**Journal of Artificial Intelligence and Engineering Applications**

*Website: https://ioinformatic.org/*

*15th October 2025. Vol. 5. No. 1; e-ISSN: 2808-4519*

# Hangul Handwritten One-Syllable Character Recognition Using CNN With ResNet Architecture and SVM

**Rena Rama Rosalinda[1]*, Basuki Rahmat[2], Eka Prakarsa Mandyartha[3]**

*[1,2,3]Universitas Pembangunan Nasional "Veteran" Jawa Timur*
*21081010190@student.upnjatim.ac.id[1*], basukirahmat.if@upnjatim.ac.id[2], eka_prakarsa.fik@upnjatim.ac.id[3]*

**Abstract**

Foreign language skills are one of the gateways to opening up opportunities for better education and employment. The use of technology can help with this, especially handwriting recognition technology. However, the use of limited datasets is often a problem. This study uses a Convolutional Neural Network (CNN) model with a Residual Network (ResNet) architecture and a Support Vector Machine (SVM). ResNet, as a feature extraction method for the data, is capable of capturing data patterns without losing much of the original data information. Meanwhile, the SVM algorithm, as a data classifier, is capable of working well with limited data. This research uses hyperparameters of linear kernel, polynomial kernel, Radial Basis Function (RBF) kernel, and Sigmoid kernel. Additionally, the hyperparameters C and Gamma values were also used. The research results indicate that the best model accuracy was obtained from the model trained with a linear kernel and a C value of 0.1, with an accuracy of 81.72% and an accuracy on the test data of 87.50%.

***Keywords**: Convolutional Neural Network (CNN), Handwritten Recognition, Residual Network (ResNet), Support Vector Machine (SVM)*

## 1. Introduction

Education and health are fundamental aspects that play a crucial role in sustaining human life. However, in recent times, the education sector in Indonesia has been facing significant challenges. Government budget efficiency policies have resulted in rising tuition fees for higher education. This situation does not align with the increasing difficulty of finding employment in Indonesia, compounded by the growing number of layoffs in various companies. As a result, job opportunities have become more limited, forcing recent university graduates to compete directly with experienced workers. These circumstances have led some Indonesians to seek better opportunities abroad, with South Korea being one of the preferred destinations. Several collaborative programs between the Indonesian and South Korean governments are available, including the Global Korea Scholarship (GKS) for higher education and the government-to-government employment placement program. However, participation in these programs requires language proficiency, particularly in reading and writing skills.

The differences between the Latin alphabet used in the Indonesian language and the Hangul writing system in Korean characters considerable challenges, especially in developing the ability to write in Hangul. The development of modern handwriting recognition technology is carried out to assist in learning foreign languages. Handwriting recognition or computer vision uses artificial intelligence such as deep learning in the process. The model trains training data and learns training data to predict new data. The model learns patterns in previous handwriting image data, then recognizes new handwriting image data based on the information that the model has learned. Handwriting recognition is divided into three types, namely digit recognition, word or letter recognition, and sentence recognition at a more advanced stage[1].

The use of the Convolutional Neural Network (CNN) model is very commonly used in image recognition cases or datasets, but CNN requires a lot of labeled data for training. A lack of data can prevent the model from learning representative patterns, causing CNNs to often experience overfitting because the model memorizes the training data examples, making it difficult to classify new data[2].
The use of Support Vector Machines (SVMs) can work well on small datasets because SVMs only require support vectors or several data points closest to the hyperplane to form a model. In addition, SVM is also capable of handling datasets with unbalanced data, but SVM is also susceptible to bias in highly unbalanced data, so the distribution of data for each class needs to be considered[3].
To obtain a good classification process, the use of a CNN model architecture to extract image data also needs to be considered. The ResNet architecture excels at extracting image data because it uses residual blocks, where the initial image data input information is added to the output in each block. This allows the model to learn complex features without losing important initial details. Unlike regular CNN models, which can change or lose all of their initial information when passing through many layers, this is very important to consider when creating models with limited data[4].

Based on the problems described above, this study will create a system that can recognize single-syllable Hangul handwritten characters using a Convolutional Neural Network (CNN) model with a 50-layer Residual Network (ResNet50) architecture and a Support Vector Machine (SVM) algorithm. This study will use secondary data from the SERI95 Hangul dataset, taking 24 classes of single-syllable Hangul characters. Each class contains 100 image data, with a total of 2400 image data in the dataset. The characters used are 뼈 (bbyeo), 빈 (bin), 봄 (bom), 책 (chaeg), 닭 (dalg), 뜰 (ddeul), 둘 (dul), 개 (gae), 강 (gang), 흘 (heul), 효 (hyo), 집 (jib), 칼 (kal), 코 (ko), 물 (mul), 명 (myeong), 높 (nop), 팬 (paen), 폐 (pye), 련 (ryeon), 류 (ryu), 승 (seung), 쌀 (ssal), and 텈 (teog).

# 2. Research Method

Methodology of this research consist of 7 main steps, it starts from literature review from previous work and method that used in this research to evaluation from model.

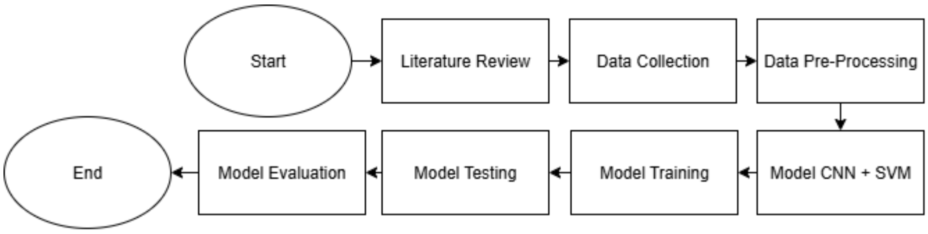


**Fig. 1:** Research Method

## 2.1 Literature Review

Previous research related to image recognition and the methods used in this study served as a reference for the selection of models and architectures. Sarwati et al. [5] conducted research on Arabic handwriting recognition using the ResNet50 model. The results showed that the model was able to classify test data with an accuracy rate of 95.03%. For training time efficiency, this study will apply the transfer learning method to the ResNet50 model for feature extraction. Fauzan Muhammad et al. [6] compared the ResNet50 model with another popular model, VGG16, using the transfer learning method. The results showed that ResNet50 excelled in short training time and achieved higher accuracy performance. Research using the method used in this study has previously been conducted in the field of image recognition in medicine by Farhana Islam et al. [7] The study tested several widely used transfer learning models, such as AlexNet, ResNet18, ResNet50, VGG16, and VGG19. The results showed that the model using the ResNet50 and SVM methods achieved the highest accuracy of 98.71%.

## 2.2 Data Collection

The data used in this study is secondary data with the following details:

**Table. 1:** Dataset Information

| No | Description | Dataset Information |
|---|---|---|
| 1 | Data Type | PNG |
| 2 | Total Data | 2.400 Images |
| 3 | Classes | 24 Classes/Labels |
| 4 | Data per Classes | 100 Images |
| 5 | Image Size | 100 x 100 Pixel |

The data used is a dataset of a collection of single-syllable Hangul handwriting images, a dataset with png data type, measuring 100x100 pixels, with a total of 2,400 image data.

## 2.3 Data Pre-processing

Dataset label originally used Hangul characters according to the image data, it is necessary to label the classes first using the Latin alphabet. The initial image size in the image data is 100 x 100 pixels, the image data size needs to be changed to 224 x 224 pixels to match the size of the image data in the dataset that has been trained with the transfer learning model, after that the data must be normalized. Data sharing in this research uses the Pareto principle concept, with a data sharing ratio of 80% training data and 20% data for testing. The concept of sharing data using the Pareto principle has been widely used in several studies, because it is effective in limited resources[8]. A total of 2,374 data were used for training data, and 1,160 data were used for testing data.

## 2.4 Model Convolutional Neural Network (CNN) dan Support Vector Machine (SVM)

The design of the model is a crucial stage in this research. After conducting a literature review, the selection of models, architectures, and algorithms can be carried out in a more systematic manner. This stage also enables the creation of visual representations of the applied models, which facilitate the understanding of processes occurring within the dataset during the training phase. Furthermore, a well-structured model design provides a solid foundation for evaluating performance, improving interpretability, and ensuring the reliability of the research.

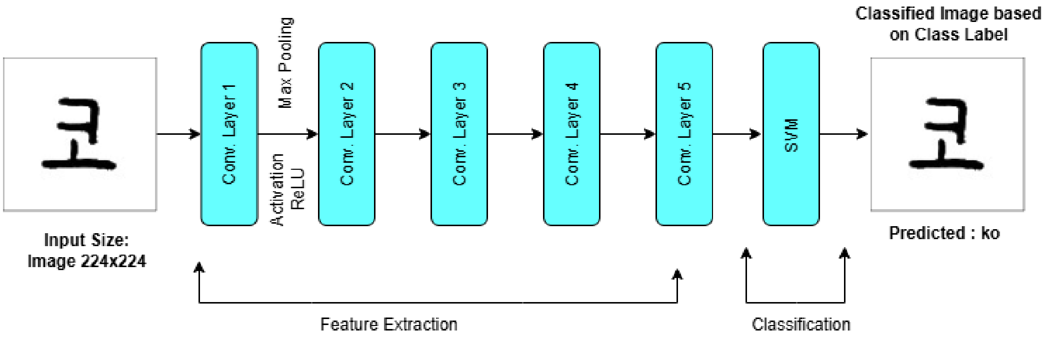


**Fig. 2:** Residual Network and Algoritm SVM

Figure 2 shows the design of the model used, the application of the Convolutional Neural Network (CNN) model with ResNet50 architecture using the transfer learning method as feature extraction in handwritten character images to analyze image patterns and shapes. After feature extraction, the final stage is the classification stage using the Support Vector Machine (SVM) algorithm based on the feature extraction that has been carried out using ResNet50.

## 2.5 Model Training

Model will be trained with Kernel, C, and Gamma hyperparameters to obtain optimal results. Table 2 shows the hyperparameters that will be used during model training.

**Table 2:** Hyperparameter Model

| Hyperparameter | Value |
|---|---|
| Kernel | 0.1, 1, 10, 100 |
| C | Linear, RBF, Poly, Sigmoid |
| Gamma | 0.1, 1, 10, 100 |

For linear kernels, several C values will be trained to obtain the best hyperparameter results. Meanwhile, for non-linear kernels such as RBF kernels, polynomial kernels, and sigmoid kernels, several C and Gamma values will be tested. The model will be trained using k-fold with a value of 5. K-fold is used to see if the model is able to generalize the data and prevent overfitting. K-fold uses a cross-validation process that divides all training data into validation data, so that the model can recognize the data well and not overfit.[9]. The equations related to each kernel are given in Equation 1 for the linear kernel, Equation 2 for the RBF kernel, Equation 3 for the polynomial kernel, and Equation 4 for the sigmoid kernel.

$$K(x_i, x) = (x_i x) \tag{1}$$

$$K(x_i, x) = \exp\left(-\gamma \parallel \mathrm{xi} - \mathrm{x} \parallel^2\right) \tag{2}$$

$$K(x_i, x) = (\gamma x_i x)^d \tag{3}$$

$$K(x_i, x) = \tanh(\gamma(x_i x) + c) \tag{4}$$

## 2.6 Model Testing

Model testing was conducted using new data that had been divided according to the Pareto principle, with 20% of the 1,160 data points in the dataset being tested. The SVM model that had been trained with the best accuracy based on several previous hyperparameters was tested using the test data and evaluated to determine whether the model could work with new data and was not overfitting due to over-recognition of the training data.

## 2.7 Model Evaluation

Model evaluation process will be assisted by confusion matrix calculations. The confusion matrix helps evaluate how well the model classifies Hangul handwriting images. The evaluation metric used in the confusion matrix is the accuracy metric. Accuracy provides an overview of the extent to which the model can make correct predictions. The equation for obtaining the accuracy value is shown in equation 5.

$$Accuracy = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN} + \mathrm{TN}} \tag{5}$$

# 3. Result and Discussion

The results of the research with several hyperparameters will be grouped according to each kernel, so that the best hyperparameter results from each kernel can be evaluated. The highest accuracy value in each kernel will be taken and re-evaluated to obtain the highest value from all kernels. The model selected is the value with the highest accuracy in all kernels, which will be selected for further use in the test data and evaluated with accuracy in the test data.

**Table 2:** Training Results with Linear Kernel Hyperparameters

| Parameter | Fold 1 | Fold 2 | Fold 3 | Fold 4 | Fold 5 | Mean | Standar Deviasi |
|---|---|---|---|---|---|---|---|
| C : 0.1 | 0.8286 | 0.8239 | 0.8391 | 0.8435 | 0.8043 | 0.8279 | 0.0137 |
| C : 1 | 0.8286 | 0.8239 | 0.8391 | 0.8435 | 0.8043 | 0.8279 | 0.0137 |
| C : 10 | 0.8286 | 0.8239 | 0.8391 | 0.8435 | 0.8043 | 0.8279 | 0.0137 |
| C : 100 | 0.8286 | 0.8239 | 0.8391 | 0.8435 | 0.8043 | 0.8279 | 0.0137 |

Table 2 shows the results of model training with a linear kernel and several variations of the C value, which functions as a hyperplane margin distance regulator and minimizes error during classification. All accuracy values averaged 82.8% with a small standard deviation of 0.0137. This indicates that the model is stable and not overly influenced by data variations in each fold. Furthermore, changes in the C parameter value do not alter the model's performance, meaning the dataset is relatively linear, so variations in C do not significantly affect the hyperplane. The accuracy values across folds range from 0.8043 to 0.8435, indicating that the data distribution in each fold is quite similar.

**Table 3:** Training Results with RBF Kernel Hyperparameters

| Parameter | Fold 1 | Fold 2 | Fold 3 | Fold 4 | Fold 5 | Mean | Standar Deviasi |
|---|---|---|---|---|---|---|---|
| C :0.1 Gamma :0.1 | 0.1562 | 0.1875 | 0.1641 | 0.0.1927 | 0.2005 | 0.1802 | 0.0171 |
| C :0.1 Gamma :1 | 0.3099 | 0.3620 | 0.2969 | 0.3438 | 0.3542 | 0.3333 | 0.0255 |
| C :0.1 Gamma :10 | 0.0417 | 0.0417 | 0.0417 | 0.0417 | 0.0417 | 0.0417 | 0.0000 |
| C :0.1 Gamma :100 | 0.0417 | 0.0417 | 0.0417 | 0.0417 | 0.0417 | 0.0417 | 0.0000 |
| C :1 Gamma :0.1 | 0.0859 | 0.0781 | 0.0729 | 0.0703 | 0.0859 | 0.0786 | 0.0065 |
| C : 1 Gamma : 1 | 0.2995 | 0.3594 | 0.2839 | 0.3203 | 0.3229 | 0.3172 | 0.0255 |
| C : 1 Gamma : 10 | 0.0417 | 0.0417 | 0.0417 | 0.0417 | 0.0417 | 0.0417 | 0.0000 |
| C : 1 Gamma : 100 | 0.0417 | 0.0417 | 0.0417 | 0.0417 | 0.0417 | 0.0417 | 0.0000 |
| C : 10 Gamma : 0.1 | 0.0990 | 0.0964 | 0.0833 | 0.0911 | 0.0938 | 0.0927 | 0.0054 |
| C : 10 Gamma : 1 | 0.2865 | 0.3568 | 0.2865 | 0.3099 | 0.2786 | 0.3036 | 0.0286 |
| C : 10 Gamma : 10 | 0.0417 | 0.0417 | 0.0417 | 0.0417 | 0.0417 | 0.0417 | 0.0000 |
| C : 10 Gamma: 100 | 0.0417 | 0.0417 | 0.0417 | 0.0417 | 0.0417 | 0.0417 | 0.0000 |
| C : 100 Gamma : 0.1 | 0.0990 | 0.0964 | 0.0833 | 0.0911 | 0.0938 | 0.0927 | 0.0054 |
| C : 100 Gamma : 1 | 0.2865 | 0.3568 | 0.2865 | 0.3099 | 0.2786 | 0.3036 | 0.0286 |
| C : 100 Gamma : 10 | 0.0417 | 0.0417 | 0.0417 | 0.0417 | 0.0417 | 0.0417 | 0.0000 |
| C : 100 Gamma : 100 | 0.0417 | 0.0417 | 0.0417 | 0.0417 | 0.0417 | 0.0417 | 0.0000 |

Table 3 shows the training results of the RBF kernel with several hyperparameters tested. The best performance in training with this kernel was obtained with the combination of C = 0.1 and gamma = 1, with an average accuracy of 0.3333, C = 1 and gamma = 1, with an average accuracy of 0.3172, and C = 10/100 and gamma = 1, with an average accuracy of 0.3036. Of the three highest hyperparameter combinations, gamma=1 is the most stable for this dataset in determining how far a data point influences its surroundings. The worst gamma performance was in training using gamma 10 and 100, which produced a value of 0.0417. This indicates that the model experienced severe overfitting, because gamma was too large, causing the model to focus only on very close points and not generalize. For the influence of a small C value of 0.1, the results were quite good when gamma = 1, even slightly better than a large C. With large C values of 10 and 100, the results were not much different from C = 1 when gamma = 1, but were worse when gamma was too small (0.1) or too large. The relatively small standard deviation of around 0.02–0.03 in the good accuracy results means that the results between folds were quite stable. In poor results with gamma 10 and 100, the standard deviation is even 0.00 because the model fails completely in all folds with identical results.

The conclusion from the above training is that when gamma is too small, i.e., with a value of 0.1, the model is too simple and does not capture the data patterns sufficiently (underfitting). When gamma is too large, i.e., with values of 10 and 100, the model is too complex and only memorizes the training data (severe overfitting). Therefore, gamma with a value of 1 provides optimal accuracy performance among all accuracy results on the RBF kernel. For the C value, a smaller value of 0.1 provides better results than larger C values, meaning that stronger regularization is more suitable for this dataset. The combination of C = 0.1 and Gamma = 1 is the best configuration with a Mean = 0.3333 and Standard Deviation = 0.0255.

**Table 4:** Training Results with Polynomial Kernel Hyperparameters

| Parameter | Fold 1 | Fold 2 | Fold 3 | Fold 4 | Fold 5 | Mean | Standar Deviasi |
|---|---|---|---|---|---|---|---|
| C : 0.1 Gamma : 0.1 | 0.7005 | 0.6484 | 0.6172 | 0.6953 | 0.6901 | 0.6703 | 0.0323 |
| C : 0.1 Gamma : 1 | 0.7005 | 0.6484 | 0.6172 | 0.6953 | 0.6901 | 0.6703 | 0.0323 |
| C : 0.1 Gamma : 10 | 0.7005 | 0.6484 | 0.6172 | 0.6953 | 0.6901 | 0.6703 | 0.0323 |
| C : 0.1 Gamma : 100 | 0.7005 | 0.6484 | 0.6172 | 0.6953 | 0.6901 | 0.6703 | 0.0323 |
| C : 1 Gamma : 0.1 | 0.7005 | 0.6484 | 0.6172 | 0.6953 | 0.6901 | 0.6703 | 0.0323 |
| C : 1 Gamma : 1 | 0.7005 | 0.6484 | 0.6172 | 0.6953 | 0.6901 | 0.6703 | 0.0323 |
| C : 1 Gamma : 10 | 0.7005 | 0.6484 | 0.6172 | 0.6953 | 0.6901 | 0.6703 | 0.0323 |
| C : 1 Gamma : 100 | 0.7005 | 0.6484 | 0.6172 | 0.6953 | 0.6901 | 0.6703 | 0.0323 |
| C : 10 Gamma : 0.1 | 0.7005 | 0.6484 | 0.6172 | 0.6953 | 0.6901 | 0.6703 | 0.0323 |
| C : 10 Gamma : 1.0 | 0.7005 | 0.6484 | 0.6172 | 0.6953 | 0.6901 | 0.6703 | 0.0323 |
| C : 10 Gamma : 10 | 0.7005 | 0.6484 | 0.6172 | 0.6953 | 0.6901 | 0.6703 | 0.0323 |
| C : 10 Gamma : 100 | 0.7005 | 0.6484 | 0.6172 | 0.6953 | 0.6901 | 0.6703 | 0.0323 |
| C : 100 Gamma : 0.1 | 0.7005 | 0.6484 | 0.6172 | 0.6953 | 0.6901 | 0.6703 | 0.0323 |
| C : 100 Gamma : 1 | 0.7005 | 0.6484 | 0.6172 | 0.6953 | 0.6901 | 0.6703 | 0.0323 |
| C : 100 Gamma : 10 | 0.7005 | 0.6484 | 0.6172 | 0.6953 | 0.6901 | 0.6703 | 0.0323 |
| C : 100 Gamma : 100 | 0.7005 | 0.6484 | 0.6172 | 0.6953 | 0.6901 | 0.6703 | 0.0323 |

Table 4 shows the results of training with a polynomial kernel, indicating similar values across all parameter combinations. The average accuracy is 0.6703. When training the dataset with a polynomial kernel, the C and gamma values do not affect the model results. This is because the previous training results with a linear kernel also showed that the dataset tends to be linear. The standard deviation values for all combinations are quite small, at 0.0323, meaning that the model performance is stable across different data sets.

**Table 5:** Training Results with Sigmoid Kernel Hyperparameters

| Parameter | Fold 1 | Fold 2 | Fold 3 | Fold 4 | Fold 5 | Mean | Standar Deviasi |
|---|---|---|---|---|---|---|---|
| C : 0.1 Gamma : 0.1 | 0.2318 | 0.2188 | 0.2292 | 0.2422 | 0.2578 | 0.2359 | 0.0132 |
| C : 0.1 Gamma : 1 | 0.2422 | 0.2109 | 0.2396 | 0.2370 | 0.2370 | 0.2333 | 0.0114 |
| C : 0.1 Gamma : 10 | 0.2396 | 0.2083 | 0.2500 | 0.2292 | 0.2552 | 0.2365 | 0.0167 |
| C : 0.1 Gamma : 100 | 0.2370 | 0.2109 | 0.2474 | 0.2370 | 0.2526 | 0.2370 | 0.0144 |
| C : 1 Gamma : 0.1 | 0.2292 | 0.2474 | 0.2318 | 0.2161 | 0.2474 | 0.2344 | 0.0119 |
| C :  Gamma : 1 | 0.2422 | 0.2500 | 0.2370 | 0.2266 | 0.2526 | 0.2417 | 0.0094 |
| C : 1 Gamma : 10 | 0.2344 | 0.2708 | 0.2630 | 0.2448 | 0.2422 | 0.2510 | 0.0136 |
| C : 1 Gamma : 100 | 0.2448 | 0.2500 | 0.2292 | 0.2760 | 0.2344 | 0.2469 | 0.0163 |
| C : 10 Gamma : 0.1 | 0.2448 | 0.2604 | 0.2188 | 0.2135 | 0.2734 | 0.2422 | 0.0232 |
| C : 10 Gamma : 1.0 | 0.2370 | 0.2448 | 0.2370 | 0.2422 | 0.2474 | 0.2417 | 0.0042 |
| C : 10 Gamma : 10 | 0.2578 | 0.2422 | 0.2188 | 0.2500 | 0.2578 | 0.2453 | 0.0145 |
| C : 10 Gamma : 100 | 0.2448 | 0.2604 | 0.2526 | 0.2630 | 0.2474 | 0.2536 | 0.0071 |
| C : 100 Gamma : 0.1 | 0.2422 | 0.2630 | 0.2135 | 0.2188 | 0.2500 | 0.2375 | 0.0187 |
| C : 100 Gamma : 1 | 0.2500 | 0.2370 | 0.2266 | 0.2214 | 0.2734 | 0.2417 | 0.0187 |
| C : 100 Gamma : 10 | 0.2578 | 0.2448 | 0.2214 | 0.2552 | 0.2500 | 0.2458 | 0.0130 |
| C : 100 Gamma : 100 | 0.2448 | 0.2578 | 0.2552 | 0.2604 | 0.2474 | 0.2531 | 0.0060 |

Table 5 shows the training results with the sigmoid kernel. The average accuracy value ranges from 0.23 to 0.25, with a relatively small standard deviation of 0.004–0.023, meaning that the model is quite consistent even though its performance is not high. The best performance was achieved with the combination of C = 10 and gamma = 100, which produced an average accuracy value of 0.2536, and the combination of C = 100 and gamma = 100, which produced an average accuracy value of 0.2531. A large C value provides looser regularization, meaning that the model adjusts itself to the training data as much as possible. For gamma usage in this kernel, the best accuracy value is achieved with a very large gamma of 100, resulting in 0.253. However, even though it has been optimized, the sigmoid kernel is still not suitable for this dataset, compared to the results of the linear kernel.

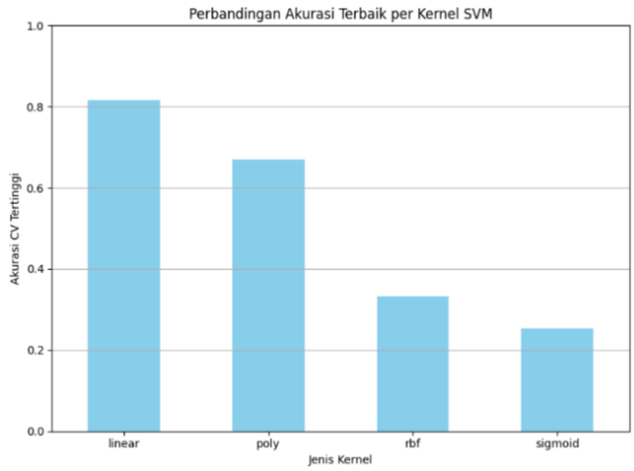


**Fig. 3:** Model Training highest Accuracy

Figure 3 shows that the linear kernel has the highest accuracy, followed by the polynomial kernel in second place, the RBF kernel in third place, and the sigmoid kernel in last place. The results of training the SVM algorithm using a linear kernel achieved an accuracy of 81.72% on the validation data, the sigmoid kernel achieved 25.36% on the validation data, the RBF kernel achieved an accuracy of 33.33% on the validation data, and the polynomial kernel achieved an accuracy of 69.03% on the validation data. The best training result was achieved by training the SVM using a linear kernel.

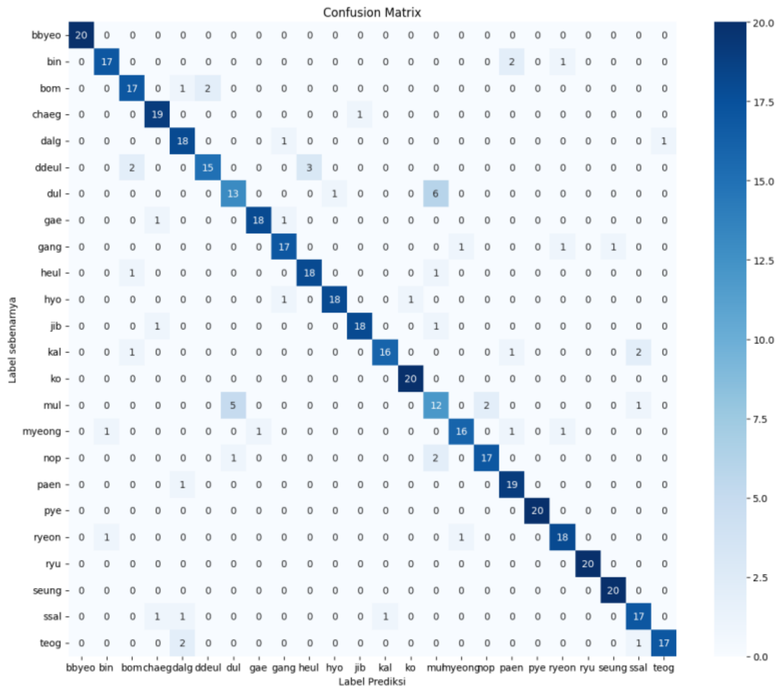

**Fig. 4:** Confusion Matrix

Figure 4 shows the confusion matrix results for the test data using the kernel with the highest accuracy during previous training. The kernel used was a linear kernel with a C value of 0.1. The accuracy results for the test data showed a total accuracy of 87.50%. This accuracy result is considered good for recognizing handwritten Hangul single-syllable character data. Almost all labels have a high number of correct predictions, and there are even some classes with perfect accuracy. However, there are several classes that experience misclassification of labels, for example, the label dul with 13 correct predictions, which was mistakenly classified as the class ddeul with 2 predictions and the class gae with 6 predictions. Similar patterns between certain classes make it slightly difficult for the linear model to separate them. As seen from the many classes with near-perfect accuracy, the model performs well. The SVM model with a linear kernel works optimally for this dataset, as the majority of classes are recognized very well.

## 4. Conclusion

Based on the results of the research, validation accuracy from model 81.72%. Among the tested kernels, the linear kernel achieved the highest accuracy on the validation data during the training process using the Support Vector Machine (SVM) algorithm. The best performance was obtained with the parameter C = 0.1, which has a direct effect on the classification outcome. A smaller C value helps improve the model's ability to generalize the data. Furthermore, on the testing data, the SVM model with a linear kernel C=0.1 achieved an accuracy of 87.50%, indicating that the model can effectively recognize the data and obtain optimal accuracy on unseen data.

## References


[1] R. Budiawan, I. Arief, R. Munir, and D. Mahayana, "Pergeseran Paradigma pada Penelitian Pengenalan Tulisan Tangan," *Jurnal Filsafat Indonesia Undiksha*, Jun. 2023.

[2] L. Windrim, R. Ramakrishnan, A. Melkumyan, and R. J. Murphy, "Hyperspectral CNN Classification with Limited Training Samples."

[3] L. M. Abu Zohair, "Prediction of Student's performance by modelling small dataset size," *International Journal of Educational Technology in Higher Education*, vol. 16, no. 1, Dec. 2019, doi: 10.1186/s41239-019-0160-3.

[4] C.-S. Yang and C.-C. Hsieh, "High Accuracy Text Detection using ResNet as Feature Extractor," in *2019 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE)*, 2019, pp. 92–95. doi: 10.1109/ECICE47484.2019.8942666.

[5] S. Rahayu, S. Sandiwarno, E. D. Putra, M. Utami, and H. Setiawan, "Model Sequential Resnet50 Untuk Pengenalan Tulisan Tangan Aksara Arab Article Info ABSTRAK," *JSAI: Journal Scientific and Applied Informatics*, vol. 06, no. 02, 2023, doi: 10.36085.

[6] F. Muhammad, A. Murni Arimurthy, and D. Chahyati, "Transfer Learning dengan Metode Fine Tuning pada Model Network VGG16 dan ResNet50," *Indonesian Journal of Computer Science Attribution (2023)*, vol. 12, no. 1, p. 361, 2023.

[7] F. Islam, M. H. Rahman, Nurjahan, M. S. Hossain, and S. Ahmed, "A Novel Method for Diagnosing Alzheimer's Disease from MRI Scans using the ResNet50 Feature Extractor and the SVM Classifier," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 6, pp. 1233–1242, 2023, doi: 10.14569/IJACSA.2023.01406131.

[8] A. Y. Mahmoud, "Preliminary Introduction and Implementation of Novel Machine Learning Algorithm Utilising Pareto Principle: Classification of Small Biomedical Health-Related Datasets," in *Advances in Computational Intelligence Systems*, M. and M. L. S. Panoutsos George and Mahfouf, Ed., Cham: Springer Nature Switzerland, 2024, pp. 129–141.

[9] H. Tabrizchi, M. M. Javidi, and V. Amirzadeh, "Estimates of residential building energy consumption using a multi-verse optimizer-based support vector machine with k-fold cross-validation," *Evolving Systems*, vol. 12, no. 3, pp. 755–767, Sep. 2021, doi: 10.1007/s12530-019-09283-8.