sign Language Recognition," *SPACES,* 2018.

[4] B. Dabwan, "CONVOLUTIONAL NEURAL NETWORK-BASED SIGN LANGUAGE TRANSLATION SYSTEM," *International Journal of Engineering, Science and Mathematics,* vol. 9, no. 6, pp. 47-57, 2020.

[5] A. R. Syulistyo*, D. S. Hormansyah and P. Y. Saputra, "SIBI (Sistem Isyarat Bahasa Indonesia) translation using Convolutional Neural Network (CNN)," *The 1st Annual Technology, Applied Science and Engineering Conference,* 2020.

[6] O. Koller, S. Zargaran, H. Ney and R. Bowden, Deep Sign: Hybrid CNN-HMM for Continuous Sign Language Recognition, Germany: RWTH Aachen University, 2016.

[7] S. D. P.-J. K. B. S. Lionel Pigou, "Sign Language Recognition using Convolutional Neural Networks".

[8] A. A. G. Elhagry and R. G. Elrayes, "Egyptian Sign Language Recognition Using CNN and LSTM," *Comp. & Sys. Dept., Faculty of Engineering, Zagazig University, Egypt.*

[9] K. JADHAV, A. JAISWAL, A. MUNSHI and M. YERENDEKAR, "SIGN LANGUAGE RECOGNITION USING NEURAL NETWORK," *Department of Electronics and Telecommunication Engineering K.C. College of Engineering & Management studies & Research,* vol. 1, no. 1, 2020.

[10] A. V.*a and R. R.a, "A Deep Convolutional Neural Network Approach for Static Hand Gesture Recognition Gesture Recognition," *Procedia Computer Science,* 2020.

[11] S. Masood, A. Srivastava, H. C. Thuwal and M. Ahmad, "Real-Time Sign Language Gesture (Word) Recognition from Video Sequences Using CNN and RNN," *Springer,* 2018.

[12] V. Bheda and . . Radpour, "Using Deep Convolutional Networks for Gesture Recognition in American Sign Language," *Department of Computer Science, Department of Linguistics.*

[13] J. W. G. Putra, "Pengenalan Konsep Pembelajaran Mesin dan Deep Learning," *Research Gate,* 2020.