Hand posture classification with convolutional neural networks on VGG-19 net Architecture

S Amir, Faturrahman and Hendra
Department of Computer Science, Universitas Hasanuddin, Makassar, Indonesia

Email: supriamir@unhas.ac.id

Abstract. This study aims to classify the image depth data Hand Posture. Hand Posture is a form of hand and movement used to communicate. Hand Posture is difficult to classify because various human hand objects are complex articulation objects. The model used in this study is Convolutional Neural Networks using the VGG-19 Net architecture. Based on the results shows an increase in the percentage of classification accuracy in each subject is 0.9976, 1.0, 0.9984, 1.0, and 0.9992 respectively.

1. Introduction
Today's technological era, developments are so rapidly occurring in all fields of science including technological developments, technological developments impact the dissemination of information so easy and can be accessed anywhere. Visual information usually forms images or imagery, the development of imagery data growth can help people recognize an object so that it can present Visual information. But image data can data that does not present information when the image is unknown to the object, so in an image, it is usually not only one object but there are several objects. Many news media now use hand posture to deliver the news to help deaf people. Hand posture is a sign language that is represented by a finger shape. Various patterns of fingers can represent the letters that exist so that it can help someone to convey information. Hand Posture is used to help someone who cannot hear or cannot speak (deaf) to communicate. Hand Posture contains a shape representing each letter and number, as many as 27 alphabets and 10 basic numbers have their hand shape. So that many forms of hands need to be carried out the object of each shape so that from the shape of the hand we can know the intent of the information provided by the Hand Posture [1]. However, the human hand is a complex articulation object consisting of many interconnected parts and joints [2]. How a computer can recognize an object is something that can help people at this time. The Hand Posture dataset has distinguishing characteristics against other datasets, such as images that cover a variety of hands using different conditions of ignition.

Machine learning techniques that extract relevant features are the most common used for hand posture classification, lacking full posture recognition [3]. Convolution Neural Networks is a trend method at this time to classify images, where research related to using CNN proved to have high accuracy, as Simonyan has done and Zisserman who uses the VGG-Net architecture won the Large Scale Visual Recognition Challenge 2014 (ILSVRC2014) competition and gained 92.7% accuracy [4], Adiguna and Yustinus in CNN Based research Posture Free Hand Detection using the CUDA-Convnet2structure architecture resulting in Accuracy 93.9% [5]. Thus, the Convolutional Neural Networks algorithm is a machine learning technique that successfully classifies various objects.
2. Material and methods

2.1. Deep Learning

Deep Learning is widely used for data analysis needs that combine it with artificial intelligence. Applications from the use of deep learning are many like image classification, driverless cars, news classification and widely used by large companies, such as Google, Microsoft, Facebook, IBM, Baidu, Apple, Adobe, Netflix, NVIDIA and NEC. Deep Learning techniques provide a great model for supervised learning algorithms. By adding training data, the learning model can represent a better class. In the creation of deep learning algorithms, artificial intelligence can strengthen the learning of the trained data. Such interactions can provide feedback between learning systems and user interaction experiences to improve performance in the tasks being studied [6].

2.2. Convolutional Neural Networks

CNN for feature learning: CNN was born in the Deep Learning (DL) era. Its goal is to model high level abstractions of visual data by using multiple non-linear transformation architectures. Among the DL models, CNN shows extraordinary performance, specifically in image classification and object recognition applications. Convolutional Neural Networks (CNN) is one of the machine learning methods of developing Multilayer Perceptron (MLP) designed to process two-dimensional data. CNN belongs to the type of Deep Neural Networks because it has a network level and multiple complementarities in image data [7]. CNN can use the convolution operation as the basis of its algorithm.

2.2.1. Convolution Layer

The convolution layer puts the inputs in the form of images through a series of convoluted filters, each of which activates certain features of the image. The convolution operation is usually associated with equation (1):

\[ C(i, j) = (I*K)(i, j) = \sum_m \sum_n I(m, n)K(i-m, j-n) \]  

Where, \(C(i,j)\): matrix of convolution results, \(I\) : image template, \(K\): kernel, \(m\) :line size of template, \(n\) : column size of template, \(i\) : row size of kernel, \(j\) : column size of kernel

2.2.2. Activation Layer

The activation function is a layer that allows training to be faster and more effective by mapping the values based on the activation function used. The frequently used activation function is the ReLU activation function or Rectified linear unit which is the activation function that maps negative values to zero and maintains a positive value. ReLU function is defined with the function:

\[ f(y) = \begin{cases} y, & \text{where } y > 0 \\ 0, & \text{where } y \leq 0 \end{cases} \]  

Input \(y\) is the input of the activation of nonlinear \(f\) on all inputs so that the equation (2) equals to \(f(y) = \max(0, y)\)

2.2.3. Pooling

Pooling is a layer to simplify the output that was previously in the conjunction. The principle uses a filter with a specific size and stride that will shift the entire feature map area.
Figure 1. Pooling

The purpose of the pooling layer is to reduce the dimensions of the feature map, thereby accelerating computing because the parameters that must be updated are fewer and can cope with overfitting, the pooling layer also results in an included process Downsampling.

Figure 2. Down sampling

2.3. Fully Connected Layer

The Fully Connected Layer receives the input of a vector from the previous layer, so that when the feature map output a multidimensional array is performed flatten so that the input becomes a vector. The flattening process can be seen in figure 3.

Figure 3. Flatten

The Fully Connected layer aims to transform the data dimensions to be classified linear. The fully connected layer connects all neurons from the previous layer with the neurons in the next layer. Each activity on the previous layer is connected so that there is a backpropagation algorithm that works to renew weights based on previously obtained errors. The key to CNN's success lies in neuron relationships that reduce the amount of weight and make the network easier to optimize.

2.4. VGG-19 Net

Visual Geometry Group (VGG) Networks architecture was introduced by Simonyan and Zisserman. The architecture of VGG Net has high accuracy performance capability. The architecture created by
Visual Geometry Group has 6 types of architecture. The architecture has a repeated convolution and pooling layer. VGG-19 Net amounted to 19 layers, namely 16 convolution layer and 3 Fully Connected Layer while VGG-16 Net is only 13 convolution layers and 3 Fully Connected Layer. A deep-structure VGG-Net showed that the depth of the network is an important factor for achieving good performance [8].

2.5. Performance Evaluation
Measuring a model's performance or performance is either a parameter or the good of a model can use the Confusion Matrix. Confusion Matrix is a table that has information on the classification results done. Confusion Matrix is a method to analyze how well the performance of a model has been made in identifying data from different classes.

| Table 1. Confusion Matrix |
|---------------------------|
| Class Actual | 1 Predicted | Class 2 Predicted |
| Class Actual 1 | True Positive | False Negative |
| Class Actual 2 | False Positive | True Negative |

Class 1 declares a positive class while Class 2 declares a negative class. The results of the confusion matrix table can calculate accuracy, precision, and recall by models that have been created with the following equations.

\[
Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}
\]

\[
Precision = \frac{TP}{TP + FP} \tag{4}
\]

\[
Recall = \frac{TP}{TP + FN} \tag{5}
\]

Where, TP: True Positive, TN: True Negative, FP: False Positive, FN: False Negative.

3. Results
The dataset used is the Hand Posture dataset. The hand posture dataset used is the Byeongkeun Kang dataset derived from the Byeongkeun Kang GitHub titled Fingerspelling Recognition Dataset. This Dataset consists of 5 different subjects with 31 classes, each class having 200 images per subject, so the number of images in this data is 31,000 images. The image contained in it is a combination of the 5 hands of the subject taken with different camera positions and has been processed by the image-maker so that it appears the depth of the data in the image as figure 4.
3.1. Preprocessing

Preprocessing of data on this research is the form of labeling, resize, and data sharing.

3.1.1. Labeling. At this stage, every data in the labeling is based on its class.

**Table 2. Label Data**

| Class | Label | Class | Label |
|-------|-------|-------|-------|
| 1     | A     | 16    | Q     |
| 2     | B     | 17    | R     |
| 3     | C     | 18    | S     |
| 4     | D     | 19    | T     |
| 5     | E     | 20    | U     |
| 6     | F     | 21    | V     |
| 7     | G     | 22    | X     |
| 8     | H     | 23    | Y     |
| 9     | I     | 24    | 1     |
| 10    | K     | 25    | 2     |
| 11    | L     | 26    | 3     |
| 12    | M     | 27    | 4     |
| 13    | N     | 28    | 5     |
| 14    | O     | 29    | 7     |
| 15    | P     | 30    | 8     |
| 31    | 9     |

3.1.2. Resize. At this stage, all data will be resized to 224 x 224 x 3 which was previously the size is 256 x 256 x 3.

3.1.3. Data Sharing. Datasets consisting of each subject will be in their data for training and data testing. The data sharing of each subject in the training data is 80% and 20% for testing. So that each subject has data like the table 3 below:
### Table 3. Data Training and Data Testing

| Data   | Subject 1 | Subject 2 | Subject 3 | Subject 4 | Subject 5 |
|--------|-----------|-----------|-----------|-----------|-----------|
| Training | 4960      | 4960      | 4960      | 4960      | 4960      |
| Testing  | 1240      | 1240      | 1240      | 1240      | 1240      |

### 3.2. Training

The development of the model for training based on the training data of each subject with the Convolutional Neural Networks with the architecture VGG-19 Net. CNN's architecture shows in table 4.

### Table 4. Model Summary

| No. | Layer             | Output          | Parameter |
|-----|-------------------|-----------------|-----------|
| 1   | Convolutional     | 224,224,64      | 1792      |
| 2   | Convolutional     | 224,224,64      | 36928     |
| 3   | Max Pooling       | 112,112,64      | 0         |
| 4   | Convolutional     | 112,112,128     | 73856     |
| 5   | Convolutional     | 112,112,128     | 147584    |
| 6   | Max Pooling       | 56,56,128       | 219       |
| 7   | Convolutional     | 56,56,256       | 295168    |
| 8   | Convolutional     | 56,56,256       | 590080    |
| 9   | Convolutional     | 56,56,256       | 590080    |
| 10  | Convolutional     | 56,56,256       | 590080    |
| 11  | Max Pooling       | 28,28,256       | 0         |
| 12  | Convolutional     | 28,28,512       | 1180160   |
| 13  | Convolutional     | 28,28,512       | 1179904   |
| 14  | Convolutional     | 28,28,512       | 1180160   |
| 15  | Convolutional     | 28,28,512       | 2359808   |
| 16  | Max Pooling       | 14,14,512       | 0         |
| 17  | Convolutional     | 14,14,512       | 1179904   |
| 18  | Convolutional     | 14,14,512       | 1180160   |
| 19  | Convolutional     | 14,14,512       | 2359808   |
| 20  | Convolutional     | 14,14,512       | 2359808   |
| 21  | Max Pooling       | 7,7,512         | 0         |
| 22  | Flatten           | 25088           | 0         |
| 23  | Dense             | 4096            | 102764544 |
| 24  | Dense             | 4096            | 16781312  |
| 25  | Dense             | 31              | 127007    |

The architecture used has 16 convolution layers and 3 fully connected layer. The convolution layer with filters differs from 64, 128, 256 and 512, each convolution layer has a 3 x 3 kernel and uses a batch of normalization to improve the performance and stability of the model and end with pooling the max pooling. After passing the 16 convolution layer is the fully connected layer which amounted to 3 and ends with Softmax activation function for classifying by weight. Total parameters or weights formed from models as much as 135,030,879 neurons. Each subject will be training as 100 epochs.
Table 5. Result Accuracy and Loss

| Subject  | Subject 2 | Subject 3 | Subject 4 | Subject 5 |
|----------|-----------|-----------|-----------|-----------|
| Accuracy | 0.9976    | 1.0       | 0.9984    | 1.0       | 0.9992    |
| Loss     | 0.0071    | 2.2017e-04| 0.0149    | 7.2526e-05| 0.0021    |

Table 6. Result Recall and Precision Subject 1

| Subject 1 | Value |
|-----------|-------|
| True      | 1237  |
| False     | 3     |
| Recall    | 0.9967|
| Precision | 0.9974|

Table 7. Result Recall and Precision Subject 2

| Subject 2 | Value |
|-----------|-------|
| True      | 1240  |
| False     | 0     |
| Recall    | 1     |
| Precision | 1     |

Table 8. Result Recall and Precision Subject 3

| Subject 3 | Value |
|-----------|-------|
| True      | 1238  |
| False     | 2     |
| Recall    | 0.9987|
| Precision | 0.9987|

Table 9. Result Recall and Precision Subject 4

| Subject 4 | Value |
|-----------|-------|
| True      | 1240  |
| False     | 0     |
| Recall    | 1     |
| Precision | 1     |

Table 10. Result Recall and Precision Subject 5

| Subject 5 | Value |
|-----------|-------|
| True      | 1239  |
| Falsees   | 1     |
| Recall    | 0.9990|
| Precision | 0.9993|
4. Conclusion
In this paper, we proposed a hand posture classification method based on VGG-19 Net using depth data. The VGG-19 Net architecture in this study used the input shape (224 x 224 x 3), with a 3 x 3 kernel, and the number of epochs retained per subject was 100. The data used for training models was 4960 resulted in accuracy for subject 1, subject 2, subject 3, subject, 4, and subject 5 respectively 0.9976, 1.0, 0.9984, 1.0, and 0.9992 in data testing as 1240. The deeper the layer and the more epoch done on the architecture of VGG-19 Net then it will improve the accuracy and reduce the value of a loss on the dataset used.

References
[1] Takimoto H, Yoshimori S, Mitsukura Y and Fukumi M 2010 Hand posture recognition robust for posture changing in complex background. Journal of Signal Processing, 14 483-490
[2] Dominio F, Donadeo M, Marin G, Zanuttigh P and Cortelazzo G.M 2013 Hand gesture recognition with depth data. ARTEMIS’13 Proceedings of the 4th ACM IEEE international workshop on Analysis and retrieval of tracked events and motion in Imagery Stream Barcelona 9-16
[3] Simonyan K and Zisserman A 2015 Very Deep Convolutional Networks for Large-Scale Image Recognition ICLR Oxford University of Oxford
[4] Adiguna R and Soelistio Y E 2018 CNN Based Posture-Free Hand Detection. 10th The International Conference on Information Technology and Electrical Engineering. Indonesia
[5] Bishop M C 2016 Pattern Recognition and Machine Learning Springer
[6] Alpaydin E 2009 Introduction to Machine Learning; Second Edition MIT Press London,
[7] Minaee S, Abdolrashidi A and Wang Y 2016 An experimental study of deep convolutional features for iris recognition. IEEE Signal Processing in Medicine and Biology Symposium (SPMB) Philadelphia, PA, 1-6.