Hand Gesture Recognition using Convolutional Neural Network

Eduardo Brilliandy
Computer Science Department
School of Computer Science
Bina Nusantara University
Jakarta, Indonesia
eduardo.brilliandy@binus.ac.id

Javier Islamey
Computer Science Department
School of Computer Science
Bina Nusantara University
Jakarta, Indonesia
javier.islamey@binus.ac.id

***Abstract*—Hand gestures are one of the ways people communicate with each other. Hand gesture recognition allows hand gestures to interact with technology. A Convolutional Neural Network model is used to recognize and classify different hand gestures. The model is trained using a dataset that contains 10 different hand gestures. The model resulted in an accuracy of 98.2%. However, the model struggles to classify similar hand gestures when implemented.**

Keywords—Hand Gesture Recognition, Convolutional Neural Network

I. INTRODUCTION

Hand gestures are one of the ways people communicate with each other. It can convey information in a simple yet efficient way. Hand gestures can also be used to interact with technology. This is made possible through hand gesture recognition. Hand gesture recognition has been applied in various areas, such as Sign Language Recognition (SLR), 3D virtual environment, home automation, and even interactions with personal gadgets at home [1].

Computers recognize hand gestures by receiving an image input of the gesture itself and classifying it into one of the defined gesture categories. However, image inputs may have some problems such as different lighting, colour, contrast, background, etc. These problems add more dimensionality to images and may cause overfitting or excess consumption of resources for some image classification models. To solve this problem, a Convolutional Neural Network model was used because it can reduce the dimensionality of an image while still preserving the information it has.

II. LITERATURE STUDY

CNN has been primarily used for image-focused tasks for good reason. The architecture of CNN makes it suitable for tasks that use images by enabling the user to encode image-specific features [2]. It also requires less parameters, which means the weights are reusable [3]. This is known as Parameter Sharing and is one of the methods used to reduce the number of parameters [2]. This means that CNN can be powerful while also easier to set up.

Sohrab et. al. implemented CNN for recognizing Bengali hand signs from video feed. The machine learning methods used have built-in training classifiers with structures that work with Histogram of Oriented Gradients (HOG). Their dataset was preprocessed, such as denoising and grayscale transformation. Feature extraction was also done. The model was then trained using eight layers. While this is enough to detect most of the gestures, more layers and samples are required to properly detect identical gestures. Additionally, the model performed better when the video feed was at 30 FPS. The model achieved an accuracy of 98.75% [4].

Norah et. al. created a model for assisting people who have experienced stroke to communicate through hand gestures using CNN. Compared to previous research, the motions used in this research also included 3-D motion. The images used different background colors and illumination. The image frames used are converted to grayscale and resized to fit the model. Then, the model was trained through seven layers for 50 epochs. The model reached an accuracy of 99.12% [5].

Li et. al. also implemented CNN for gesture recognition. The preprocessing steps include depth image denoising using joint bilateral filtering and color segmentation. To reduce errors, a standard back propagation algorithm was used to modify the weights and thresholds of the hidden layer neurons. Then, SVM was used to improve the model. Since unlabeled samples are ignored by CNN, they will be labeled using the prediction capabilities of SVM. The model achieved 98.52% accuracy [6].

III. METHODOLOGY

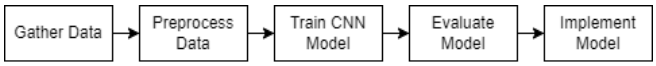

Fig. 1. Methodology Flowchart

The first step in the experiment is to gather the data that will be used to train the CNN model. The dataset used is retrieved from Kaggle consisting of 5243 images of 10 different hand gestures, such as 'call_me', 'fingers_crossed', 'okay', 'paper', 'peace', 'rock', 'rock_on', 'scissor', 'thumbs', 'up' [7]. Each hand gesture has around 500 images. Example images can be seen below.

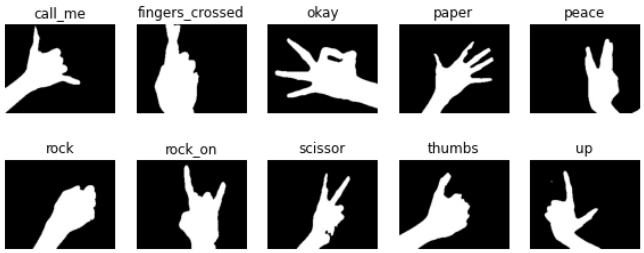

Fig. 2. Dataset Images

The next step is to preprocess the data. First, the model can only receive images of size 128 x 128, so the dataset is resized to make it usable for the model. Afterwards, the images are changed into grayscale for better processing. Then, they are blurred using Gaussian Blur with a kernel size of 5 to remove noise. Each input image is processed using Adaptive Binary Thresholding to separate the background and the hand. Adaptive Binary Thresholding can help produce better segmentation results and save time and resources. Afterwards, the images are put into the model. However, since the images in the dataset have already undergone binary thresholding, these preprocessing steps will only be applied during the implementation step.

After the preprocessing step, the data were used to train the CNN model. To do this, the dataset is split into 80% training data, 10% validation data, and 10% testing data. The CNN model contains 3 Convolutional layers with a 5x5 kernel size and contains dropout layers to reduce overfitting. The model is trained for 50 epochs. After the model has been trained, it was evaluated using the testing data to find the accuracy and loss values. Additionally, an accuracy and loss graph of the training and validation data were used to determine how well the model was trained.

Lastly, the trained model is implemented into a python program to find the effectiveness and robustness to predict real hand gestures from a video input.

IV. Results & Analysis

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 62, 62, 32)	832
max_pooling2d (MaxPooling2D)	(None, 31, 31, 32)	0
dropout (Dropout)	(None, 31, 31, 32)	0
conv2d_1 (Conv2D)	(None, 27, 27, 64)	51264
max_pooling2d_1 (MaxPooling2D)	(None, 13, 13, 64)	0
dropout_1 (Dropout)	(None, 13, 13, 64)	0
conv2d_2 (Conv2D)	(None, 9, 9, 128)	204928
max_pooling2d_2 (MaxPooling2D)	(None, 4, 4, 128)	0
dropout_2 (Dropout)	(None, 4, 4, 128)	0
flatten (Flatten)	(None, 2048)	0
dense (Dense)	(None, 128)	262272
dropout_3 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 10)	1290

Total params: 520,586
Trainable params: 520,586
Non-trainable params: 0

Fig. 3. CNN Model Result

Figure 3 shows the result of the trained CNN model. When evaluated using the testing data, the model achieved a high accuracy of 0.982 and a loss of 0.073. This means that the model performs well.

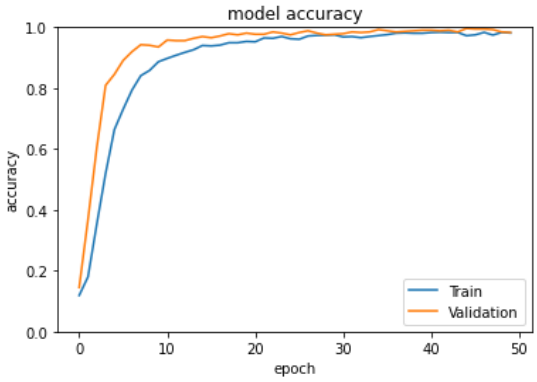

Fig. 4. Accuracy Graph

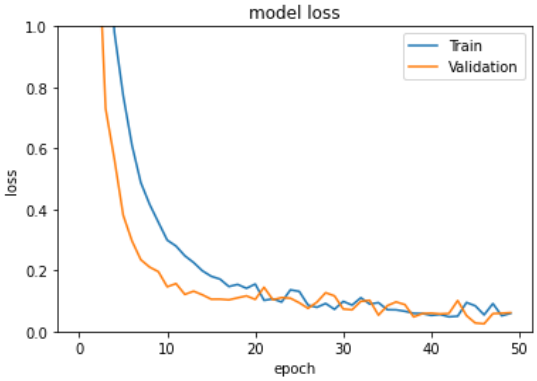

Fig. 5. Loss Graph

Figures 4 and 5 show the graph of the accuracy and loss of the model using the training and validation data for each epoch during training. From the graphs, we can see that the values for training and validation are similar, therefore we can assume that the model is not overfitting.

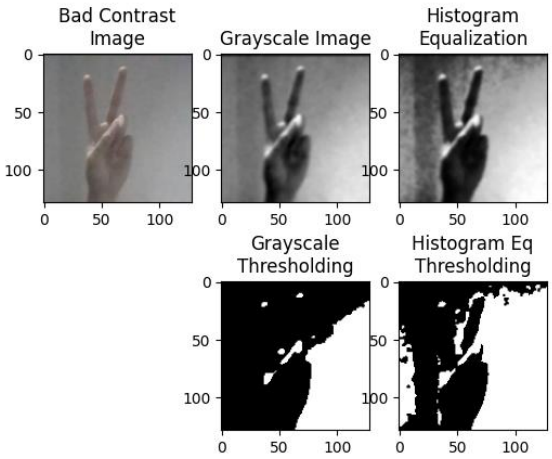

Fig. 6. Poor Binary Thresholding

However, when the model is implemented to detect real hand gestures from a video input, some problems occur. First, the adaptive thresholding method used to separate between the object and background does not perform well when given bad contrast images, even when histogram equalization is applied.

Fig. 7. Correct Predictions

Fig. 8. Incorrect Predictions

Another problem is even though the results from the evaluation are good, when implemented to predict real hand gestures from a video input, the model is only able to correctly predict 7 out of 10 hand gestures correctly. It is found that the incorrectly labeled gestures are ones that have other similar gestures. For example, the 'thumbs' and 'call_me' gestures are quite similar which leads to the model labeling both gestures as 'thumbs'.

V. Conclusion

In conclusion, using a Convolutional Neural Network model to classify different hand gestures performs decently well. However, the model struggles to differentiate between similar hand gestures. To get better results, a better method to separate foreground and background could be used. Additionally, a deep neural network model such as ResNet or AlexNet could be tested.

References

[1] M. Oudah, A. Al-Naji, and J. Chahl, "Hand gesture recognition based on computer vision: a review of techniques," *J Imaging*, vol. 6, no. 8, p. 73, 2020.

[2] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *arXiv preprint arXiv:1511.08458*, 2015.

[3] S. Saha, "A comprehensive guide to Convolutional Neural Networks-the eli5 way," Medium, 16-Nov-2022. [Online]. Available: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53.

[4] S. Hossain, D. Sarma, T. Mittra, M. N. Alam, I. Saha, and F. T. Johora, "Bengali hand sign gestures recognition using convolutional neural network," in *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*, 2020, pp. 636–641.

[5] N. Alnaim, M. Abbod, and A. Albar, "Hand gesture recognition using convolutional neural network for people who have experienced a stroke," in *2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, 2019, pp. 1–6.

[6] G. Li *et al.*, "Hand gesture recognition based on convolution neural network," *Cluster Comput*, vol. 22, no. 2, pp. 2719–2729, 2019.

[7] R. Sappani, "Hand gesture recognition," Kaggle, 13-Feb-2021. [Online]. Available: https://www.kaggle.com/datasets/roobansappani/hand-gesture-recognition.