

## **Penerapan Model Deep-CNN Untuk Meningkatkan Akurasi Klasifikasi Bahasa Isyarat Alfabet Menggunakan Algoritma Convolutional Neural Network**

**Fadhil Al-Hafizh\*, Alamsyah**

Program Studi Teknik Informatika, FMIPA, Universitas Negeri Semarang, Indonesia  
Gedung D5 Lt.2, Kampus Sekaran Gunungpati, Semarang 50229  
E-mail: fadhilhafizh18@students.unnes.ac.id

Diterima 6 Desember 2023      Disetujui 01 Maret 2024

### **Abstrak**

Perkembangan teknologi dalam era kecerdasan buatan telah membawa pengembangan sistem komputer yang mampu mengidentifikasi bahasa isyarat. Bahasa isyarat adalah alat komunikasi utama bagi penyandang tunarungu dan tunawicara yang digunakan oleh jutaan orang di seluruh dunia. Penelitian ini bertujuan untuk meningkatkan akurasi dalam pengenalan bahasa isyarat alfabet dengan menggunakan model Deep-CNN. Metode dalam penelitian ini dimulai dari pemilihan dataset Bahasa isyarat berdasarkan penelitian sebelumnya. Dataset yang dipakai pada penelitian ini berasal dari Kaggle mengenai American Sign Language yang berisi setiap kelas pelatihan dan uji mewakili label (0-25) dataset ini berisi data pelatihan sebanyak 27.455 data dan data uji sebanyak 7.172 data. Dalam penelitian ini menggunakan bahasa pemrograman Python dalam melakukan split data, scaling, data augmentation, training, dan evaluating. Model arsitektur yang dibangun dalam penelitian ini adalah arsitektur Deep CNN yang diimplementasikan untuk melakukan proses peningkatan akurasi klasifikasi pengenalan bahasa isyarat. Hasil pengujian menunjukkan adanya peningkatan akurasi klasifikasi bahasa isyarat alfabet dibandingkan penelitian sebelumnya. Kenaikan akurasi pada nilai model Deep CNN yang dibangun berhasil mencapai tingkat akurasi 99,72%. Model yang telah dibangun ini merupakan model terbaik di antara model-model penelitian sebelumnya.

Kata kunci: bahasa isyarat, Convolutional Neural Network, Deep-CNN, klasifikasi

### **Abstract**

*Technological developments in the era of artificial intelligence have led to the development of computer systems capable of identifying sign language. Sign language is the primary means of communication for deaf and hard of hearing people used by millions of people around the world. This research aims to improve accuracy in alphabetic sign language recognition by using the Deep-CNN model. The method in this research starts from the selection of sign language datasets based on previous research. The dataset used in this study comes from Kaggle regarding American Sign Language which contains each training and test class representing a label (0-25) This dataset contains 27,455 training data and 7,172 test data. This research uses the Python programming language in performing data splitting, scaling, data augmentation, training, and evaluating. The architectural model built in this research is the Deep CNN architecture which is implemented to carry out the process of improving the accuracy of sign language recognition classification. The test results show an increase in the accuracy of alphabetic sign language classification compared to previous research. The increase in accuracy in the value of the Deep CNN model built managed to reach an accuracy rate of 99.72%. The model that has been built is the best model among previous research models.*

Keywords: sign language, Convolutional Neural Network, Deep-CNN, classification

### **How to cite:**

Al-Hafizh F., Alamsyah A. (2024). Penerapan model deep-CNN untuk meningkatkan akurasi klasifikasi bahasa isyarat alfabet menggunakan algoritma convolutional neural network. *Indonesian Journal of Mathematics and Natural Sciences*, 47(1), 1-11.

### **PENDAHULUAN**

Perkembangan teknologi membawa dunia ke era kecerdasan buatan, di mana sistem komputer yang ditengahi oleh kecerdasan buatan mampu memahami dunia visual dengan kecerdasan yang

melampaui kemampuan manusia dalam menyelesaikan tugas tertentu (Feng *et al.*, 2019). Salah satu fungsi utamanya adalah kemampuannya untuk mengidentifikasi data visual yang bermanfaat bagi kelompok tertentu. Salah satu contoh identifikasi data visual yang penting adalah kemampuan untuk mengenali bahasa isyarat, yang dapat sangat bermanfaat bagi individu tunarungu (Lim *et al.*, 2019).

Bahasa isyarat merupakan bahasa komunikasi sehari-hari antara penyandang tunarungu dan bisu, yang merupakan cara komunikasi paling nyaman dan alami antara penyandang tunarungu dan bisu, dan juga merupakan alat utama sekolah pendidikan luar biasa untuk mengajar dan menyampaikan gagasan (He, 2019). Diketahui bahwa sekitar 2.500.000 orang dari seluruh dunia menggunakan bahasa isyarat untuk berkomunikasi (Sharma & Kumar, 2021). Bahasa isyarat adalah metode komunikasi utama bagi orang-orang yang menyandang tunarungu dan tunawicara. Menggunakan isyarat tangan, posisi dan gerakan yang berbeda, kumpulan simbol bergambar untuk setiap bahasa terdapat isyarat tangan yang memerlukan salah satu atau kedua tangan (Rafi *et al.*, 2019).

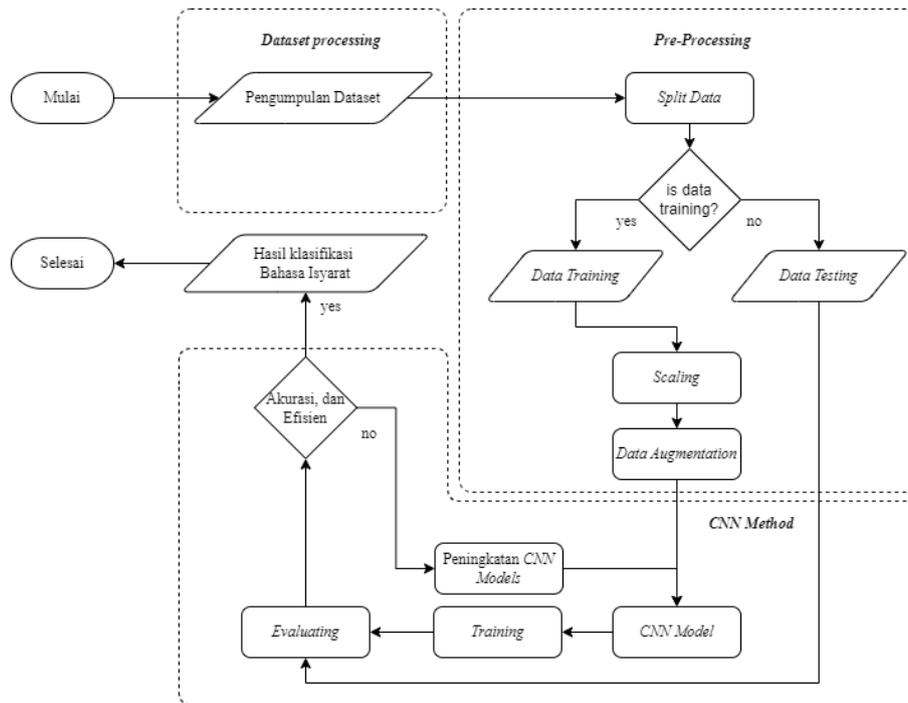
Algoritma yang digunakan dalam untuk mengidentifikasi citra bahasa isyarat alfabet adalah Convolutional Neural Network (CNN) adalah model *deep learning* yang secara khusus dirancang untuk mengolah data berstruktur seperti grid. Data semacam ini dapat berupa gambar atau baris data multi-kolom. Deep Learning merujuk pada model-model yang terdiri dari beberapa lapisan. Secara umum, sebuah model dianggap *deep* jika memiliki setidaknya satu lapisan input, satu lapisan tersembunyi, dan satu lapisan output (Nevavuori *et al.*, 2019). Metode CNN dipilih karena mampu secara efektif menggambarkan pola-pola spasial, sehingga memungkinkan ekstraksi berbagai sifat vegetasi dari citra pemantauan jarak jauh. Metode CNN dirancang untuk mempelajari fitur-fitur spasial, seperti tepi, sudut, atau bentuk lebih abstrak, yang paling baik menggambarkan kelas atau kuantitas target (Kattenborn *et al.*, 2021).

Berdasarkan penelitian yang dilakukan Cayamcela dan Lim (2019), Xiao *et al.* (2022), Bilgin dan Mutludogan (2019), Jain *et al.* (2021), dan Lee *et al.* (2021) tentang identifikasi bahasa isyarat alfabet menggunakan algoritma dan dataset yang berbeda-beda sehingga menghasilkan akurasi yang beragam. Penelitian Cayamcela dan Lim (2019) menggunakan algoritma GoogleNet dengan akurasi sebesar 95,52% dan AlexNet sebesar 99,39%. Penelitian Xiao *et al.* (2022) menggunakan algoritma SLR-CapsNet dengan akurasi sebesar 98,17%. Penelitian oleh Bilgin dan Mutludogan (2019) menggunakan algoritma LeNet dengan akurasi sebesar 82% dan CapsNet sebesar 88%. Penelitian Jain *et al.* (2021) menggunakan algoritma CNN dengan akurasi sebesar 97,344% dan 98,581%, menggunakan SVM dengan akurasi sebesar 81,49%. Penelitian Lee *et al.* (2021) menggunakan algoritma kombinasi Long-Short Term Memory Recurrent Neural Network (LSTM-RNN) dengan K-Nearest-Neighbour (KNN) dengan akurasi sebesar 99,44% dan 91,82%. Namun penelitian-penelitian tersebut belum mendapatkan hasil akurasi yang optimal dalam implementasi proses klasifikasi pengenalan bahasa isyarat.

Dalam penelitian ini, menggunakan arsitektur Deep-CNN yang telah dimodifikasi kompleksitas dan parameternya serta data augmentation yang diperbanyak untuk meningkatkan pengenalan klasifikasi bahasa isyarat. Penelitian ini berfokus untuk meningkatkan akurasi dengan menggunakan model Deep-CNN pada model algoritma dalam klasifikasi bahasa isyarat alfabet. Tujuan utama penelitian ini adalah mendapatkan hasil akurasi yang lebih akurat dan efektif dalam bahasa isyarat alfabet.

## **METODE**

Penelitian ini menerapkan desain penelitian eksperimental yang menggunakan algoritma CNN dalam mengklasifikasikan bahasa isyarat alfabet. Tahapan awal dimulai dengan *dataset processing*, kemudian dataset diolah lebih lanjut di tahap *pre-processing*, tahap ini meliputi *split data* yang dibagi menjadi dua yaitu *data train* dan *data test*, *resize*, dan *data augmentation* yang bertujuan untuk memastikan data dalam kondisi bersih dan siap untuk tahap selanjutnya. kemudian dilakukan pemodelan CNN untuk melatih model. Selanjutnya, model dievaluasi menggunakan *confusion matrix* yang menghasilkan nilai akurasi, presisi dan *F1-Score*. Bagan alir desain penelitian dapat dilihat pada Gambar 1.



Gambar 1. Metode yang diusulkan

### Dataset

Dataset yang digunakan dalam identifikasi bahasa isyarat bersumber dari *Modified National Institute of Standards and Technology* (MNIST). Dataset ini berisi citra tangan yang menggambarkan huruf abjad dalam bahasa isyarat ASL (*American Sign Language*). Dataset MNIST Sign Language memiliki total 24 kelas yang mewakili berbagai huruf dalam abjad ASL seperti pada Gambar 2. Huruf “J” dan “Z” tidak termasuk dalam dataset ini, karena keduanya merupakan gerakan gestur tangan.

Dataset yang digunakan dalam penelitian ini terdiri dari 34.627 data citra tangan. Data ini dibagi menjadi dua bagian, yaitu data training dengan jumlah sebanyak 27.455 data citra dan data test sejumlah 7,172 data citra. Data training digunakan untuk melatih model atau algoritma yang digunakan dalam penelitian, sementara data test digunakan untuk menguji kinerja model tersebut.



Gambar 2. Huruf bahasa isyarat

## Pre-processing

*Pre-processing* data adalah tahap pertama dalam melakukan pembelajaran, dalam bagian ini data diproses atau dikodekan untuk memungkinkan mesin memindai atau menganalisisnya dengan cepat (Maharana *et al.*, 2022; Alamsyah & Fadila, 2021; Alamsyah *et al.*, 2021). Tujuan dari *Pre-processing* adalah untuk mengubah data ke dalam format yang sesuai untuk analisis di masa mendatang dan untuk menghilangkan kesalahan atau ketidakkonsistenan yang dapat mengubah hasil (Chaurasia *et al.*, 2023). *Pre-processing* yang dilakukan pada penelitian ini meliputi *split data*, *scaling*, *data augmentation*.

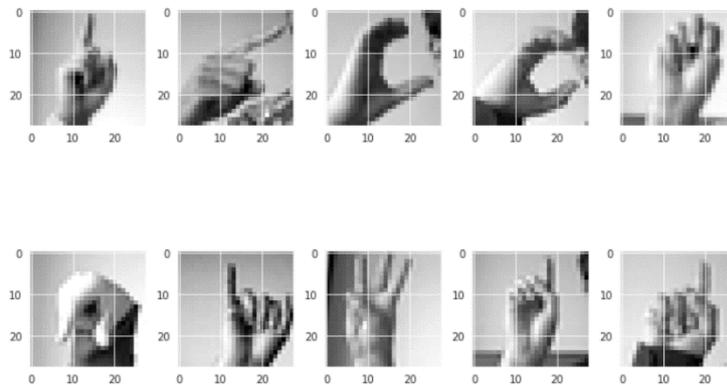
## Split Data

Dalam tahap ini, 80% dari *data train* digunakan untuk *training*. Sementara itu, sisa 20% dari keseluruhan dataset digunakan sebagai *data test*. Langkah ini juga membantu mengurangi risiko *overfitting* pada model yang sedang dikembangkan.

## Scaling

*Scaling* adalah proses mengubah ukuran gambar digital. *Scaling-down* akan membuatnya menjadi lebih kecil, sedangkan *Scaling-up* akan membuatnya menjadi lebih besar (Sopari & Chandra, 2020). Dalam penelitian ini, *scaling* digunakan untuk menyesuaikan ukuran gambar agar sesuai dengan spesifikasi model yang digunakan.

Berdasarkan dataset yang sudah diperoleh memiliki ukuran yang bervariasi. Data gambar yang awalnya berbentuk *1-D array* dengan panjang 784 ( $28 \times 28$ ), diubah menjadi *array 3D* dengan ukuran  $28 \times 28 \times 1$  (tinggi  $\times$  lebar  $\times$  saluran atau *grayscale*), kemudian parameter -1 digunakan dalam operasi *scaling* untuk menandakan bahwa jumlah baris atau data tidak berubah. Sehingga, ukuran data berubah menjadi  $-1 \times 28 \times 28 \times 1$ , yang menunjukkan bahwa jumlah data atau parameter tidak berubah, tinggi gambar adalah 28, lebar gambar adalah 28, dan terdapat satu saluran (*grayscale*) pada setiap gambar. Ilustrasi hasil *scaling* bisa dilihat pada Gambar 3.



Gambar 3. Ilustrasi hasil *scaling*

## Data Augmentation

Data augmentasi merupakan metode yang digunakan untuk mengatasi masalah *overfitting* dengan cara memperluas dataset pelatihan. Metode augmentasi untuk klasifikasi data menggunakan pembalikan (*flipping*), rotasi (*rotating*), pergeseran (*shifting*), penambahan noise, pengaburan (*blurring*), penajaman (*sharpening*) (Yun *et al.*, 2020). Augmentasi data dilakukan pada dataset pelatihan dengan menerapkan transformasi acak, seperti pergeseran horizontal, translasi pada sumbu *x* dan *y* dengan rentang tertentu. Hal ini bertujuan untuk membantu model mengingat detail yang penting dari data atau gambar pelatihan (Cayamcela & Lim, 2019).

Metode augmentasi data yang digunakan dalam penelitian ini meliputi rotasi gambar acak antara 0 dan 180 derajat, pembesaran gambar secara acak, pergeseran gambar secara acak (*horizontal* atau *vertikal*), dan pembalikan foto secara acak (*horizontal* atau *vertikal*).

## Model Convolutional Neural Network

Setelah tahap pre-processing, perancangan model CNN dibentuk untuk melakukan klasifikasi bahasa isyarat alfabet. Perancangan model ini sangat berpengaruh terhadap hasil akurasi model yang dilatih. Penelitian ini menggunakan *library Keras* untuk membangun struktur model yang diusulkan. Pendekatan yang digunakan untuk membangun model menggunakan *Keras* adalah melalui penggunaan metode *sequential*. Metode *sequential* adalah cara yang intuitif untuk membangun sebuah model di *Keras*. Dengan menggunakan model *sequential* dapat membangun model lapis demi lapis sebagai tumpukan lapisan (Manaswi, 2018)

Dalam penelitian ini, *Deep-CNN* yang telah dimodifikasi digunakan agar dapat meningkatkan akurasi dalam klasifikasi pengenalan bahasa isyarat. *Deep-CNN* dalam penelitian ini terdiri dari beberapa jenis lapisan, termasuk *Convolutional layer*, *BatchNormalization*, *Dropout*, dan *Dense*, serta menerapkan teknik *pooling* untuk mengurangi dimensi input. Arsitektur ini terdiri dari beberapa lapisan kompleks yang efektif dalam mengekstraksi fitur-fitur dari gambar input. Secara umum, istilah "deep CNN" mengacu pada arsitektur CNN yang memiliki banyak lapisan konvolusi.

*BatchNormalization* adalah proses dalam pelatihan arsitektur jaringan saraf yang memastikan input ke setiap lapisan dalam setiap batch kecil memiliki distribusi data yang konsisten, sesuai dengan yang diharapkan dari lapisan berikutnya (Alam *et al.*, 2021). Penelitian ini menggunakan *Dropout layer* untuk menghilangkan beberapa neuron yang telah dilatih, sehingga membantu mengurangi *overfitting* (Agrawal *et al.*, 2020). *Dense layer* digunakan dengan fungsi aktivasi ReLU dan terdiri dari 512 unit, yang diikuti oleh *Dropout layer* dengan tingkat *dropout* sebesar 0.3 yang bertujuan untuk mengendalikan *overfitting*. Dalam model ini menggunakan *dense layer* dengan 24 unit (*fully connected layer*) sebagai lapisan keluaran dan menerapkan fungsi *softmax* untuk memprediksi gerakan bahasa isyarat alfabet. Selain itu, kami menggunakan *flatten layer* untuk meratakan atau mengubah data menjadi vektor sebelum dihubungkan ke *fully connected layer*. sehingga didapatkan ringkasan arsitektur yang disajikan pada Gambar 4.

```
Model: "sequential_1"
```

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 28, 28, 75)	750
batch_normalization_1 (Batch Normalization)	(None, 28, 28, 75)	300
max_pooling2d_1 (MaxPooling2D)	(None, 14, 14, 75)	0
conv2d_2 (Conv2D)	(None, 14, 14, 50)	33800
dropout_1 (Dropout)	(None, 14, 14, 50)	0
batch_normalization_2 (Batch Normalization)	(None, 14, 14, 50)	200
max_pooling2d_2 (MaxPooling2D)	(None, 7, 7, 50)	0
conv2d_3 (Conv2D)	(None, 7, 7, 25)	11275
batch_normalization_3 (Batch Normalization)	(None, 7, 7, 25)	100
max_pooling2d_3 (MaxPooling2D)	(None, 4, 4, 25)	0
flatten_1 (Flatten)	(None, 400)	0
dense_1 (Dense)	(None, 512)	205312
dropout_2 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 24)	12312

```
Total params: 264,049  
Trainable params: 263,749  
Non-trainable params: 300
```

Gambar 4. Ringkasan arsitektur

## Pelatihan

Tahap ini merupakan tahap di mana akan dilakukan training pada model menggunakan data training yang telah dipersiapkan sebelumnya, di tahap ini model akan belajar untuk melakukan klasifikasi huruf alfabet pada bahasa isyarat. Proses training ini akan menggunakan arsitektur dari CNN, *Deep-CNN*. Setelah proses training selesai, model akan disimpan untuk dilakukan evaluasi pada tahap selanjutnya.

## Evaluasi

*Confusion matrix* adalah metode yang digunakan untuk mengevaluasi efektivitas dan menemukan kesalahan dalam model yang digunakan untuk klasifikasi (Rathore *et al.*, 2023). Baris dalam confusion matrix menunjukkan hasil kelas yang sebenarnya, sedangkan kolom dalam confusion matrix menunjukkan hasil kelas yang diprediksi (Xu *et al.*, 2020). Menurut Vujović (2021), beberapa metrics yang digunakan untuk menilai performa model yang telah dikembangkan, antara lain:

*Accuracy* ditentukan dengan membagi jumlah total set data (P + N) dengan jumlah dua prediksi yang akurat (TP + TN). Akurasi berkisar antara 0,00 hingga 1,0. *Accuracy* dapat dihitung dengan persamaan (1).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

*Recall* dihitung dengan membagi jumlah prediksi positif yang akurat (TP) dengan jumlah total kasus positif (P). *Recall* dapat dihitung dengan persamaan (2).

$$Recall = \frac{True\ Positive\ (TP)}{True\ Positive\ (TP) + False\ Negative\ (FN)} \quad (2)$$

*Precision* dihitung dengan membagi jumlah prediksi positif yang benar (TP) dengan jumlah total prediksi positif (TP + FP). *Precision* dapat dihitung dengan persamaan (3).

$$Precision = \frac{True\ Positive\ (TP)}{True\ Positive\ (TP) + False\ Positive\ (FP)} \quad (3)$$

*F1-score* adalah suatu indikator dari ketepatan atau akurasi sebuah tes. *F1-score* dihitung berdasarkan presisi dan recall. *F1-score* dapat dihitung dengan persamaan (4).

$$F1\ score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

## HASIL DAN PEMBAHASAN

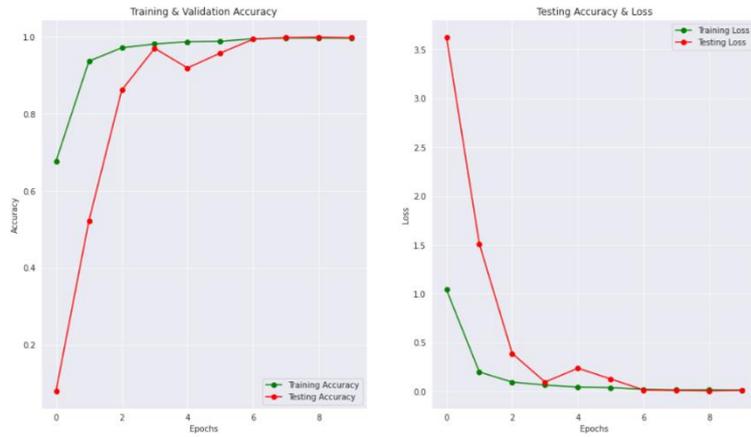
### Hasil Pelatihan

Hasil training dari model Deep CNN yang telah dibangun dengan menambahkan *batch\_size* =128, dan *epochs* =10. Hasil dari pelatihan menunjukkan bahwa semakin tinggi *epoch* maka akan semakin tinggi nilai *accuracy* dan *val\_accuracy* (Tabel 1).

Tabel 1. Tabel Training

Epoch	Lr_reduced	Accuracy	Loss	Val_accuracy	Val_loss
1	-	0.6770	1.0436	0.0806	3.6262
6	0.0005	0.9877	0.0374	0.9573	0.1256
10	-	0.9959	0.0120	0.9972	0.0085

Grafik pelatihan mencakup akurasi pelatihan dan validasi serta loss selama jumlah epoch selama pelatihan model. Dengan menggunakan warna dan penanda yang berbeda untuk setiap baris, dan menambahkan judul, legenda, dan label sumbu untuk kedua subplot. Fungsi ini berguna untuk mengevaluasi kinerja model selama pelatihan dan mengidentifikasi masalah *overfitting* atau *underfitting* yang mungkin terjadi selama pelatihan. Grafik pelatihan pada penelitian ini dapat dilihat pada Gambar 5.



Gambar 5. Grafik pelatihan

### Hasil Klasifikasi

Hasil klasifikasi dari model *Deep-CNN* yang telah dibangun dengan menggunakan dataset *sign language MNIST*. Perhitungan *Precision*, *Recall*, dan *F1-score* ini dilakukan di dalam program yang telah dibuat. Hasil klasifikasi ditulis pada Tabel 2.

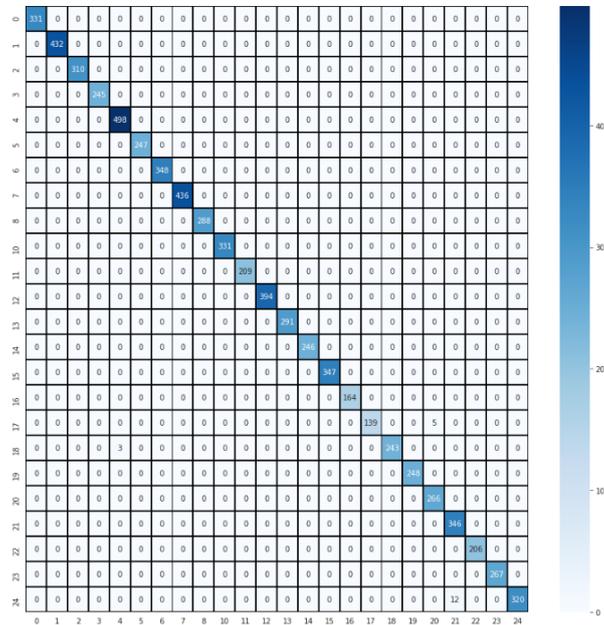
Tabel 2. Hasil klasifikasi

	Precision	Recall	F1-score	Support
Class 0 (A)	1,00	1,00	1,00	331
Class 1 (B)	1,00	1,00	1,00	432
Class 2 (C)	1,00	1,00	1,00	310
Class 3 (D)	1,00	1,00	1,00	245
Class 4 (E)	0,99	1,00	0,99	498
Class 5 (F)	1,00	1,00	1,00	247
Class 6 (G)	1,00	0,98	0,99	348
Class 7 (H)	1,00	1,00	1,00	436
Class 8 (I)	1,00	1,00	1,00	288
Class 10 (K)	1,00	1,00	1,00	331
Class 11 (L)	1,00	1,00	1,00	209
Class 12 (M)	1,00	1,00	1,00	394
Class 13 (N)	1,00	1,00	1,00	291
Class 14 (O)	1,00	1,00	1,00	246
Class 15 (P)	1,00	1,00	1,00	347
Class 16 (Q)	1,00	1,00	1,00	164
Class 17(R)	1,00	0,97	0,98	144
Class 18 (S)	1,00	0,99	0,99	246
Class 19 (T)	1,00	1,00	1,00	248
Class 20 (U)	0,98	1,00	0,99	266
Class 21 (V)	0,97	1,00	0,98	346
Class 22 (W)	1,00	1,00	1,00	206
Class 23 (X)	1,00	1,00	1,00	267
Class 24 (Y)	1,00	0,96	0,98	332
Accuracy			1,00	7172
Macro avg	1,00	1,00	1,00	7172
Weighted avg	1,00	1,00	1,00	7172

Berdasarkan Tabel 2. Support merupakan isi dataset dari setiap masing-masing kelas, sehingga jika di gabung menjadi 7.172 dataset yang merupakan total dari testing data. Nilai akurasi menunjukkan angka 1,00 yang dimana merupakan pembulatan dari hasil akhir akurasi yang di dapat yaitu 99,72%.

## Evaluasi

Dari hasil pelatihan dengan menggunakan model yang telah dibangun, didapat hasil dari *confusion matrix* seperti pada Gambar 6.



Gambar 6. *Confusion matrix*

Berdasarkan Gambar 6, dapat dilihat bahwa semua data uji sudah dapat membaca nilai aktual dan nilai prediksi, hanya terdapat beberapa yang salah dalam membaca nilai aktual dan prediksi seperti dalam membaca class 24, terdapat beberapa data citra yang dibaca dan di prediksi sebagai class 21.

## Hasil Akurasi

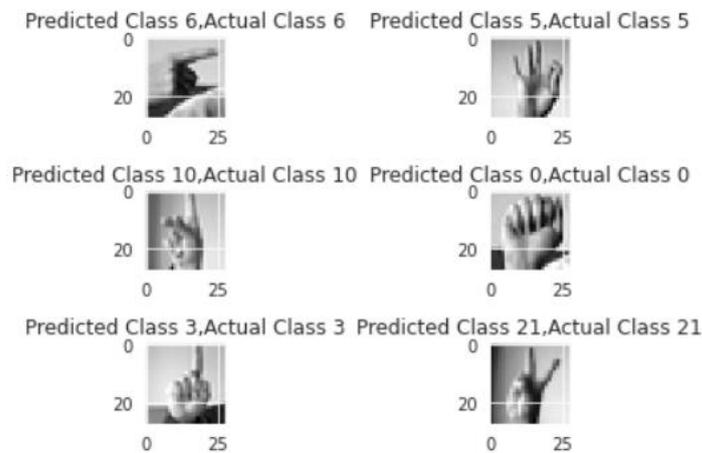
Hasil akurasi dari model *Deep-CNN* yang telah dibangun dengan menggunakan dataset *sign language MNIST*. Tabel 3 menampilkan hasil dari akurasi penelitian yang telah di latih.

	Model	Data	Akurasi (%)
Metode yang diusulkan	Deep CNN	Data Train	99,59
		Data Test	99,72

Berdasarkan Tabel 3, hasil uji akurasi yaitu 99,72%. Angka ini cukup baik dalam meningkatkan akurasi. Hal ini menunjukkan bahwa model *Deep-CNN* yang telah dibangun dengan menggunakan dataset *sign language MNIST* yang dibangun cukup baik untuk meningkatkan akurasi.

## Hasil Tes Prediksi

Setelah dilakukan pelatihan, hasil penelitian di uji menggunakan beberapa data citra yang di tampilkan secara acak yang menampilkan enam data citra beserta dengan kelasnya masing-masing dan juga hasil prediksi yang sebenarnya. Hasil prediksi di dapatkan dari data citra yang telah di latih sebelumnya, seperti pada Gambar 7.



Gambar 7. Hasil tes prediksi

### Analisis Perbandingan Kinerja

Untuk mengetahui performa dari algoritma yang dihasilkan, dilakukan perbandingan dengan hasil penelitian sebelumnya dengan memilih salah satu pengujian yang dilakukan pada dataset yang sama. Perbandingan performa akurasi dilakukan dengan menggunakan model uji, yaitu pengujian terhadap akurasi. Perbandingan nilai akurasi disajikan pada Tabel 4.

Tabel 4. Perbandingan akurasi

Referensi	Model	Akurasi (%)
Cayamcela & Lim (2019)	googLeNet	95.52
Cayamcela & Lim (2019)	AlexNet	99.39
Bilgin & Mutludogan (2019)	LeNet	82,19
Bilgin & Mutludogan (2019)	CapsNet	88,93
Bilgin & Mutludogan (2019)	CapsNet Augmented	95,08
Xiao <i>et al.</i> (2022)	SLR-CapsNet	98,17
Jain <i>et al.</i> (2021)	SVM	81.49
Lee <i>et al.</i> (2021)	LSTM-RNN	99,44%
Lee <i>et al.</i> (2021)	KNN	91,82%.
<b>Metode yang diusulkan</b>	Deep CNN	99,72

Seperti ditunjukkan pada Tabel 4, metode yang diusulkan mendapatkan hasil yang lebih baik dibandingkan dengan model pada penelitian terdahulu. Metode yang diusulkan dengan model *Deep-CNN* berhasil mendapatkan tingkat akurasi sebesar 99,72%.

### SIMPULAN

Rancangan model desain CNN dengan Arsitektur Deep-CNN yang telah dimodifikasi menghasilkan peningkatan akurasi setelah dilakukan pengujian. dengan nilai Deep-CNN 99,72%. Hasil ini menunjukkan bahwa Arsitektur Deep-CNN yang telah dimodifikasi dan data augmentation yang diperbanyak dapat meningkatkan akurasi klasifikasi pengenalan bahasa isyarat. Untuk penelitian selanjutnya diharapkan dapat meningkatkan akurasi yang diperoleh dari penelitian sebelumnya dengan berbagai modifikasi metode lainnya seperti Random Forrest, LSTM, SVM, KNN, dan metode lainnya.

### DAFTAR PUSTAKA

Agrawal, M., Ainapure, R., Agrawal, S., Bhosale, S., & Desai, S. (2020). Models for hand gesture recognition using deep learning. *2020 IEEE 5th International Conference on Computing Communication and Automation, ICCCA 2020*, 589-594. <https://doi.org/10.1109/ICCCA49541.2020.9250846>

- Alam, M. S., Tanvir, M., Saha, D. K., & Das, S. K. (2021). Two-dimensional convolutional neural network approach for real-time bangla sign language characters recognition and translation. *SN Computer Science*, 2(387), 1-13. <https://doi.org/10.1007/s42979-021-00783-6>
- Alamsyah, A., & Fadila, T. (2020). Increased accuracy of prediction hepatitis disease using the application of principal component analysis on a support vector machine. *Journal of Physics: Conference Series*. 1968 012016. <https://doi.org/10.1088/1742-6596/1968/1/012016>
- Alamsyah, A., Prasetyo, B., Hakim, M., & Pradana, F. (2021). Prediction of COVID-19 using recurrent neural network model. *Scientific Journal of Informatics*, 8(1), 98-103. <https://doi.org/10.15294/sji.v8i1.30070>.
- Bilgin, M., & Mutludogan, K. (2019). *American Sign Language Character Recognition with Capsule Networks. 2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, Ankara, Turkey, 2019, 1-6. <https://doi.org/10.1109/ISMSIT.2019.8932829>
- Cayamcela, M. E. M., & Lim, W. (2019). *Fine-tuning a pre-trained Convolutional Neural Network Model to translate American Sign Language in Real-time. 2019 International Conference on Computing, Networking and Communications (ICNC)*, Honolulu, HI, USA, 2019, 100-104, doi: <https://doi.org/10.1109/ICNC.2019.8685536>
- Chaurasia, A., Gautam, A., Rajkumar, R., & Chander, A. S. (2023). Road traffic optimization using image processing and clustering algorithms. *Advances in Engineering Software*, 181. <https://doi.org/10.1016/j.advengsoft.2023.103460>
- Feng, X., Jiang, Y., Yang, X., Du, M., & Li, X. (2019). Computer vision algorithms and hardware implementations: A survey. *Integration*, 69, 309-320. <https://doi.org/10.1016/j.vlsi.2019.07.005>
- He, S. (2019). Research of a sign language translation system based on deep learning. *Proceedings - 2019 International Conference on Artificial Intelligence and Advanced Manufacturing, AIAM 2019*, 392-396. <https://doi.org/10.1109/AIAM48774.2019.00083>
- Jain, V., Jain, A., Chauhan, A., Kotla, S. S., & Gautam, A. (2021). American sign language recognition using support vector machine and convolutional neural network. *International Journal of Information Technology (Singapore)*, 13(3), 1193-1200. <https://doi.org/10.1007/s41870-021-00617-x>
- Kattenborn, T., Leitloff, J., Schiefer, F., & Hinz, S. (2021). Review on convolutional neural networks (cnn) in vegetation remote sensing. *ISPRS Journal of Photogrammetry and Remote Sensing*, 173, 24-49. <https://doi.org/10.1016/j.isprsjprs.2020.12.010>
- Lee, C. K. M., Ng, K. K. H., Chen, C. H., Lau, H. C. W., Chung, S. Y., & Tsoi, T. (2021). American sign language recognition and training method with recurrent neural network. *Expert Systems with Applications*, 167. <https://doi.org/10.1016/j.eswa.2020.114403>
- Lim, K. M., Tan, A. W. C., Lee, C. P., & Tan, S. C. (2019). Isolated sign language recognition using convolutional neural network hand modelling and hand energy image. *Multimedia Tools and Applications*, 78(14), 19917-19944. <https://doi.org/10.1007/s11042-019-7263-7>
- Maharana, K., Mondal, S., & Nemade, B. (2022). A review: data pre-processing and data augmentation techniques. *Global Transitions Proceedings*, 3(1), 91-99. <https://doi.org/10.1016/j.gltp.2022.04.020>
- Manaswi, N. K. (2018). Deep learning with applications using python. Apress. <https://doi.org/10.1007/978-1-4842-3516-4>
- Nevavuori, P., Narra, N., & Lipping, T. (2019). Crop yield prediction with deep convolutional neural networks. *Computers and Electronics in Agriculture*, 163. <https://doi.org/10.1016/j.compag.2019.104859>
- Rafi, A. M., Nawal, N., Nima, L., Bayev, N. S. N., Shahna, Dr. C., & Fattah, S. A. (2019). Image-based bengali sign language alphabet recognition for deaf and dumb community. *2019 IEEE Global Humanitarian Technology Conference (GHTC)*, Seattle, WA, USA, 2019, 1-7, doi: <https://doi.org/10.1109/GHTC46095.2019.9033031>
- Rathore, P. W. S., Hussain, M., Malik, M. B., & Amin, Y. (2023). Well log analysis and comparison of supervised machine learning algorithms for lithofacies identification in pab formation, lower indus basin. *Journal of Applied Geophysics*, 105199. <https://doi.org/10.1016/j.jappgeo.2023.105199>
- Sharma, S., & Kumar, K. (2021). ASL-3DCNN: American sign language recognition technique using 3-D convolutional neural networks. *Multimedia Tools and Applications*, 80(17), 26319-26331. <https://doi.org/10.1007/s11042-021-10768-5>

- Soppari, K., & Chandra, N. S. (2020). Development of improved whale optimization-based FCM clustering for image watermarking. *Computer Science Review*, 37. <https://doi.org/10.1016/j.cosrev.2020.100287>
- Vujović, Ž. Đ. (2021). Classification model evaluation metrics. *IJACSA International Journal of Advanced Computer Science and Applications*, 12(6). <https://doi.org/10.14569/IJACSA.2021.0120670>
- Xiao, H., Yang, Y., Yu, K., Tian, J., Cai, X., Muhammad, U., & Chen, J. (2022). Sign language digits and alphabets recognition by capsule networks. *Journal of Ambient Intelligence and Humanized Computing*, 13(4), 2131-2141. <https://doi.org/10.1007/s12652-021-02974-8>
- Xu, J., Zhang, Y., & Miao, D. (2020). Three-way confusion matrix for classification: A measure driven view. *Information Sciences*, 507, 772-794. <https://doi.org/10.1016/j.ins.2019.06.064>
- Yun, J. P., Shin, W. C., Koo, G., Kim, M. S., Lee, C., & Lee, S. J. (2020). Automated defect inspection system for metal surfaces based on deep learning and data augmentation. *Journal of Manufacturing Systems*, 55, 317–324. <https://doi.org/10.1016/j.jmsy.2020.03.009>