Scientific Journal of Informatics
Vol. 10, No. 3, Aug 2023

p-ISSN 2407-7658 http://journal.unnes.ac.id/nju/index.php/sji e-ISSN 2460-0040

Alphabet Classification of Sign System Using Convolutional Neural Network with Contrast Limited Adaptive Histogram Equalization and Canny Edge Detection

Ahmad Solikhin Gayuh Raharjo[1*], Endang Sugiharti[2]
[1,2]Computer Science Department, Faculty of Mathematics and Natural Sciences, Universitas Negeri Semarang, Indonesia

Abstract.
Purpose: There are deaf people who have problems in communicating orally because they do not have the ability to speak and hear. The sign system is used as a solution to this problem, but not everyone understands the use and meaning of the sign system, even in terms of the alphabet. Therefore, it is necessary to classify a sign system in the form of American Sign Language (ASL) using Artificial Intelligence technology to get good results.
Methods: This research focuses on improving the accuracy of ASL alphabet classification using the VGG-19 and ResNet50 architecture of the Convolutional Neural Network (CNN) method combined with Contrast Limited Adaptive Histogram Equalization (CLAHE) to improve the detail quality of images and Canny Edge Detection to produce images that focus on the objects in it. The focused result is the accuracy value. This study uses the ASL alphabet dataset from Kaggle.
Results: Based on the test results, there are three best accuracy results. The first is using the ResNet50 architecture, CLAHE, and an image size of 128 x 128 pixels with an accuracy of 99.9%, followed by the ResNet50 architecture, CLAHE + Canny Edge Detection, and an image size of 128 x 128 pixels with an accuracy of 99.82 %, and in third place are the VGG-19 architecture, CLAHE, and an image size of 128 x 128 pixels with an accuracy of 98.93%.
Novelty: The novelty of this study is the increase in the accuracy value of ASL image classification from previous studies.

Keywords: Classification, American sign language, Convolutional neural network, CLAHE, Canny edge detection
Received May 2023 / **Revised** May 2023 / **Accepted** June 2023

INTRODUCTION

Communication is one of the most important things in human life. In communicating there is verbal which is done by speaking, and non-verbal which is done by using hand gestures, head movements, and symbols [1]. Communication is the basic and the most important thing, the achievement of public relations and other important things is the result of understanding communication [2]. The most frequently used communication is the verbal way, but there are deaf people who do not have the ability to communicate verbally [3]. The use of a sign system is a solution for this communication problem for deaf people because it uses hand gestures that symbolize certain meanings, the way it is used makes the sign system a form of non-verbal communication. Sign system is also known as sign language, where American Sign language (ASL) is a very popular sign language because many countries implemented it into their sign language. Indeed, not everyone knows the meaning of sign language, even in terms of the alphabet.

Classification is a way to group objects based on the shape, color, and its properties. The classification method is a tool for creating a model that can predict a class based on its attributes, more specifically, it is differentiated based on information from the data [4]. The purpose of classification is to make predictions using numeric data that represents the class [5]. Classification can be used to determine the alphabet of sign language, in carrying out this classification it can be automated with Artificial Intelligence (AI) technology. AI is one of the technological developments products which machines can be implemented with intelligence

*Corresponding author.
Email addresses: ahmadsgr39@students.unnes.ac.id (Raharjo), endangsugiharti@mail.unnes.ac.id (Sugiharti)
DOI: 10.15294/sji.v10i3.44137

like the human brain. An algorithm is needed to embed intelligence in a machine, the algorithm is named Machine Learning (ML). ML is the scope of AI that can transform data into models full of information from the data and later will implement into machines [6]. Deep Learning (DL) is still the scope of ML, but has better capabilities in dealing with big data. DL utilizes an Artificial Neural Network that is inspired by the way human neurons work [7].

Convolutional Neural Network (CNN) is one of the DL algorithms that is focused and very popular in image classification. To apply CNN, there are many architectures that can be used, such as Visual Geometry Group (VGG) and Residual Network (ResNet) [8]. Research by Triyadi et al. A study to classify cataracts using CNN with the VGG-19 and ResNet50 architectures, the result is 91.06% for VGG-19 and 93.5% for ResNet50 [9]. According to [10], their study used CT-Scan image to classify SARS-CoV-2 using ResNet50 architecture and produces an accuracy of 95%. Based on previous research, CNN has proven its ability to classify images so that it can also classify ASL alphabets.

Preprocessing is processing the original data before it is applied to the next step, the aim is to improve the quality or make the data according to the next stage so the output results are in accordance with what is desired. Preprocessing of image data is divided into two, image enhancement to remove noise, increasing brightness and adjusting contrast [11]. Meanwhile, image segmentation is used to change the image to shape what we desire, such as gray level and edge detection. The application of preprocessing image data in classifications work can improve the results of the model accuracy. The stages of image preprocessing are image enhancement followed by image segmentation to obtain information from the image [12].

Previous research by [13] using color moment + Hu moment + GLCM + SVM to classify the ASL alphabet has a low accuracy of 87%, SVM has a weakness to process large data such as the alphabet which has 26 classes. All preprocessing combined in SVM will not have a significant effect, because the core of the problem lies in the SVM algorithm itself. Research [14] classifying the ASL alphabet using MobileNetV2 with an accuracy of 98.67%, the weakness in this study is not applying preprocessing to the dataset. This causes a lack of information obtained from the dataset and a lack of clarity about what information the model must learn. Research by [15] uses TSM + ResNet50 for ASL alphabet classification with 97.57% accuracy, this research also doesn't get better information from the dataset because only add the calculation process. Adding TSM to the algorithm without preprocessing the data will not have a big effect if the main problem is in the dataset.

Based on the explanation and weaknesses in previous studies above, this research is focuses on improving the accuracy of ASL alphabet classification by implementing VGG-19 and ResNet50 architecture of CNN method combined with Contrast Limited Adaptive Histogram Equalization (CLAHE) as image enhancement and Canny Edge Detection as image segmentation using a dataset from Kaggle named “ASL Alphabet”.

METHODS

The research stages are shown in Figure 1. Each step in Figure 1 is explained in detail at the next section.

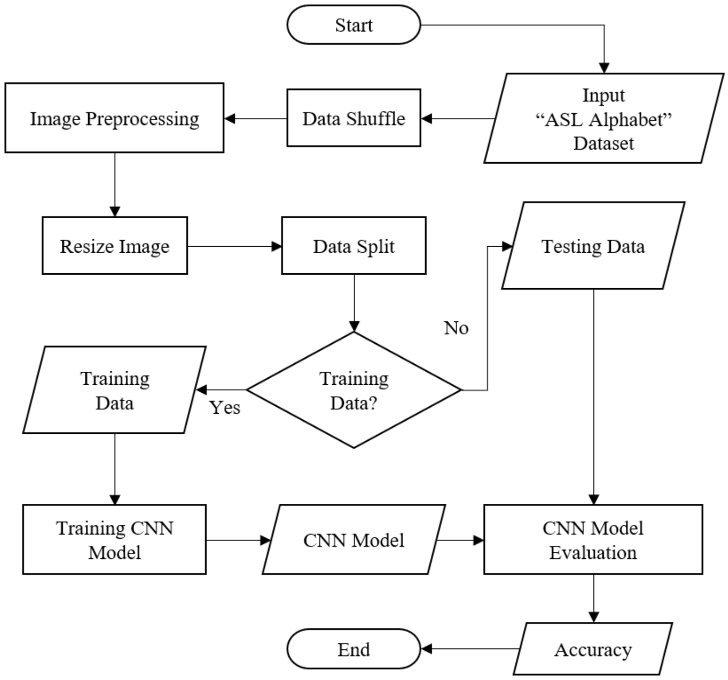

Figure 1. Flowchart of the proposed method

Data Collection

The dataset used in this study is the "ASL Alphabet" dataset from Kaggle repository. This dataset consists of 26 classes based on the number of alphabets where each class has 3,000 images. A sample of the dataset used can be seen in Figure 2.

Figure 2. Dataset sample

Data Shuffle

The dataset that has been obtained is then shuffled which aims to randomize the previously ordered dataset by the alphabet, the purpose of data shuffle is also to prevent the model from learning based on dataset patterns. The goal is that in the next process the data is divided properly.

ImageProcessing

In this stage, preprocessing of the original image data is carried out using CLAHE to increase contrast and brightness and Canny Edge Detection to detect the edges of objects. This aims to improve image quality and prepare the image to the desired shape.

CLAHE preprocessing can produce images with better improvement in visibility level than the original image, CLAHE can use both images colored and graylevel [16]. The CLAHE method overcomes limitations with a global approach by dividing the image into sub-images and applying contrast enhancement locally, this method used two essential parameters, tiles to determine the size of sub-images and clip limit as a histogram limit [17]. CLAHE produces better and more effective image quality than the Adaptive Histogram Equalization (AHE) on Magnetic Resonance Imaging (MRI) images [18].

CLAHE uses clip as the upper limit and divides the image based on the size of the tiles that have been set before. The steps in the CLAHE implementation process are dividing the original image into sub-images based on predetermined tiles, calculating the pixel levels for each sub-image, finding the average pixel levels for each sub-image from the number of pixels of the X and Y dimensions as shown in Equation 1.

$$\boldsymbol{N_{avg}} = (\boldsymbol{NrX} \times \boldsymbol{NrY}) / \boldsymbol{N_{levels}} \tag{1}$$

Normalizing the clip limit using input clip limit, region of the size, and highest pixels in sub-image as shown in Equation 2.

$$\boldsymbol{N_{clip}} = \frac{\boldsymbol{M}}{\boldsymbol{256}}\left(\boldsymbol{1} + \frac{\boldsymbol{\alpha}}{\boldsymbol{100}}(\boldsymbol{levels_{max}} - \boldsymbol{1})\right) \tag{2}$$

Last step is calculation to find the actual clip limit in Equation 3.

$$\boldsymbol{N_{CL}} = \boldsymbol{N_{clip}} \times \boldsymbol{N_{avg}} \tag{3}$$

Canny Edge Detection is a method that can detect the edges of objects, but this method has a weakness in detecting if the image has a lot of noise and does not have good object details [19]. The weakness in Canny Edge Detection can be solved with previous CLAHE which will improve the object detail from the image. Canny Edge Detection is the best method for detecting edges compared to Sobel, Prewit, and Robert methods because it uses upper and lower thresholds with flexibility in value [20]. The steps of implementing Canny Edge Detection include the filtering process using Gaussian Kernel, looking for intensity gradient values from the image for the x and y axes, removing unnecessary pixel edges, determining the upper and lower threshold values, and identifying endpoint whether the pixel is an edge or not [21]. The steps in implementing Canny Edge Detection are applying a 5 x 5 Gaussian blur kernel to the image, calculating the gradient of the x and y axes shown in Equation 4.

$$\begin{gathered} I_x = I * D_x \\ I_y = I * D_y \end{gathered} \tag{4}$$

Calculating the edge gradient value in Equation 5.

$$|G| = \sqrt{I_x^2 + I_y^2} \tag{5}$$

Calculating operations looking for gradient orientation in Equation 6.

$$\theta = \arctan\left(\frac{I_x}{I_y}\right) \tag{6}$$

Calculating the upper and bottom threshold are shown in Equation 7.

$$\begin{gathered} th_{weak} = \max(G) * 0{,}1 \\ th_{strong} = \max(G) * 0{,}5 \end{gathered} \tag{7}$$

The last steps determined the pixel whether it is and edge or not based on predetermined upper and lower thresholds.

Resize Image

Image resizing is a process in which the image is resized from original dimensions which is 200 x 200 pixels to 32 x 32 pixels and 128 x 128 pixels. The purpose of resizing the image is to make the image have the same pixel size, this is because the training process must input the image with a predetermined size [22]. In addition to making the same image size, this stage is also intended to lighten the dataset so it does not require very large computations needed. An illustration of the resized image process can be seen in Figure 3.

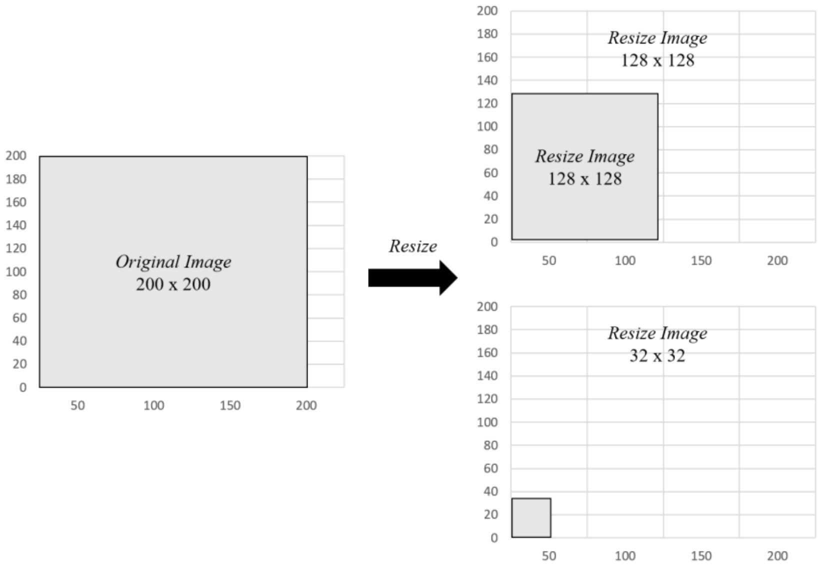

Figure 1. Resize image

Data Split

Data split is the process of dividing data into two separated data, there is training data and testing data. In general, the ratio used in splitting data is 80:20, this is based on the well-known Pareto principle, but there is no certainty what the best ratio to use [23]. Based on Pareto principle the division is done with ratio 20:80, in detail the distribution of the dataset is shown in Table 1.

Table 1. Data Split

Training Data Percentage	Testing Data Percentage	Amount of Training Data	Amount of Testing Data	Total Data
80%	20%	62400	15600	78000

CNN Model Training

The training model is a stage where the model will be formed by training data that has been prepared beforehand to carry out the classification. The training process will use the VGG-19 and ResNet50 architectures from CNN method. The CNN algorithm is part of a deep learning model for processing image data, CNN is designed to be able automatically recognize patterns in images from simple patterns to complex patterns [24]. CNN is divided into 3 main layers, there are convolutional layer, pooling layer, and fully connected layer [25]. The convolution layer is used to extract input data and its weight is determined randomly, the pooling layer is also known as down-sampling because it takes samples from the data set to make it the most important data, an the fully connected layer is a collection of neurons that have been grouped and connected to each other [26]. The layers on CNN are illustrated in Figure 4.

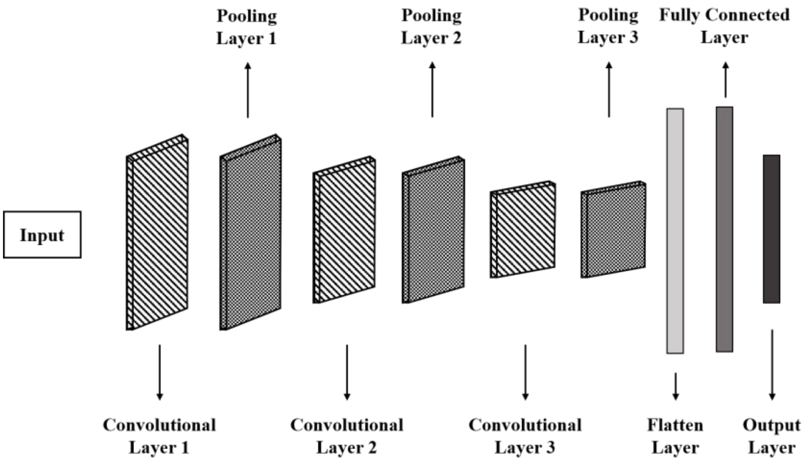

Figure 2. CNN layer

The example of the convolution process on an image with size 3 x 3 that applied 2 x 2 kernel can be seen in Figure 5.

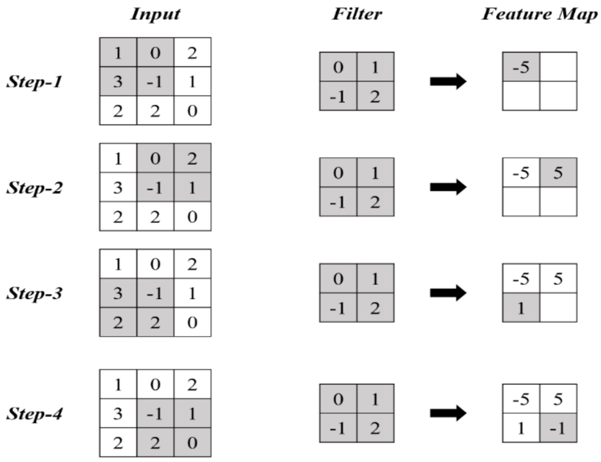

Figure 3. Convolution process

The calculations for the convolution process can be seen in Equation 8, where I is the input image, K is the kernel that is determined randomly, m and n describe the rows and columns in the image [25].

$$S(i,j) = (I * K)\ (i,j) = \sum_m \sum_n I(m,n)\ K(i-m, j-n) \tag{8}$$

There are many types of pooling layer, such as max pooling, min pooling, and average pooling. The pooling layer method is shown in Figure 6. In this research using average pooling.

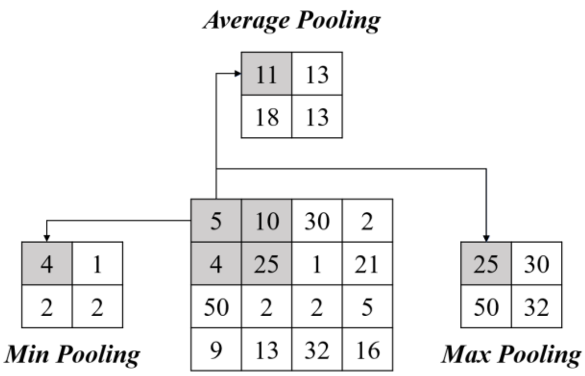

Figure 6. Pooling layer types

The use of CNN algorithm cannot be separated from the activation function in several layers, Rectified Linear (ReLu) and softmax are examples of activation functions that can be used on CNN with different functions, calculations, and uses. ReLu has low computation compared to sigmoid and tanh, but the absence of an upper value limit is a weakness in ReLu [27]. Calculations from the ReLu activation function can be seen in Equation 9.

$$f(x) = \max(0, x) = \begin{cases} x, x \geq 0 \\ 0, x < 0 \end{cases} \tag{9}$$

Softmax is the activation function used in multi-class classification with the output result is value between 0 and 1 obtained from the calculation in Equation 10. The result value is the probability of each class and softmax activation is highly recommended for the output layer in DL [28].

$$f(x_i) = \frac{e^{(x_i)}}{\sum_{j=1}^{k} e^{xj}} \tag{10}$$

Apart from the activation function, DL also uses an optimizer to update the weights to minimize the loss function value and improve the prediction results from the model. Adaptive Moment Estimation (ADAM) is an example of an optimizer that can be used and has lower computation than other optimizers, Equation 11 shows the calculation of the updated weights from ADAM. The way ADAM works is to calculate the learning rate for each parameter and use a quadratic gradient at a learning rate such as RMSProp and a moving average of a gradient such as Momentum [8].

$$w_t = w_t + \Delta w_t \tag{11}$$

VGG is the architecture of CNN, VGG-19 is one of the bases of VGG architecture which has 16 convolutional layers and 3 fully connected layers [29]. VGG-19 has a characteristic where the number of filters in each convolutional layer will increase in number but have a smaller size. Figure 7 shows the architecture of VGG-19.

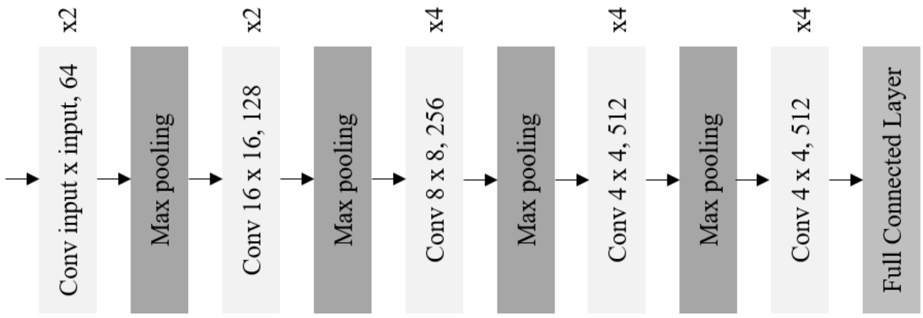

Figure 7. VGG-19 Architecture

ResNet is an architecture from CNN that is designed to have a very deep network to overcome the vanishing gradient issue, ResNet50 is the type of architecture that is most commonly used [8]. In ResNet50 there are two paths in a block, where the left path is a bottleneck structure for learning new features and the right path is called the skip or shortcut path which will later be combined and then an activation function is applied. The structure of ResNet50 can be seen in Figure 8. The blocks in ResNet are called residuals, these blocks will bring the information from the data before the convolution process is carried out and combine it so that the model can study information from the input data more deeply compared to other architectures [30].

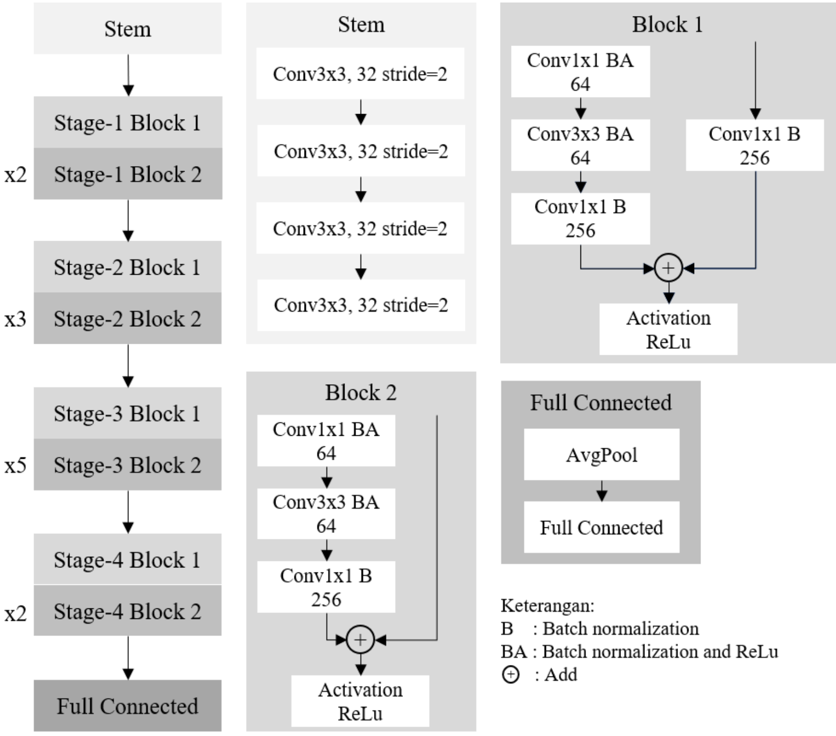

Figure 8. ResNet50 architecture

CNN Model Evalution

At this stage an evaluation will be carried out on the model that was formed in the previous stage using data testing to obtain an accuracy value as the main indicator. In this process all models that have been formed will be compared and evaluated whether the model is included in the feasible category or vice versa. The calculation to get the accuracy value can be seen in Equation 12, where T is the correct prediction and n is the total data.

$$Accuracy = \frac{\sum T}{n} \, x \; 100\% \qquad (12)$$

RESULTS AND DISCUSSIONS

The initial results in this study are images that have been preprocessed using CLAHE and Canny Edge Detection, these new images become new datasets which will later be used in the training process. For the first dataset that was applied CLAHE preprocessing to produce images with more visible details, a sample of the CLAHE dataset result can be seen in Figure 9.

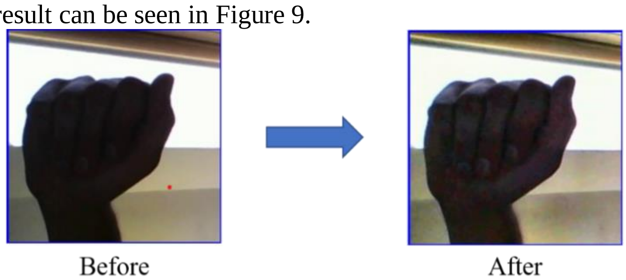

Figure 9. CLAHE image sample

Figure 9 shows better image quality after CLAHE, the details of the fingers become more visible and the hand object in the image are clearer because CLAHE adjusts contrast and brightness in the image. The dataset that has been applied to CLAHE and Canny Edge Detection preprocessing produces black and white images with white borders, sample from the CLAHE + Canny Edge Detection dataset are shown in Figure 10.

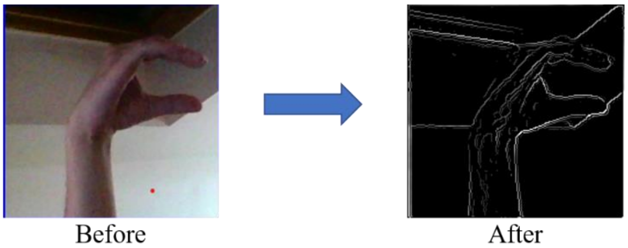

Figure 10. CLAHE + Canny edge detection

Figure 10 shows the images after Canny Edge Detection which produce a black and white image with the border between the hand object and its background colored as white. The border will become information of the object during the training process. Furthermore, the results of the trials from this research will be carried out by combining the images pixel size, the preprocessing used, and the CNN architecture. So, 8 trials were obtained from this combination, details of the trial scenario are shown in Table 2.

Table 2. Trials scenario

Trial	Size (pixels)	Preprocessing	Architecture
1	32 x 32	CLAHE	VGG-19
2	128 x 128	CLAHE	VGG-19
3	32 x 32	CLAHE	ResNet50
4	128 x 128	CLAHE	ResNet50
5	32 x 32	CLAHE + Canny Edge Detection	VGG-19
6	128 x 128	CLAHE + Canny Edge Detection	VGG-19
7	32 x 32	CLAHE + Canny Edge Detection	ResNet50
8	128 x 128	CLAHE + Canny Edge Detection	ResNet50

All trials were carried out using the same tuning parameters in the process parameters using 20 epochs, batch size 32, ADAM optimizer, learning rate 0.001, average pooling layer, softmax output layer, and categorical cross entropy for loss function. Comparison of the results of each model is shown in Figure 11.

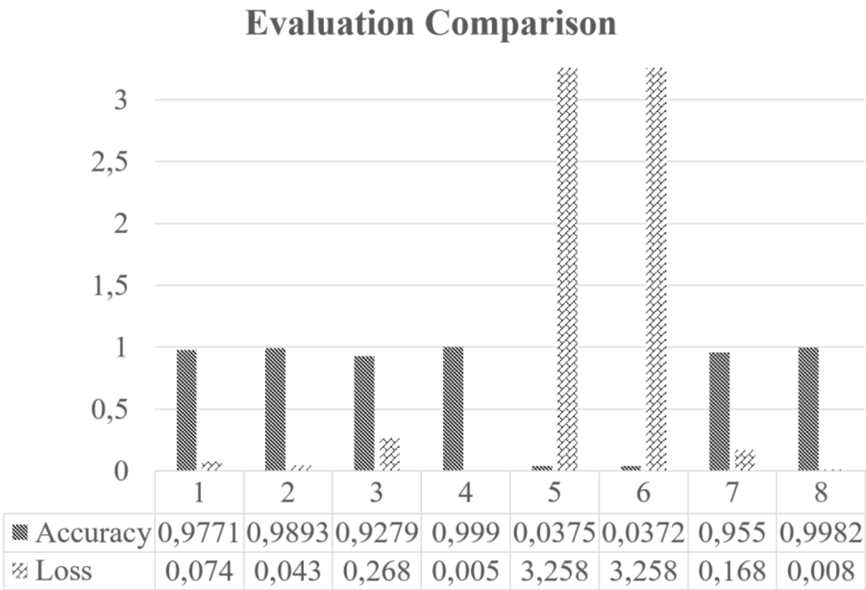

Figure 4. Results comparison of each trial

Figure 11 shows the results of 8 trials conducted on this study. From the figure it is known that trial 4 using CLAHE, ResNet50, and 128 pixels images became the best with 99.9% accuracy and loss value of 0.005, trial 8 using CLAHE + Canny Edge Detection, ResNet50, and 128 pixels images also received good accuracy results of 99.82% and loss value of 0.008. This is because trials 4 and 8 use the ResNet50 architecture which conducts deeper training with the residual concept as can be seen in Figure 8, trial 4 is better because the images have more information from the objects such as color and shape compared to the image on trial 8 which only retrieves information from border of the object.

Trial 2 using CLAHE, VGG-19, and 128 pixels images also had good results with accuracy of 98.93% and loss value of 0.043. The results from VGG-19 are not as good as those from ResNet50 because the architectural structure of VGG-19 is not very deep as can be seen in Figure 7, while the ASL alphabet dataset has quite a lot of 26 classes. The large number of classes in the dataset makes the model formed by the deeper architecture or have more layers will get better results.

Trials 5 and 6 were failed because they produced accuracy below 5% and very large loss values, both of these trials used CLAHE + Canny Edge Detection and VGG-19 with only different image sizes. This failure is caused by three things, first is because the dataset itself uses black and white images so that information from pixels only has a value of 0 (black) and 1 (white), second is the ReLu activation function shown in Equation 9 will tend to get a result of 0 after convolution process are carried out, the last is the use of linear VGG-19 which will cause information from the image to be lost unlike ResNet50 in trials 7 and 8 with its residual concept which carries information from the previous image to prevent loss of information after the convolution process.

The image size also affects the results of accuracy, it can be seen that trial 2 is better than trial 1 because it uses a larger image size, this also happened between trials 3 and 4 and trials 7 and 8. Larger image sizes have more pixel information in the image so that they will produce better models during the training process.

From a total of 8 trials conducted in this study, there were 3 trials with very good accuracy results and able to improve compared to previous studies. Comparison of the accuracy obtained from this study with previous research is shown in Table 3.

Table 3. Comparison of accuracy result

Algorithm	Reference	Accuracy
MobileNetV2	[14]	98.67%
color moment + Hu moment + GLCM + SVM	[13]	87%
TSM + ResNet50	[15]	97.57%
Proposed Method (CLAHE + VGG-19 + 128 pixel)		98.93%
Proposed Method (CLAHE + Canny Edge Detection + ResNet50 + 128 pixel)		99.82%
Proposed Method (CLAHE + ResNet50 + 128 pixel)		99.9%

Table 3 shows that the 3 trials conducted in this study obtained better accuracy than the previous study by Lum et al. [14] which uses the MobileNetV2 architecture, the most improvement is an accuracy obtained by the proposed method using CLAHE + Canny Edge Detection + ResNet50 + 128 pixel with 1.23% increased accuracy.

CONCLUSION

Based on the research results obtained, it can be concluded that the preprocessing of CLAHE and Canny Edge Detection as well implemented of VGG-19 and ResNet50 architectures of CNN method can produce good accuracy in ASL alphabet classification. The first best accuracy result has a value of 99.9%, second place with 99.82% accuracy, and third place has 98.93% accuracy. The three accuracy results have increased from previous studies with an accuracy of 98.63%.

REFERENCES

[1] W. Megawati and R. Hartono, “The Impact of Teachersâ€™ Verbal and Non-Verbal Communication on Studentsâ€™ Motivation in Learning English,” *English Educ. J.*, vol. 10, no. 4, pp. 436–448, Dec. 2020, doi: 10.15294/eej.v10i4.39157.

[2] R. Genç, “The Importance of Communication in Sustainability & Sustainable Strategies,” *Procedia Manuf.*, vol. 8, pp. 511–516, 2017, doi: 10.1016/j.promfg.2017.02.065.

[3] S. A. E. El-Din and M. A. A. El-Ghany, “Sign Language Interpreter System: An alternative system for machine learning,” in *2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES)*, Oct. 2020, pp. 332–337. doi: 10.1109/NILES50944.2020.9257958.

[4] M. Cocchi, A. Biancolillo, and F. Marini, “Chemometric Methods for Classification and Feature Selection,” 2018, pp. 265–299. doi: 10.1016/bs.coac.2018.08.006.

[5] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*. 2012.

[6] T. W. Edgar and D. O. Manz, *Research Methods for Cyber Security*. 2017.

[7] P. Rajendra Kumar and E. B. K. Manash, “Deep learning: a branch of machine learning,” *J. Phys. Conf. Ser.*, vol. 1228, no. 1, p. 012045, May 2019, doi: 10.1088/1742-6596/1228/1/012045.

[8] L. Alzubaidi *et al.*, “Review of deep learning: concepts, CNN architectures, challenges, applications, future directions,” *J. Big Data*, vol. 8, no. 1, p. 53, Mar. 2021, doi: 10.1186/s40537-021-00444-8.

[9] A. B. Triyadi, A. Bustamam, and P. Anki, “Deep Learning in Image Classification using VGG-19 and Residual Networks for Cataract Detection,” in *2022 2nd International Conference on Information Technology and Education (ICIT&E)*, Jan. 2022, pp. 293–297. doi: 10.1109/ICITE54466.2022.9759886.

[10] K. L. Kohsasih and B. H. Hayadi, “Classification SARS-CoV-2 Disease based on CT-Scan Image Using Convolutional Neural Network,” *Sci. J. Informatics*, vol. 9, no. 2, pp. 197–204, Nov. 2022, doi: 10.15294/sji.v9i2.36583.

[11] A. Bansal and N. Singh, “Image Enhancement Techniques: A Review,” *ASIAN J. Converg. Technol.*, vol. 6, no. 2, pp. 7–11, Aug. 2020, doi: 10.33130/AJCT.2020v06i02.002.

[12] W. Zhou, X. Ma, and Y. Zhang, “Research on Image Preprocessing Algorithm and Deep Learning of Iris Recognition,” *J. Phys. Conf. Ser.*, vol. 1621, no. 1, p. 012008, Aug. 2020, doi: 10.1088/1742-6596/1621/1/012008.

[13] S. Kaslay, T. Kesarkar, and K. Shinde, “ASL GESTURE RECOGNITION USING VARIOUS FEATURE EXTRACTION TECHNIQUES AND SVM,” *Int. Res. J. Eng. Technol.*, vol. 7, no. 6, 2020.

[14] K. Y. Lum, Y. H. Goh, and Y. Bin Lee, “American Sign Language Recognition Based on MobileNetV2,” *Adv. Sci. Technol. Eng. Syst. J.*, vol. 5, no. 6, pp. 481–488, Nov. 2020, doi: 10.25046/aj050657.

[15] Y. Ma, T. Xu, and K. Kim, “Two-Stream Mixed Convolutional Neural Network for American Sign Language Recognition,” *Sensors*, vol. 22, no. 16, p. 5959, Aug. 2022, doi: 10.3390/s22165959.

[16] G. Yadav, S. Maheshwari, and A. Agarwal, “Contrast limited adaptive histogram equalization based enhancement for real time video system,” in *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Sep. 2014, pp. 2392–2397. doi: 10.1109/ICACCI.2014.6968381.

[17] G. F. C. Campos, S. M. Mastelini, G. J. Aguiar, R. G. Mantovani, L. F. de Melo, and S. Barbon, “Machine learning hyperparameter selection for Contrast Limited Adaptive Histogram Equalization,” *EURASIP J. Image Video Process.*, vol. 2019, no. 1, p. 59, Dec. 2019, doi: 10.1186/s13640-019-0445-4.

[18] C. Rubini* and N. Pavithra, “Contrast Enhancementof MRI Images using AHE and CLAHE Techniques,” *Int. J. Innov. Technol. Explor. Eng.*, vol. 9, no. 2, pp. 2442–2445, Dec. 2019, doi: 10.35940/ijitee.B7017.129219.

[19] X. Ma, B. Li, Y. Zhang, and M. Yan, “The Canny Edge Detection and Its Improvement,” 2012, pp. 50–58. doi: 10.1007/978-3-642-33478-8_7.

[20] D. Sundani, S. Widiyanto, Y. Karyanti, and D. T. Wardani, “Identification of Image Edge Using Quantum Canny Edge Detection Algorithm,” *J. ICT Res. Appl.*, vol. 13, no. 2, pp. 133–144, Sep. 2019, doi: 10.5614/itbj.ict.res.appl.2019.13.2.4.

[21] S. T. H. Kieu, A. Bade, and M. H. A. Hijazi, “Modified canny edge detection technique for identifying endpoints,” *J. Phys. Conf. Ser.*, vol. 2314, no. 1, p. 012023, Aug. 2022, doi: 10.1088/1742-6596/2314/1/012023.

[22] M. Hashemi, “Enlarging smaller images before inputting into convolutional neural network: zero-

padding vs. interpolation," *J. Big Data*, vol. 6, no. 1, p. 98, Dec. 2019, doi: 10.1186/s40537-019-0263-7.

[23] V. R. Joseph, "Optimal ratio for data splitting," *Stat. Anal. Data Min. ASA Data Sci. J.*, vol. 15, no. 4, pp. 531–538, Aug. 2022, doi: 10.1002/sam.11583.

[24] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," *Insights Imaging*, vol. 9, no. 4, pp. 611–629, Aug. 2018, doi: 10.1007/s13244-018-0639-9.

[25] A. Anton, N. F. Nissa, A. Janiati, N. Cahya, and P. Astuti, "Application of Deep Learning Using Convolutional Neural Network (CNN) Method For Women's Skin Classification," *Sci. J. Informatics*, vol. 8, no. 1, pp. 144–153, May 2021, doi: 10.15294/sji.v8i1.26888.

[26] M. M. Taye, "Theoretical Understanding of Convolutional Neural Network: Concepts, Architectures, Applications, Future Directions," *Computation*, vol. 11, no. 3, p. 52, Mar. 2023, doi: 10.3390/computation11030052.

[27] J. Feng and S. Lu, "Performance Analysis of Various Activation Functions in Artificial Neural Networks," *J. Phys. Conf. Ser.*, vol. 1237, no. 2, p. 022030, Jun. 2019, doi: 10.1088/1742-6596/1237/2/022030.

[28] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation Functions: Comparison of trends in Practice and Research for Deep Learning," Nov. 2018, [Online]. Available: http://arxiv.org/abs/1811.03378

[29] T.-H. Nguyen, T.-N. Nguyen, and B.-V. Ngo, "A VGG-19 Model with Transfer Learning and Image Segmentation for Classification of Tomato Leaf Disease," *AgriEngineering*, vol. 4, no. 4, pp. 871–887, Oct. 2022, doi: 10.3390/agriengineering4040056.

[30] M. Shafiq and Z. Gu, "Deep Residual Learning for Image Recognition: A Survey," *Appl. Sci.*, vol. 12, no. 18, p. 8972, Sep. 2022, doi: 10.3390/app12188972.