



KLASIFIKASI ALFABET BAHASA ISYARAT INDONESIA (BISINDO) DENGAN MENGGUNAKAN ALGORITMA CONVOLUTIONAL NEURAL NETWORK

Intan Patresia Tambun*

Program Studi Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Negeri Malang

*Email : intan.patresia.2003126@students.um.ac.id

Abstrak

Bahasa Isyarat Indonesia (BISINDO) adalah sistem komunikasi yang digunakan oleh komunitas penyandang tunarungu dan tunawicara di Indonesia. Namun, tidak semua masyarakat Indonesia memahami bahasa isyarat yang digunakan. Dengan perkembangan teknologi yang semakin pesat, memungkinkan adanya penerjemah secara real-time untuk membantu komunikasi antara komunitas tunawicara dan tunarungu dengan masyarakat umum.

Penelitian ini melakukan pendekatan klasifikasi BISINDO menggunakan algoritma Convolutional Neural Network (CNN) dengan memanfaatkan pustaka seperti MediaPipe untuk mendeteksi dan melacak tangan dalam video, kemudian mengekstrak fitur yang relevan dari setiap frame menggunakan OpenCV. Pra-pemrosesan data dilakukan dengan melakukan normalisasi dan augmentasi untuk meningkatkan keberagaman dan kualitas dataset. Dataset ini kemudian dilatih dengan menggunakan algoritma CNN. Model terdiri dari lapisan konvolusi, pooling, dan dense, kemudian menggunakan teknik dropout untuk mengurangi overfitting. Model CNN yang telah dilatih mencapai akurasi sekitar 90% dalam mengenali alfabet BISINDO.

Penelitian ini memberikan kontribusi penting untuk deteksi dan klasifikasi otomatis alfabet BISINDO dan membuka jalan bagi pengembangan sistem lebih lanjut untuk meningkatkan komunikasi dan inklusi sosial penyandang tunarungu dan tunawicara di masyarakat.

Kata kunci: Convolutional Neural Network (CNN), BISINDO, Tensorflow, MediaPipe, OpenCV

PENDAHULUAN

Bahasa isyarat merupakan salah satu medium untuk berkomunikasi bagi kaum tunarungu dan tunawicara, tetapi tidak banyak masyarakat umum yang dapat menggunakannya [1]. Menurut Badan Pusat Statistik (BPS), jumlah penduduk Indonesia yang disabilitas pada tahun 2020 adalah 22,5 juta, dimana 29% merupakan komunitas tunarungu dan 16,5% merupakan komunitas tunawicara. Bahasa isyarat dilakukan dengan menggunakan gestur tangan dan wajah dalam mengekspresikan pemikiran kaum disabilitas kepada masyarakat umum. Hal ini menciptakan kesenjangan komunikasi yang cukup besar. Dengan perkembangan teknologi yang semakin pesat, banyak diciptakan alat bantu komunikasi untuk kaum disabilitas dengan masyarakat umum [2]. Alat bantu inilah yang akan membantu mengurangi kesenjangan komunikasi antara komunitas tunarungu, tunawicara, dan masyarakat umum. Bahasa isyarat sendiri ada banyak jenisnya di seluruh dunia, mulai dari Bahasa Isyarat Indonesia (BISINDO), Sistem Isyarat Bahasa Indonesia (SIBI), American Sign Language (ASL), dan lainnya. Di Indonesia sendiri, BISINDO merupakan Bahasa isyarat yang muncul secara alamiah pada komunitas disabilitas dan digunakan dalam kehidupan sehari-hari.

Pada penelitian ini, akan dilakukan klasifikasi alfabet Bahasa Isyarat Indonesia (BISINDO) dengan menggunakan algoritma CNN. Convolutional Neural Network (CNN) merupakan salah satu arsitektur dalam Deep Learning. CNN muncul dari studi mengenai korteks otak, dan telah digunakan pada pengenalan gambar komputer sejak tahun 1980-an. CNN telah berhasil mencapai kinerja yang melebihi manusia pada beberapa permasalahan visual komputer yang kompleks dalam 10 tahun terakhir ini dikarenakan pesatnya perkembangan komputer. Tidak hanya pada permasalahan visual komputer seperti sistem pencarian gambar, mobil dengan kemampuan self-driving, sistem klasifikasi video otomatis, tetapi pengenalan suara dan juga pengolahan bahasa alami (natural language processing) [3]. Selain algoritma CNN, penelitian ini juga menggunakan pustaka seperti MediaPipe untuk mendeteksi tangan, serta OpenCV untuk mengekstrak fitur yang relevan dari setiap frame.

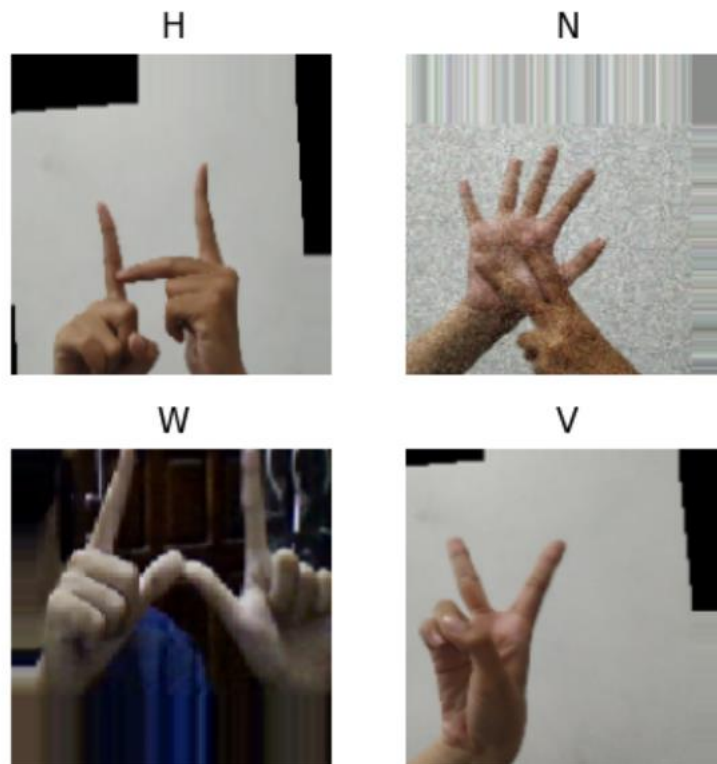
Penelitian ini bertujuan untuk menghasilkan model CNN yang dapat mengenali alfabet BISINDO dengan tingkat akurasi >85% agar dapat dikembangkan menjadi aplikasi yang membantu mengurangi kesenjangan dalam berkomunikasi.

METODE

Penelitian ini dilakukan dalam beberapa tahap, yakni pengumpulan data, pra-pemrosesan data, augmentasi data, membangun model, kemudian menguji model. Algoritma yang akan digunakan dalam penelitian ini adalah CNN untuk klasifikasi gambar.

Pengumpulan data dan pra-pemrosesan data dilakukan dengan memanfaatkan dataset yang telah tersedia di *Kaggle* kemudian menambahkannya secara manual. Total dataset yang digunakan adalah sebanyak 3095 gambar, di mana 80% merupakan data untuk pelatihan dan 20% digunakan untuk pengujian. Ketika dataset telah dipisah. Pra-pemrosesan data dilakukan dengan *data cleaning*. Langkah ini melibatkan identifikasi nilai yang hilang, data yang tidak valid, ataupun data yang tidak relevan. Pada tahapan ini, peneliti melakukan penghapusan data yang tidak sesuai dan pemilihan atribut yang lengkap.

Kemudian akan dilakukan augmentasi data dengan membuat objek 'ImageDataGenerator' dengan parameter augmentasi seperti, *rescale*, *shear_range*, *width_shift_range*, *height_shift_range*, *zoom_range*, *fill_mode*, dan *horizontal_flip*. Pada tahap ini juga akan dibuat generator pelatihan dan generator validasi menggunakan metode 'flow_from_directory' dari 'ImageDataGenerator'. Generator ini kemudian akan memuat gambar dari direktori pelatihan dan direktori validasi, dan mengaplikasikan augmentasi data pada gambar saat membangkitkan batch data. Variasi augmentasi yang telah diterapkan dapat dilihat pada gambar berikut.



Gambar 1. Hasil Augmentasi Data

Setelah melakukan pra-pemrosesan dan augmentasi data, tahap selanjutnya adalah membangun model menggunakan algoritma CNN. Sebelum membangun model, akan ditambahkan fungsi *callback* yang berfungsi untuk menghentikan pelatihan ketika suatu keadaan sudah terpenuhi. Manfaat dari mendefinisikan fungsi *callback* adalah untuk membantu menghemat waktu dan sumber daya komputasi dengan menghindari pelatihan yang berlebihan dan mencegah *overfitting*. Fungsi *callback* pada penelitian ini menggunakan metrik $loss < 0.2$ untuk menghentikan pelatihan jika *loss* yang dihasilkan sudah kurang dari 20%.

Secara umum, CNN mempunyai tiga jenis lapisan, yaitu lapisan konvolusi, pooling, dan lapisan yang terhubung penuh (*dense*). Model yang telah dibangun adalah sebagai berikut.

```
model = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(16, (3,3), activation='relu', input_shape=input_shape),
    tf.keras.layers.MaxPooling2D(2,2),
```

```

tf.keras.layers.Conv2D(32, (3,3), activation='relu'),
tf.keras.layers.MaxPooling2D(2,2),
tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
tf.keras.layers.MaxPooling2D(2,2),
tf.keras.layers.Flatten(),
tf.keras.layers.Dense(256, activation='relu'),
tf.keras.layers.Dropout(0.5),
tf.keras.layers.Dense(26, activation='softmax')
]

```

Berikut merupakan penjelasan mengenai baris kode di atas.

- Baris kode `tf.keras.models.Sequential` akan membuat model barisan (*sequential*), di mana dapat menambahkan lapisan-lapisan konvolusi, pooling, ataupun dense [3].
- Baris kode `tf.keras.layers.Conv2D` merupakan lapisan konvolusi 2D. Pada lapisan konvolusi, kisi informasi diduplikasi dengan beberapa konvolusi bagian untuk memberikan peta elemen yang menjelaskan apa jenis sorotan yang bertahan dalam sebuah gambar [2]. Lapisan konvolusi yang pertama memiliki 16 filter dengan ukuran (3,3) dan menggunakan fungsi aktivasi ReLU.
- Baris kode `tf.keras.layers.MaxPooling2D` merupakan lapisan pooling maksimal yang bertujuan untuk menurunkan sampel peta fitur [2]. Ukuran pooling yang digunakan adalah (2,2).
- Baris kode `tf.keras.layers.Flatten()` merupakan lapisan yang memetakan fitur 2D ke vektor 1D [3].
- Baris kode `tf.keras.layers.Dense` merupakan lapisan terkoneksi penuh yang menjalankan klasifikasi pada fitur yang terekstraksi. Lapisan *dense* yang pertama memiliki 256 unit dan menggunakan fungsi aktivasi ReLU. Lapisan *dense* yang kedua memiliki 26 unit dan menggunakan fungsi aktivasi softmax.
- Baris kode `tf.keras.layers.Dropout` merupakan lapisan yang mengaplikasikan regularisasi *dropout* ke dalam lapisan sebelumnya untuk mencegah *overfitting*.

Model yang telah dibangun ini kemudian dioptimasi menggunakan *optimizer* Adam dan menggunakan fungsi *loss* `categorical_crossentropy`. Ringkasan model yang telah dibangun dapat dilihat di bawah ini.

Model: "sequential"

| Layer (type) | Output Shape | Param # |
|---------------------------------|----------------------|---------|
| conv2d (Conv2D) | (None, 148, 148, 16) | 448 |
| max_pooling2d (MaxPooling2D) | (None, 74, 74, 16) | 0 |
| conv2d_1 (Conv2D) | (None, 72, 72, 32) | 4640 |
| max_pooling2d_1 (MaxPooling 2D) | (None, 36, 36, 32) | 0 |
| conv2d_2 (Conv2D) | (None, 34, 34, 64) | 18496 |
| max_pooling2d_2 (MaxPooling 2D) | (None, 17, 17, 64) | 0 |
| flatten (Flatten) | (None, 18496) | 0 |
| dense (Dense) | (None, 256) | 4735232 |
| dropout (Dropout) | (None, 256) | 0 |
| dense_1 (Dense) | (None, 26) | 6682 |

=====
Total params: 4,765,498
Trainable params: 4,765,498
Non-trainable params: 0

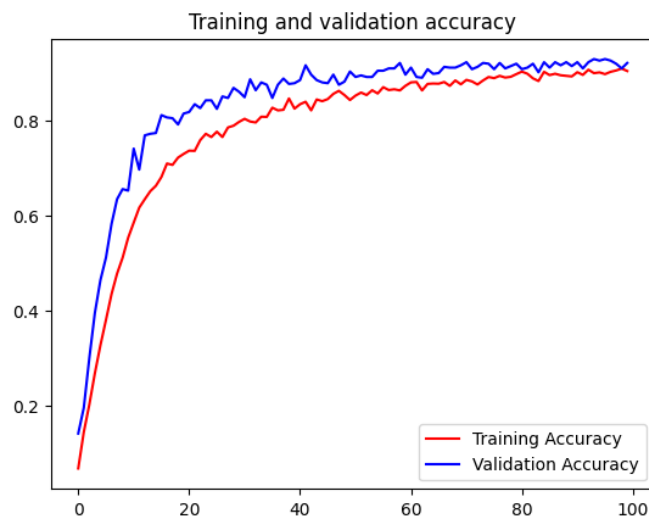
HASIL DAN PEMBAHASAN

Pembentukan model dilakukan pada Visual Studio Code dengan Bahasa pemrograman Python. Hasil dari pelatihan model yang dilakukan sebanyak 100 *epoch* adalah sebagai berikut.

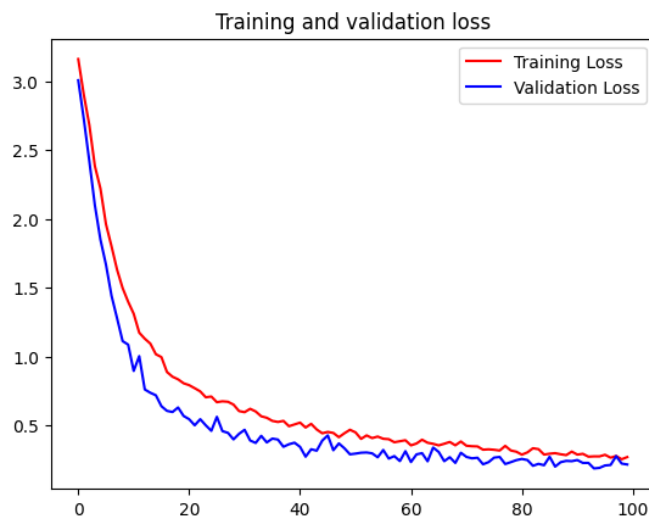
Epoch 100

Train Loss: 0.2707 - Train Accuracy: 0.9029

Validation Loss: 0.2169 - Validation Accuracy: 0.9199



Gambar 2. Grafik Akurasi Pelatihan dan Validasi

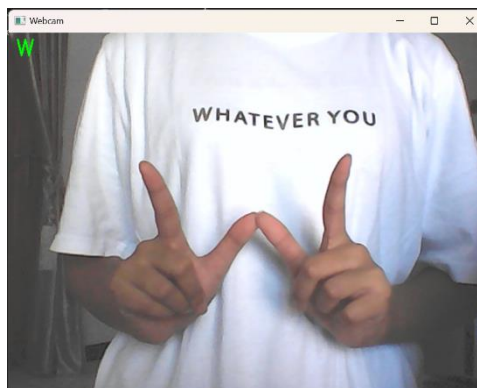


Gambar 3. Grafik Loss Pelatihan dan Validasi

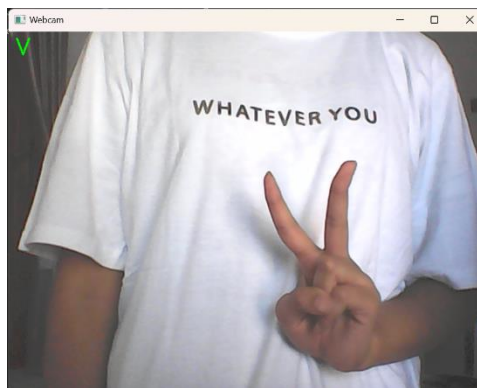
Dilihat dari akurasi, model yang dihasilkan cukup baik dalam memprediksi alfabet BISINDO. Model yang dihasilkan kemudian disimpan ke dalam format HDF5 atau .h5. Model inilah yang akan digunakan untuk memprediksi melalui *webcam*. Untuk menampilkan hasil prediksi secara *real-time* diperlukan bantuan pustaka seperti MediaPipe dan OpenCV. MediaPipe berfungsi untuk menganalisis citra dan ekstraksi fitur, sedangkan OpenCV berfungsi untuk memanipulasi gambar dan video dari kamera. Beberapa hasil yang didapat dari prediksi secara *real-time* adalah sebagai berikut.



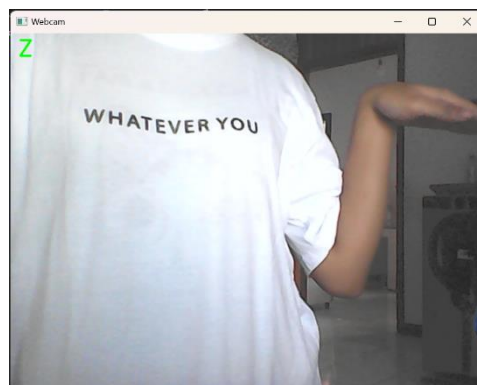
Gambar 4. Hasil Prediksi Huruf F



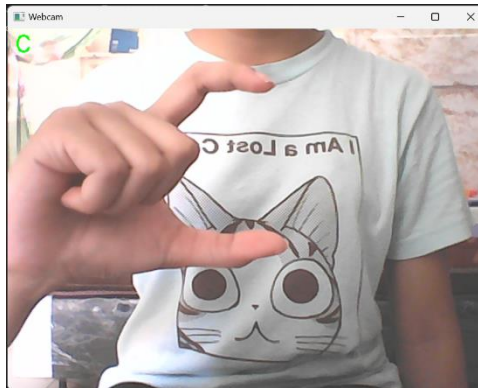
Gambar 5. Hasil Prediksi Huruf W



Gambar 6. Hasil Prediksi Huruf V



Gambar 7. Hasil Prediksi Huruf Z



Gambar 8. Hasil Prediksi Huruf C

PENUTUP

Model yang telah dihasilkan cukup baik dalam memprediksi alfabet BISINDO. Walau demikian, model yang dihasilkan masih mengalami *overfitting*, sehingga masih harus diperbaiki agar mendapatkan hasil prediksi yang lebih akurat. Diharapkan melalui penelitian ini dapat membantu komunitas tunarungu dan tunawicara dalam berkomunikasi kepada masyarakat umum dan dapat mengurangi kesenjangan dalam berkomunikasi.

DAFTAR RUJUKAN

- [1] M. Kumar, P. Gupta, R. K. Jha, A. Bhatia, K. Jha, dan B. K. Shah, “Sign language alphabet recognition using convolution neural network,” dalam *Proceedings - 5th International Conference on Intelligent Computing and Control Systems, ICICCS 2021*, Institute of Electrical and Electronics Engineers Inc., Mei 2021, hlm. 1859–1865. doi: 10.1109/ICICCS51141.2021.9432296.
- [2] M. M. Hasan, A. Y. Srizon, A. Sayeed, dan M. A. M. Hasan, “Classification of sign language characters by applying a deep convolutional neural network,” dalam *2020 2nd International Conference on Advanced Information and Communication Technology, ICAICT 2020*, Institute of Electrical and Electronics Engineers Inc., Nov 2020, hlm. 434–438. doi: 10.1109/ICAICT51780.2020.9333456.
- [3] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 3 ed. O’Reilly Media, Inc., 2023. [Daring]. Tersedia pada: <https://oreilly.com>