Applied Information System and Management (AISM)
Volume 6, (1) 2023, p. 39-46
P-ISSN: 2621-2536; E-ISSN: 2621-2544; DOI: 10.15408/aism.v6i1.29820

Classification of Sign Language in Real Time Using Convolutional Neural Network

Moh. Badri Tamam[1], Hozairi[2], Miftahul Walid[3], Januario Freitas Araujo Bernardo[4]

***Abstract*— Communication between people is essential for daily life activities. However, humans are created with their own strengths and weaknesses. One of them is the difficulty of communication and interaction for people with hearing and speech impairments. Sign language is a language for people who have difficulty hearing and speaking. However, sign language is not popular in society, and people who have it will have more difficulties. This research aims to classify hand gestures of sign language into letters using a convolutional neural network (CNN). The dataset is obtained from Kaggle, with a total of 34,627 data divided by the ratio of training and testing data of 80:20. From the test results, the letters of the alphabet that can be translated are: A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, S, T, U, V, W, X, Y, and Z. Furthermore, validation accuracy is obtained. In this study, a very high validation accuracy was obtained. The easiest letters to guess are V and N, while the most difficult letters to guess are n, c, j, and z. With different preprocessing, the loss value can be reduced, giving a higher accuracy of 95.4%.**

***Index Terms*—Sign language, classification, CNN, python.**

I. INTRODUCTION

Sign language users around the world, more or less. Every country, even every region, has its own sign language. On one side of the numbers owned by the general public, the ability to communicate with sign language is very limited [1]. The problem will arise when the person is deaf or speech-impaired and people want to communicate with ordinary people who do not understand sign language. Those who can hear can learn and understand written language as a representation of spoken language, using writing to encode phonemes [2]. However, for deaf people, this correspondence cannot be done because writing is only for the deaf, who have great difficulty in reading and writing because there is no correspondence directly between their natural languages (sign language) and written language [3].

Difficulty in communicating will affect life and interpersonal relationships in society. Difficulty in communicating between the deaf and the deaf who can hear can cause problems in the integration process of deaf people into society at large. Of course, it is necessary to look for a solution to this problem so that the communication process with ordinary people can go smoothly, which will improve harmonious relations between components of society [4].

The need for an interpreter from sign language into written language gets very big. One example was who used Doppler radar, and later VGG got 87.5% accuracy [5]. Other was also used to classify using a cellphone camera. Classification gets 92.8% as a result of its accuracy. Therefore, the focus of this study is to provide an overview of the implementation of the classification of sign language in real-time using CNN [6].

OpenCV (Open-Source Computer Vision Library) is one of the libraries of software intended for dynamic image processing in real time, built by Intel and now supported by Willow Garage and Itseez. OpenCV is released under the permissive BSD license, which is more accessible than the GPL and gives complete freedom with no need for commercial use of the source code. He also has a language-enabled interface for programming C++, C, Python, and Java, including on Windows, Linux, Mac OS, iOS, and Android. OpenCV is designed for efficiency in computing and focuses on real-time applications [7].

OpenCV is an example of implementing OpenCV in Python. A camera installed in the parking lot can read the number plate. This number plate is converted from analogue to digital and then processed into characters to become usable data with the necessary information. In essence, OpenCV, along with Python, is used to process images or videos (stack frames or images) according to their respective purposes, which involves the camera capturing the image and then processing it on the computer [8].

In addition to Python, OpenCV can be used in C++ and Java, but the easiest of the three is Python due to its simplicity.I've been studying C++ for OpenCV, and it turns out the code is quite complicated [9]. OpenCV, also called the Open-Source Computer Vision Library, is an open-source library for computer vision, machine learning, and image processing. It

Received: 21 December 2022; Revised: 1 February 2023; Accepted: 17 February 2023

[1]M. B. Tamam, Informatics Engineering Study Program, Islamic University Madura, Indonesia (e-mail: badri.uimadura@gmail.com).
[2]Hozairi, Informatics Engineering Study Program, Islamic University Madura, Indonesia (e-mail: dr.hozairi@gmail.com).
[3]M. Walid, Informatics Engineering Study Program, Islamic University Madura, Indonesia (e-mail: miftahwalid@gmail.com).
[4]J. F. A. Bernardo, University of Dili, Timor-Leste (e-mail: essadejaneiro89@gmail.com).

plays a significant role in real-time operation, which is essential in today's systems. For example, one can process images and videos to identify objects, faces, or even the handwriting of a human using OpenCV. The data can be in several forms: video sequences, views from multiple cameras, multidimensional data from a 3D scanner, or even a medical scanning device [10].

One method that can be used to measure the performance of a classification system is the confusion matrix. The confusion matrix shows a comparison between how the system classified the results and how they should have been classified. The confusion matrix is a two-dimensional table with real data in the columns and predicted data in the rows. It has its own class, with the real data in the columns and the predicted data in the rows [11]. This study attempts to determine how well CNN models perform at classifying hand sign language into letters using letters that are easy to translate or predict.

II. Related Work

In many fields, including image processing, CNN is crucial. It has a significant impact on numerous fields. CNN is employed in nanotechnologies like semiconductor production for defect detection and classification.

Handwritten digit recognition has become an issue of interest among researchers. There are a large number of papers and articles being published these days about this topic. Research shows that deep learning algorithms like multilayer CNN using Keras with Theano and TensorFlow give the highest accuracy compared with the most widely used machine learning algorithms like SVM, KNN, and RFC. Because of their highest accuracy, CNN is being used on a large scale in image classification and video. Many researchers are trying to make sentiment recognition in a sentence. CNN is used in natural language processing and sentiment recognition by varying parameter.

Deep learning (DL) has amazing abilities and smart computer vision, which are important for modelling complex data like image data. CNN is one of the deep learning methods that currently has the best results in image recognition. CNN tries to copy the system for recognising images in the visual cortex of the human brain so that it can process information about images. Learning has a weakness, namely the process of ancient training CNN is a deep neural network algorithm, the most commonly applied to analyse visual images. CNN is a multilayer perceptron. Its neurons are connected to all the neurons in the next layer. However, CNN found a hierarchical pattern in the data and aggregated those pixels as more complex than smaller and more simple pixels. Hence, CNN's performance comes from its connectedness, and the complexity of image pixels is excellent [12].

Deep learning is a subfield of machine learning. It was inspired by the structure and function of the brain and concerned with similar algorithms. These algorithms, inspired by the human brain, learn from large amounts of data just like we humans learn from experience [1]. The deep learning algorithms repeat the process each time, tweaking it a little to improve the outcome [4].

The CNN algorithm is based on biological processes, and connectivity between neurons resembles the organisation of the animal visual cortex. CNN uses less preprocessing in comparison with other image classification algorithms [13]. CNN learns filters in image algorithms normally. CNN is already widely used in applications of image recognition and video, providing system recommendations, image classification, image analysis medicine, and natural language processing [14].

DL is a type of machine learning that is mostly used to classify images, find objects, and process natural language. DL is an algorithm for automatically choosing data features that is based on a neural network. It does not need a lot of artificial feature engineering. Instead, it combines low-level features to take form.

The convolutional neural network is a feed-forward artificial neural network that is based on the structure of the visual cortex of animals. They can be used for a lot of things, like recognising images and videos, making recommendations, and processing natural language. As shown, CNN is generally made up of two main parts: convolutional layers and pooling layers. The convolutional layer is the most important part of a CNN. It creates feature maps by finding the dot product between a local area in the feature maps that came in and a filter. Each of the feature maps is followed by a nonlinear function, such as the Rectified Linear Unit (ReLU) nonlinearity, which is often used because it is easy to calculate. The pooling layer reduces the number of samples in feature maps by figuring out the maximum or average value for each sub-region. Usually, the fully connected layers follow several stacked convolutional and pooling layers, and the last fully connected layer is the softmax layer computing the scores for each class.

Yann LeCun first introduced CNN in 1988. CNN is one method that initiates the occurrence of deep learning. The difference between CNN and ANN is that CNN has architecture-optimized additions to existing features on the input image. The main components of CNN include: [15]

- Input layer
- Convolution Layer
- Activation Function
- Unification Layer
- Fully Connected Layers

It has many advantages, especially for machine learning-based programming. Like generally Java and C languages, which are open source, interactive, modular, dynamic, object-based, and others, the Python language has many libraries that can easily be used for machine learning, such as Numpy (for operations on vectors and matrices) [16], Scikit-learn (data analysis and statistics), Pandas Data Frame, Matplotlib (graphical data visualization), and Keras (a neural API network that works on top of Tensor Flow or Theano) [6].

OpenCV is an open-source library with the specific purpose of performing image processing. The bottom line is that computers have similar capabilities to human visual processing. OpenCV has provided many basic computer vision algorithms [5]. OpenCV also provides an object detection module that uses

Applied Information System and Management (AISM)
Volume 6, (1) 2023, p. 39-46
P-ISSN: 2621-2536; E-ISSN: 2621-2544; DOI: 10.15408/aism.v6i1.29820

the Viola-Jones algorithm [7].

Python is a high-level programming language that is interpreter-driven, interactive, object-oriented, and can operate on almost all platforms: Mac, Linux, and Windows. Python is an easy-to-learn programming language due to its clear syntax and can be combined with the use of ready-to-use modules, and efficient high-level data structures [8]. Tools are required to calculate and analyse enormous amounts of data in a more effective and efficient manner. In fact, Python is one of the suggested tools. The enhanced library support for Python in recent years, particularly for pandas, has made it a potent choice for data analytic jobs [9]. Computer vision, when all is said and done, means to double (or, in a few cases, compensate) human vision and, customarily, has been utilised as a part of performing routine monotonous undertakings, for example, classification in monstrous mechanical production systems [17]. Today, scrutinise on machine vision is spreading gigantically, so it is very nearly impossible to organise every last bit of its subtopics. Notwithstanding this, one can rundown a few important provisions, for example, face processing (i.e., gesture recognition and facial expression), machine human cooperation, swarm reconnaissance, and substance-based picture recovery [18].

All of the applications stated above require detection of faces, which can be simply viewed as a preprocessing step for obtaining the "object". The face is our primary centre of consideration in social life, assuming an imperative role in passing on feelings and character. We can perceive various appearances adapted all around our lifespan and distinguish faces considerably after numerous years of division [19].

The weighted sum of all the neurons in a layer becomes the input of a neuron in the next layer, adding a biassed value. In CNN, the layer has three dimensions. Here, all the neurons are not fully connected. Instead, every neuron in the layer is connected to the local receptive field [20]. A cost function is generated in order to train the network. It compares the output of the network with the desired output. The signal propagates back to the system again and again to update the shared weights and biases in all the receptive fields to minimise the value of the cost function, which increases the network's performance [21].

Matplotlib is for creating more interactive data visualization that is easy to read and easy to analyze. Visualization is what it does: it changes the data. With Python, rigid tables can be converted into graphic shapes capable of displaying changes and differences in data more clearly.The result of the data analysis displays statistics [22].

In contrast with proprietary CAPTCHA schemes, those open-source CAPTCHA libraries may pose greater security risks to powerful learning machines. Our work has demonstrated a scenario where a malicious learning machine can be trained infinitely through those tampered open-source libraries to perform such a chosen-plaintext attack. That is, a chosen-plaintext attack against CAPTCHAs can be performed by making unlimited use of an encryption machine (open-source libraries) for generating well-labeled ciphertext (labeled CAPTCHAs) for a malicious learning machine [23].

III. Research Method

This methodology enables thorough and succinct research. A description of the dataset, data processing, and coding will be given at the outset of learn. Problem identification comes first in the research process, then a literature review, data collecting, and data entry during preprocessing. Preprocessing is followed by the system design phase, which refers to the CNN model, and implementation is followed by testing and model performance (Fig. 1).

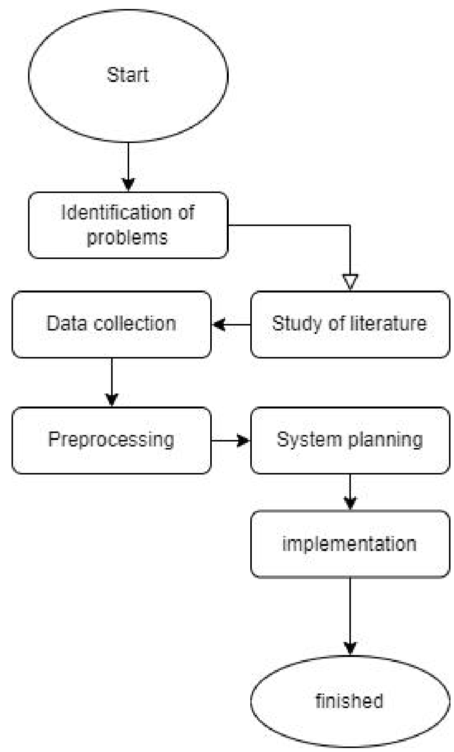


Figure 1. Research Stages

A. *Identification of problems*

There will still be issues for deaf persons who cannot communicate adequately because speech helps us communicate. In the identification phase of the research project on recognising handwritten alphabet letters using images and CNN [5], [24].

B. *Study of literature*

At this stage, the author conducted a literature study for research, namely collecting data from Kaggle.

C. *Dataset*

The Kaggle dataset, which consists of images of people posing their hands as verbal clues, was utilised. Each image in this collection has a size of 200×200 pixels and consists of 78,500 photos totaling 1.04 GB. There are 26 classes in this collection, which include all 26 letters of the alphabet (from "A" to "Z"). There are three additional classes in this dataset called "none," "del," and "space," however they were not included in this study because they were not necessary.

D. System planning

At this stage, the researchers designed a classification model using the CNN VGG-16 architectural method.

E. Processing

Image preprocessing is done using some libraries from Python, including NumPy, CV2, Matplotlib, and Pyplot. The data that is input for the first time is in the form of images (Fig. 3).

```
In [3]: plt.imshow(img)
Out[3]: <matplotlib.image.AxesImage at 0x1b6aa56e848>
```

Figure 2. Processed Image Data

After the image enters the second process, i.e., including the 785 datasets taken from Kaggle as well (Fig. 3).

```
count the number of rows
print(train.shape)
print(test. shape)
(27455, 785)
(7172, 785)

put train data & test data into the array
train_set = np.array(train, type = 'float32')
test_set = np.array(test, dtype= 'float32')
train.head()
```

	label	pixel1	pixel2	pixel3	pixel4	pixel5	pixel6	pixel7	pixel8	pixel9
0	3	107	118	127	134	139	143	146	150	153
1	6	155	157	156	156	156	157	156	158	158
2	2	187	188	188	187	187	186	187	188	187
3	2	211	211	212	212	211	210	211	210	210
4	13	164	167	170	172	176	179	180	184	185

5 rows × 785 columns

Figure 3. Datasets

```
create class
class_names =['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H',
'I', 'J', 'K', 'L' , 'M', 'N', 'O', 'P', 'Q', 'R', 'S',
'T', 'U', 'V', 'W', 'X',' Y']
```

After that, the process of entering training and test data into the array, creating a class name, and displaying random data to match all image dimensions began. Then you'll get the size of the SAS train, choose a random data train starting at 0 and show the chosen image (Figs. 4 and 5).

```
(-0.5, 27.5, 27.5, -0.5)
```

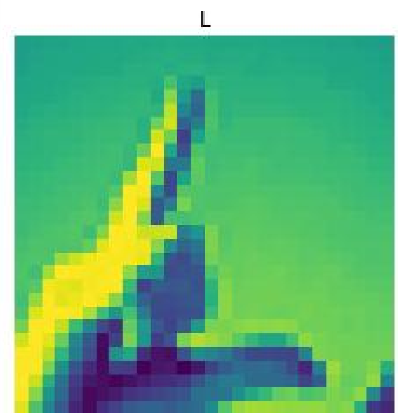


Figure 4. Displays Images Randomly

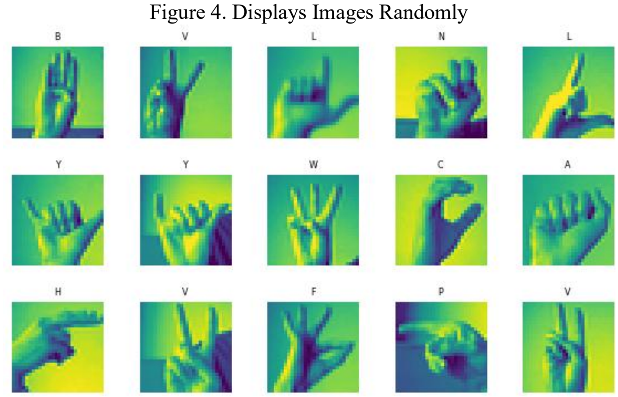


Figure 5. Displays The Selected Image

The next process is preparing training and test data, which are displayed in Figs. 6 and 7.

```
X_train = train_set[:, 1:] / 255
y_train = train_set[:, 0]
```

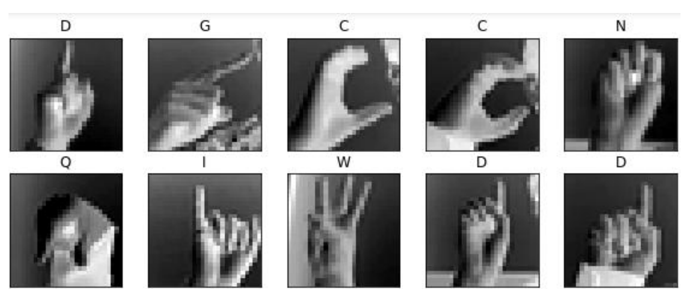


Figure 6. Visualization of Training Images I and C

Applied Information System and Management (AISM)
Volume 6, (1) 2023, p. 39-46
P-ISSN: 2621-2536; E-ISSN: 2621-2544; DOI: 10.15408/aism.v6i1.29820

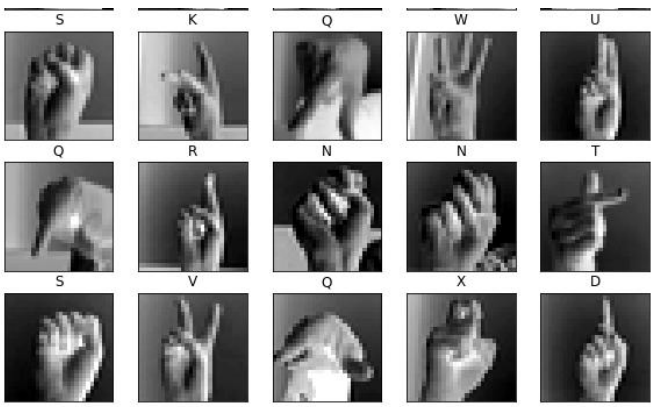


Figure 7. Visualization of Training Images V, W, L, I, and O

F. Implementation

In order to make the alphabet operate at this point, the researcher employed the Python programming language along with the Tensorflow and Keras frameworks. To determine accuracy and the F1 score, the researcher assesses the model that was created using the confusion matrix.

CNN model results from
cnn_model. summary()

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape
   Param #
=================================================================
conv2d (Conv2D)              (None, 26, 26, 32)
   320
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 13, 13, 32)
   0
_________________________________________________________________
acc: 0.9685 - val_loss: 0.0108 - val_acc: 0.9996
Epoch 12/55
21964/21964 [==============================] - 61s 3
ms/sample - loss: 0.0910 - acc: 0.9715 - val_loss: 0.
0077 - val_acc: 0.9998
Epoch 13/55
21964/21964 [==============================] - 60s 3
ms/sample - loss: 0.0846 - acc: 0.9730 - val_loss: 0.
0064 - val_acc: 1.0000
Epoch 14/55
21964/21964 [==============================] - 62s 3
ms/sample - loss: 0.0771 - acc: 0.9753 - val_loss: 0.
0052 - val_acc: 1.0000
Epoch 15/55
21964/21964 [==============================] - 61s 3
ms/sample - loss: 0.0715 - acc: 0.9769 - val_loss: 0.
0041 - val_acc: 1.0000
Epoch 16/55
21964/21964 [==============================] - 61s 3
ms/sample - loss: 0.0638 - acc: 0.9799 - val_loss: 0.
0033 - val_acc: 1.0000
Epoch 22/55
Epoch 28/55
21964/21964 [==============================] - 62s 3
ms/sample - loss: 0.0334 - acc: 0.9897 - val_loss: 6.
4232e-04 - val_acc: 1.0000
Epoch 29/55
21964/21964 [==============================] - 61s 3
ms/sample - loss: 0.0330 - acc: 0.9891 - val_loss: 7.
3649e-04 - val_acc: 1.0000
Epoch 30/55
21964/21964 [==============================] - 61s 3
ms/sample - loss: 0.0321 -
21964/21964
21964/21964 [==============================] - 60s 3
ms/sample - loss: 0.0229 - acc: 0.9929 - val_loss: 4.
2094e-04 - val_acc: 1.0000
Epoch 42/55
21964/21964 [==============================] - 60s 3
ms/sample - loss: 0.0219 - acc: 0.9930 - val_loss: 2.
3793e-04 - val_acc: 1.0000
21964/21964 [==============================] - 60s 3
ms/sample - loss: 0.0196 - acc: 0.9934 - val_loss: 3.
3073e-04 - val_acc: 1.0000
21964/21964 [==============================] - 62s 3
ms/sample - loss: 0.0173 - acc: 0.9951 - val_loss: 1.
0458e-04 - val_acc: 1.0000
Epoch 54/55
21964/21964 [==============================] - 61s 3
ms/sample - loss: 0.0175 - acc: 0.9948 - val_loss: 1.
6882e-04 - val_acc: 1.0000
Epoch 55/55
21964/21964 [==============================] - 61s 3
ms/sample - loss: 0.0157 - acc: 0.9953 - val_loss: 1.
3220e-04 - val_acc: 1.0000
```

Researchers use K-fold cross calidation with 55 folds and 55 iterations. Evaluation will be carried out using the confusion matrix method to calculate accuracy and the F1 score.

IV. Result

Implementation is a stage of preprocessing that involves turning images into frames and assigning an alphabetic gesture to the part of the hand that is being shown.Then, cropping is performed on the gesture area. The final result of preprocessing will be stored in the system's internal storage. The results indicate that CNN is able to recognise the alphabet successfully. In the dataset training process, we used 416 images with epoch 55 and batch size 32 to get the same model optimised with a training precision of 95.4%. There is a good match between the accuracy of the training plot and the accuracy of the validation plot shown in the graph, where the accuracy increases gradually and with a small difference. The test uses the k-fold cross-validation technique with a 5-fold amount. Results from testing are calculated using the confusion matrix to get the accuracy, recall, precision, and F1 score. The number of true positives is 42, according to the classification evaluation using the confusion matrix. With the recognition of the sign language letters L, S, M, N, and V, the evaluation calculation's results showed an accuracy of 95.4%.

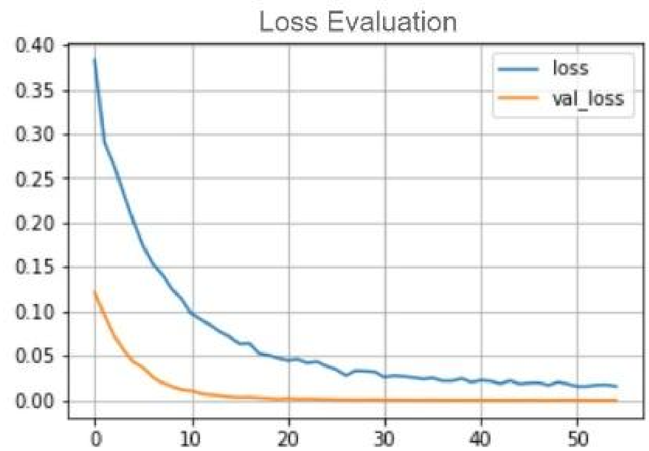


Figure 8. Loss Evaluation

As seen in Figure 8, the evaluation of loss is very good, with a loss of 55 and a val_loss of 55. It can be known that the higher the number of ages, the better the accuracy value. Here are the test results in the training model: a graphical accuracy model of process data training with a total of 55 epochs, shown in Figure 7 of the epochal results for the resulting accuracy of 99%, close to perfect.

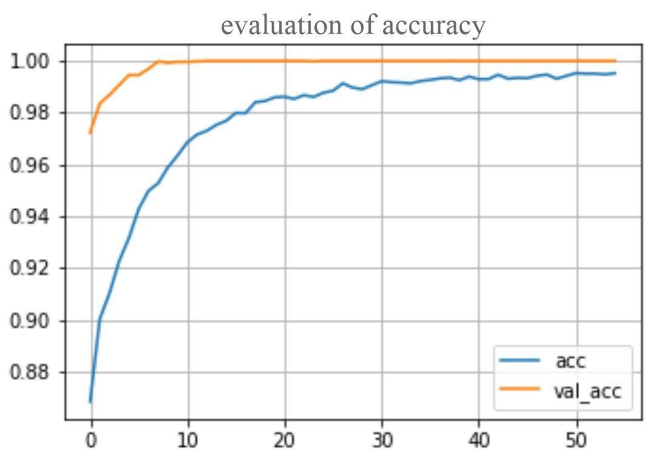
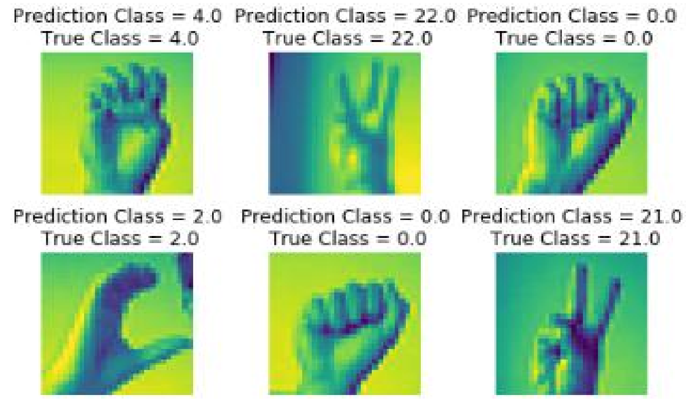


Figure 9. Accuracy Evaluation

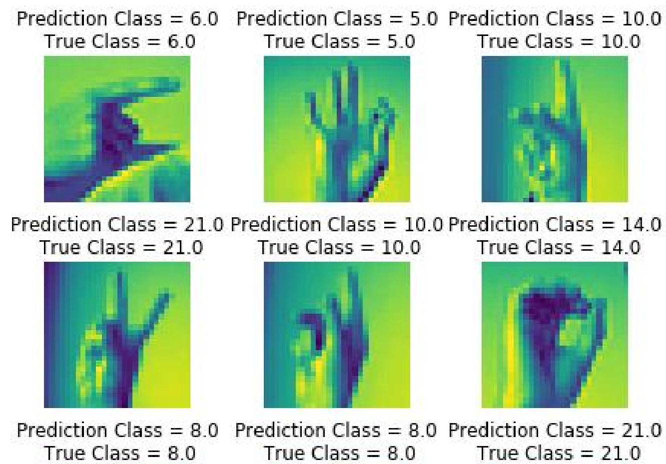
In Fig. 9, the two graphs do not show many differences in accuracy depending on whether a horizontal flip is used or not. Moreover, both feature graphics with good accuracy on training, but on validation, the graph is not stable. After knowing the results of the epoch, the process of visualising new prediction results is carried out with `subplots_adjust(wspace=0.5)`.

Figure 10 illustrates that while the test performed on the dataset itself yielded the right prediction results, this is not necessarily the case with other images. This approach produces classification results with 95% accuracy.

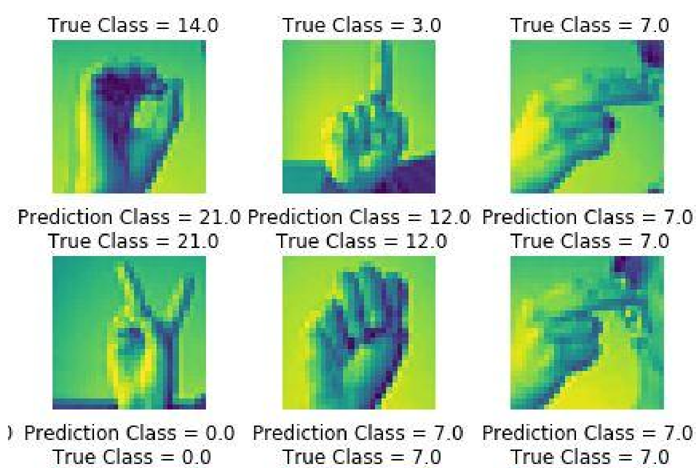
Several studies have been done in the past. One of them is recognising sign language using the Haar classifier tracking method and classifying training image data sets with the K Nearest Neighbors algorithm. The accuracy of this app's recognition is 89.6%, and it can only tell the difference between 19 of the 26 existing letters. Some of the letters it can't tell apart are M, N, S, S, T, J, and Z [5]. This is due to the high level of similarity between the letters of the alphabet. This is due to the high level of similarity between sign letters.



(a)



(b)



(c)

Figure 10. Prediction Results

Real-time sign language recognition applications are still being studied, but there are still a lot of problems. One of the things that affects the results is the tracking process, especially in separating the object from the background.The feature extraction process is carried out from forms that are unable to distinguish sign characters due to the high level of similarity of training image data and the use of auxiliary methods that are less than optimal in recognising each sign character. In comparison to previous research, the current research uses the tracking method, namely, using many hand styles in real-time with the CNN and expecting to be able to track hand movements from each frame to minimise image dimensions. The results can provide the ability to improve accuracy, feature extraction, and real-time hand movements in various poses according to the image dimensions. hand movements in real-time with various poses in accordance with the Indonesian Sign Language System with much better accuracy than before.

V. Conclusion

CNN models are created using several layers such as Cv2, split, matloplob, and numpay. Several things have been done to find out the difference in accuracy between pre-processing data and real-time processing. From this research, the accuracy result is 95.4%. There is also a high loss during validation accuracy, while the loss in training is quite stable. The best-predicted letters are v and n, while the worst are n, c, j, and z.

For future development, several things need to be done to improve the model, including changing the layer and doing different preprocessing to reduce the loss value and provide higher accuracy. It also uses more accurate methods by looking at how the motion changes when the frame is found and recognized. In this paper, a new architecture called CNN-CapsNet is proposed to deal with the task of remote sensing image scene classification. It is based on deep CNN's strong ability to learn features and CapsNet's property of equivariance. The goal is to improve the accuracy of remote sensing image scene classification. The proposed architecture is composed of two parts. First, a pre-trained deep CNN, such as VGG-16, is fully trained on the ImageNet dataset, and its intermediate convolutional layer is used as an initial feature map extractor. Then, the initial feature maps are fed into a newly designed CapsNet to label the remote-sensing image scenes. Experimental results on three challenging benchmark datasets show that the proposed architecture achieves higher accuracy than state-of-the-art methods.

References

[1] Z. K. S. Domas and R. Rakhmadi, “Peningkatan Performa Decision Tree dengan AdaBoost untuk Klasifikasi Kekurangtransparanan Informasi Anti-Korupsi,” *Appl. Inf. Syst. Manag.*, vol. 5, no. 2, pp. 75–82, Oct. 2022, doi: 10.15408/aism.v5i2.24887.

[2] A. Oktavianto and S. F. Persada, “Persepsi Publik Tentang Pembelajaran Daring di Indonesia: Studi Menggunakan ELK Stack dan Python untuk Analisis Sentimen di Twitter,” *J. Tek. ITS*, vol. 9, no. 2, pp. A170–A175, 2021, doi: 10.12962/j23373539.v9i2.54277.

[3] S. Mujilahwati, “Visualisasi Data Hasil Klasifikasi Naïve Bayes Dengan Matplotlib Pada Python,” *Pros. SNST Fak. Tek.*, vol. 1, no. 1, pp. 205–211, 2021, [Online]. Available: https://publikasiilmiah.unwahas.ac.id/index.php/PROSIDING_SNST_FT/article/view/5164%0Ahttps://publikasiilmiah.unwahas.ac.id/index.php/PROSIDING_SNST_FT/article/download/5164/3787.

[4] P. A. Winata, “Klasifikasi Naive Bayes Keparahan Trauma Pasien Menggunakan Data Neuro Cognitive Dan Data Physiologic dengan Python,” *Seminar Nas. Mat. Geom. Stat. dan Komputas*, 2022, [Online]. Available: https://jurnal.unej.ac.id/index.php/prosiding/article/view/33500/11662.

[5] S. Mujilahwati, M. Sholihin, and R. Wardhani, “Optimasi Hyperparameter TensorFlow dengan Menggunakan Optuna di Python: Study Kasus Klasifikasi Dokumen Abstrak Skripsi,” *J. Media Inform. Budidarma*, vol. 5, no. 3, pp. 1084–1089, 2021, doi: 10.30865/mib.v5i3.3090.

[6] L. A. Septiandi, E. M. Yuniarno, and A. Zaini, “Deteksi Kedipan dengan Metode CNN dan Percentage of Eyelid Closure (PERCLOS),” *J. Tek. ITS*, vol. 10, no. 1, pp. A56–A63, 2021, doi: 10.12962/j23373539.v10i1.61174.

[7] W. Kurniawan and A. Harjoko, “Pengenalan Bahasa Isyarat dengan Metode Segmentasi Warna Kulit dan Center of Gravity,” *Indones. J. Electron. Instrum. Syst.*, vol. 1, no. 2, pp. 67–78, 2011, doi: 10.22146/ijeis.1964.

[8] B. Nugroho and E. Y. Puspaningrum, “Kinerja Metode CNN untuk Klasifikasi Pneumonia dengan Variasi Ukuran Citra Input,” *J. Teknol. Inf. dan Ilmu Komput.*, vol. 8, no. 3, pp. 533–538, 2021, doi: 10.25126/jtiik.2021834515.

[9] A. A. Gafar and J. Y. Sari, “Sistem Pengenalan Bahasa Isyarat Indonesia dengan Menggunakan Metode Fuzzy K-Nearest Neighbor,” *J. Ultim.*, vol. 9, no. 2, pp. 122–128, 2018, doi: 10.31937/ti.v9i2.671.

[10] A. Willyanto, D. Alamsyah, and H. Irsyad, “Identifikasi Tulisan Tangan Aksara Jepang Hiragana Menggunakan Metode CNN Arsitektur VGG-16,” *J. Algoritm.*, vol. 2, no. 1, pp. 1–11, Oct. 2021, doi: 10.35957/algoritme.v2i1.1450.

[11] N. Sazqiah *et al.*, “Pengenalan Aksara Lampung Menggunakan Metode CNN (Convolutional Neural Network),” *Semin. Nas. Ins. Prof.*, vol. 2, no. 1, pp. 1–5, 2022, doi: 10.23960/snip.v2i1.165.

[12] I. Rahayuningsih, A. D. Wibawa, and E. Pramunanto, “Klasifikasi Bahasa Isyarat Indonesia Berbasis Sinyal EMG Menggunakan Fitur Time Domain (MAV, RMS, VAR, SSI),” *J. Tek. ITS*, vol. 7, no. 1, pp. A175–A180, 2018, doi: 10.12962/j23373539.v7i1.29967.

[13] N. A. Hasma, F. Arnia, R. Muharar, and M. K. Muchamad, “Pengenalan Gerakan Isyarat Bahasa Indonesia Menggunakan Algoritma SURF dan K-Nearest Neighbor,” *KITEKTRO: Jurnal Komputer, Informasi Teknologi dan Elektro*, vol. 7, no. 1, pp. 50–54, 2022.

[14] A. Lianardo, “Klasifikasi Gejala Penyakit Daun pada Tanaman Singkong Berbasis Vision Menggunakan Metode CNN dengan Arsitektur Mobilenet,” *e-Proceeding of Engineering*, vol. 8, no. 6, pp. 3176–3179, 2022, doi: 10.34818/eoe.v9i6.18980.

[15] Darmatasia, “Pengenalan Sistem Isyarat Bahasa Indonesia (SIBI) Menggunakan Gradient-Convolutional Neural Network,” *J. Instek*, vol. 6, no. 1, pp. 56–65, 2021, doi: 10.24252/instek.v6i1.18637.

[16] R. I. Borman, B. Priyopradono, and A. R. Syah, “Klasifikasi Objek Kode Tangan pada Pengenalan Isyarat Alphabet Bahasa Isyarat Indonesia (BISINDO),” *Semin. Nas. Inform. dan Apl.*, no. September, pp. 1–4, 2018.

[17] F. Devina, C. Citra, and E. Tanuwijaya, “Klasifikasi Bahasa Isyarat Amerika menggunakan Convolutional Neural Network,” *J. Sist. dan Teknol. Inf.*, vol. 10, no. 1, pp. 139–144, Jan. 2022, doi: 10.26418/justin.v10i1.47184.

[18] S. Ependi *et al.*, “Klasifikasi Pendeteksi Wajah Berhijab Mengunakan Metode CNN (Convlutional Neural Network),” *J. Pendidik. Tambusai*, vol. 6, no. 1, pp. 3157–3164, 2022, [Online]. Available: https://jptam.org/index.php/jptam/article/view/3363.

[19] 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, 2021, doi: 10.15294/sji.v8i1.26888

[20] B. D. Prasetya, F. S. Pamungkas, and I. Kharisudin, “Pemodelan dan

Peramalan Data Saham dengan Analisis Time Series menggunakan Python," *Prism. Pros. Semin. Nas. Mat.*, vol. 3, pp. 714–718, 2020, [Online]. Available: https://journal.unnes.ac.id/sju/index.php/prisma/ISSN.

[21] R. Magdalena, S. Saidah, N. K. C. Pratiwi, and A. T. Putra, "Klasifikasi Tutupan Lahan Melalui Citra Satelit SPOT-6 dengan Metode Convolutional Neural Network (CNN)," *J. Edukasi dan Penelit. Inform.*, vol. 7, no. 3, pp. 335–339, 2021, doi: 10.26418/jp.v7i3.48195.

[22] V. Data, P. Covid, M. B. Tamam, and A. Hozairi, "Indonesia dan Malaysia Data Visualization of the Spread of Covid 19 in Indonesia and Malaysia," *Jurnal SimanteC*, vol. 11, no. 1, pp. 13–18, Dec. 2022, doi: 10.21107/simantec.v11i1.14252.

[23] N. Yu and K. Darling, "A Low-cost Approach to Crack Python CAPTCHAs using AI-based Chosen-plaintext Attack," *Appl. Sci.*, vol. 9, no. 10, pp. 1–17, 2019, doi: 10.3390/app9102010.

[24] N. Hanum Harani, C. Prianto, and M. Hasanah, "Deteksi Objek dan Pengenalan Karakter Plat Nomor Kendaraan Indonesia Menggunakan Metode Convolutional Neural Network (CNN) Berbasis Python," *J. Tek. Inform.*, vol. 11, no. 3, pp. 47–53, 2019.