

Sistem Klasifikasi Alfabet Bahasa Isyarat Indonesia Menggunakan CNN dengan MobileNetV2 berbasis Android

Reihan Saputra*¹, Gentur Wahyu Nyipto Wibowo², Akhmad Khanif Zyen³

^{1,2} Teknik Informatika, Universitas Islam Nahdlatul Ulama Jepara, Jepara

³ Jl. Taman Siswa, Pekeng, Kauman, Tahunan, Kec. Tahunan, Kabupaten Jepara, Jawa Tengah

e-mail: *¹rehansaputra1006@gmail.com, ²gentur@unisnu.ac.id, ³khanif.zyen@unisnu.ac.id

Abstrak

Komunikasi merupakan kebutuhan dasar manusia, termasuk bagi tunarungu dan tunawicara yang menghadapi kesulitan dalam berinteraksi menggunakan bahasa verbal. Salah satu permasalahannya adalah kurangnya teknologi yang optimal dalam mendukung komunikasi dengan Bahasa Isyarat Indonesia (BISINDO) secara real-time. Penelitian ini bertujuan untuk menguji kemampuan MobileNetV2 dalam mengklasifikasikan alfabet BISINDO sebagai solusi teknologi yang mendukung komunikasi bagi tunarungu dan tunawicara secara real-time. Dataset penelitian ini terdiri dari 5200 gambar alfabet BISINDO (200 gambar per kelas) yang diproses dengan augmentasi data. Data dibagi menjadi data train 80%, data validation 10%, dan data test 10%. Transfer learning digunakan untuk melatih model MobileNetV2 dengan bobot awal dari ImageNet. Model dilengkapi Global Average Pooling, Batch Normalization, dan Dropout untuk meningkatkan performa. Pelatihan menggunakan optimizer Adam dengan learning rate $1e-4$. Evaluasi menghasilkan akurasi validasi sebesar 95% dan pengujian 93,85%. Model dikonversi ke TensorFlow Lite untuk digunakan pada aplikasi Android, sehingga klasifikasi alfabet BISINDO dapat dilakukan secara real-time melalui kamera. Aplikasi ini diharapkan menjadi solusi bagi masyarakat yang menghadapi kesulitan dalam berkomunikasi dengan tunarungu dan tunawicara.

Kata kunci— Convolutional Neural Network (CNN), MobileNetV2, Bahasa Isyarat

Abstract

Communication is a basic human need, including for deaf and hard of hearing people who face difficulties in interacting using verbal language. One of the problems is the lack of optimal technology in supporting communication with Indonesian Sign Language (BISINDO) in real-time. This research aims to test the ability of MobileNetV2 in classifying the BISINDO alphabet as a technological solution that supports communication for the deaf and hard of hearing in real-time. This research dataset consists of 5200 BISINDO alphabet images (200 images per class) processed with data augmentation. The data is divided into 80% train data, 10% validation data, and 10% test data. Transfer learning was used to train the MobileNetV2 model with initial weights from ImageNet. The model is equipped with Global Average Pooling, Batch Normalization, and Dropout to improve performance. Training used Adam's optimizer with a learning rate of $1e-4$. Evaluation resulted in validation accuracy of 95% and testing accuracy of 93.85%. The model is converted to TensorFlow Lite for use in Android applications, so that BISINDO alphabet classification can be done in real-time through the camera. This application is expected to be a solution for people who face difficulties in communicating with the deaf and speech impaired.

Keywords— Convolutional Neural Network (CNN), MobileNetV2, Sign Language

1. PENDAHULUAN

Komunikasi terjadi ketika dua orang atau lebih berinteraksi menggunakan bahasa yang sama dan dapat dimengerti dan dipahami oleh lawan bicara [1]. Bahasa adalah suatu bentuk pemahaman yang diperoleh melalui kebiasaan sehari-hari atau hasil dari proses pembelajaran, yang digunakan untuk menyampaikan informasi kepada orang lain [2]. Bahasa isyarat adalah komunikasi visual yang menggunakan gerakan tangan, jari, serta ekspresi wajah [3]. Di Indonesia, terdapat dua bentuk bahasa isyarat, yakni Sistem Isyarat Bahasa Indonesia (SIBI), yang bersifat lebih formal [4]. dan Bahasa Isyarat Indonesia (BISINDO), yang dikembangkan oleh GERKATIN dan komunitas tunarungu serta lebih umum digunakan dalam komunikasi sehari-hari [5]. Meskipun demikian, pemahaman masyarakat terhadap bahasa isyarat masih terbatas, sehingga penggunaannya belum optimal dalam interaksi sosial.

Keterbatasan pemahaman masyarakat terhadap bahasa isyarat menjadi kendala utama dalam komunikasi bagi penyandang tunarungu, yang berakibat pada hambatan dalam interaksi sosial [6]. Banyak individu yang tidak memiliki kemampuan memahami isyarat BISINDO, sehingga menyulitkan komunikasi sehari-hari [7]. Hambatan ini tidak hanya memengaruhi kehidupan sosial penyandang tunarungu, tetapi juga membatasi akses mereka terhadap pendidikan, layanan kesehatan, dan kesempatan kerja. Dalam situasi darurat atau di lingkungan publik, kesulitan berkomunikasi dapat menyebabkan keterlambatan dalam menerima bantuan atau informasi penting [8].

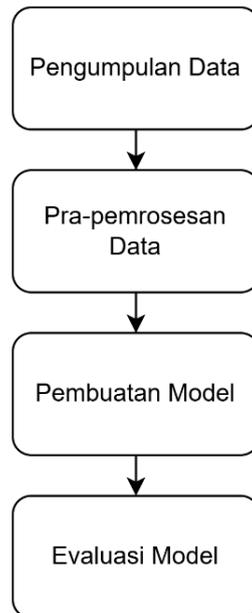
Berdasarkan data Sistem Informasi Manajemen Penyandang Disabilitas Kementerian Kesehatan RI, jumlah penyandang disabilitas meningkat dari 197.582 pada Maret 2020 menjadi 212.240 pada Maret 2022. Dari jumlah tersebut, penyandang tunarungu dan tunawicara mencapai 19.392 orang, sekitar 9,14% dari total [9]. Data tersebut menunjukkan perlunya pengembangan sistem berbasis teknologi kecerdasan buatan, pendekatan berbasis computer vision dapat digunakan untuk mengolah data visual berupa gambar atau video demi mengenali pola pada isyarat BISINDO [10]. Pada sistem berbasis computer vision, Convolutional Neural Network (CNN) menjadi metode yang banyak digunakan karena kemampuannya dalam mengenali pola visual secara akurat [11], termasuk alfabet BISINDO, sehingga dapat meningkatkan aksesibilitas komunikasi bagi penyandang tunarungu dan tunawicara [12].

Penelitian terkait bahasa isyarat menggunakan citra huruf bahasa isyarat indonesia telah dilakukan sebelumnya, adapun penelitian yang menggunakan metode *Long-Short Term Memory* menghasilkan akurasi 85 % [13]. Penelitian lain menggunakan algoritma *Convolutional Neural Network* untuk Terjemahkan Abjad Bahasa Isyarat SIBI menghasilkan akurasi 81,48% [14]. Penelitian lain menggunakan arsitektur *CNN* dengan tiga *Convolution Layer*, tiga *Pooling Layer*, dan satu *Fully Connected Layer*, menghasilkan akurasi validasi 87% dengan rata-rata akurasi 87,9%. Pada penelitian tersebut diperlukan penambahan dataset, eksplorasi parameter lain, serta pengembangan implementasi ke platform mobile [15]. Penelitian lainnya menggunakan *CNN* dengan citra berukuran 256x256 piksel pada sistem klarifikasi BISINDO, mencapai akurasi 99,82%, namun terbatas pada 9 kelas data dan hanya memprediksi gambar statis [16].

Tujuan dari penelitian ini adalah untuk menguji kemampuan model *Convolutional Neural Network (CNN)* dengan arsitektur *MobileNetV2* dalam mengklasifikasikan 26 *class* alfabet Bahasa Isyarat Indonesia (BISINDO) secara *real-time* pada platform android. *CNN* dipilih karena memiliki keunggulan dalam mengenali pola visual, sementara *MobileNetV2* dipilih karena ukuran modelnya kecil dan memanfaatkan *separable convolution* untuk mengurangi beban komputasi tanpa mengurangi kinerja [17], [18]. Keunggulan ini membuat *MobileNetV2* sangat ideal untuk diintegrasikan ke perangkat mobile [19], terutama pada platform Android yang menawarkan portabilitas, kemudahan penggunaan, dan kemampuan untuk mendukung pengenalan alfabet BISINDO secara real-time oleh pengguna secara luas.

2. METODE PENELITIAN

Penelitian ini melibatkan empat tahap utama yaitu pengumpulan data, pra-pemrosesan data, pembuatan model, dan evaluasi model.



Gambar 1 Tahapan Penelitian

2.1 Pengumpulan Data

Dataset penelitian ini memanfaatkan dataset alfabet Bahasa Isyarat Indonesia (BISINDO) yang tersedia di platform publik, yaitu data.mendeley.com. Dataset yang digunakan berjudul "*BISINDO Indonesian Sign Language: Alphabet Image Data*", yang dapat diakses melalui tautan <https://data.mendeley.com/datasets/ywnjpbcz8m/1>. Dataset ini terdiri dari 26 kelas, yang masing-masing merepresentasikan huruf alfabet. Setiap kelas memiliki 200 gambar, sehingga total keseluruhan gambar dalam dataset adalah 5200 gambar. Gambar-gambar tersebut digunakan sebagai data pelatihan dan pengujian dalam penelitian ini [20].

2.2 Pra-pemrosesan Data

Tahapan pra-pemrosesan data bertujuan mempersiapkan dataset agar sesuai dengan kebutuhan model dan memastikan pelatihan berlangsung optimal. Dataset dibagi menjadi 80% untuk data pelatihan, 10% untuk data validasi, dan 10% untuk data pengujian menggunakan pustaka *splitfolders*. Semua gambar diubah ukurannya menjadi 224x224 piksel menggunakan *cv2.resize* agar sesuai dengan spesifikasi *MobileNetV2*. Augmentasi data, yaitu proses menambahkan variasi pada dataset dengan melakukan transformasi pada gambar, diterapkan menggunakan pustaka *Albumentations* untuk meningkatkan keragaman data dan kemampuan generalisasi model. Teknik augmentasi yang digunakan meliputi *Gaussian Blur*, *Hue Saturation Value*, *Rotate*, *Horizontal Flip*, dan lain-lain. Augmentasi bertujuan agar model dapat membaca data walaupun data diambil dari sudut yang beraneka ragam, sehingga model lebih *robust* terhadap data baru [21]. Gambar 2 menunjukkan hasil berbagai teknik augmentasi pada dataset, termasuk penambahan *noise* untuk mensimulasikan variasi pencahayaan dan kualitas gambar.

Noise ini bertujuan meningkatkan kemampuan model menghadapi data berkualitas beragam saat pengujian.



Gambar 2 Augmentasi Citra

Gambar juga dinormalisasi ke rentang 0 sampai 1 untuk mempercepat pelatihan dan meningkatkan stabilitas optimasi. Augmentasi tambahan meliputi *rotation*, *shearing*, *zoom*, penyesuaian kecerahan, dan *channel shifting* untuk menghasilkan variasi dataset yang lebih beragam. Proses ini dilengkapi generator data untuk menerapkan augmentasi, menyesuaikan ukuran, dan menormalkan nilai piksel secara otomatis dalam *batch* sebelum melatih model, sehingga mendukung pelatihan yang baik dan mengurangi risiko *overfitting*.

2.3 Pembuatan Model

Setelah melalui tahapan pra-pemrosesan data, selanjutnya adalah membangun model dengan menggunakan *MobileNetV2* sebagai arsitektur dasar. *MobileNetV2*, yang merupakan pengembangan dari *MobileNet*, dirancang untuk menghasilkan efisiensi tinggi dengan menggunakan teknik *depthwise separable convolution*. Teknik ini membagi proses konvolusi menjadi dua tahap, yaitu *depthwise convolution*, yang melakukan konvolusi pada setiap saluran input secara individual, serta konvolusi pointwise (konvolusi 1x1) yang menggabungkan hasil dari *depthwise convolution*. Pendekatan ini secara signifikan mengurangi jumlah parameter dan komputasi tanpa mengurangi kinerja model.

Model yang dibangun pada penelitian ini menggunakan ukuran input gambar 224×224 piksel dengan tiga kanal warna yaitu *RGB*, yang sesuai dengan spesifikasi *MobileNetV2*. Arsitektur model terdiri dari beberapa komponen utama, dimulai dengan Input Layer untuk menerima data input berupa gambar berukuran (224, 224, 3). Bagian utama model adalah *Backbone MobileNetV2*, yang bertugas mengekstraksi fitur menggunakan kombinasi *convolution* dan *depthwise separable blocks*. Setelah itu, dilakukan agregasi fitur dengan menggunakan *Global Average Pooling 2D* untuk meratakan keluaran dari backbone. Kemudian, ditambahkan dua lapisan *Dense* (*Dense Layer 1* dan *Dense Layer 2*) masing-masing dengan 256 *neuron* yang memanfaatkan fungsi aktivasi *ReLU*, disertai *Batch Normalization* dan *Dropout* untuk mencegah *overfitting*. Lapisan terakhir adalah Output Layer, berupa *Dense Layer* dengan 26 *neuron* yang

menggunakan fungsi aktivasi yaitu *softmax* untuk menghasilkan probabilitas klasifikasi pada setiap kelas. Struktur arsitektur model dapat dilihat pada Tabel 1 Arsitektur Model berikut.

Tabel 1 Arsitektur Model

Layer Group	Type Output	Shape
Input Layer	Input Layer	(None, 224, 224, 3)
Backbone (MobileNetV2)	Conv + Depthwise Blocks	Feature Extraction
Global Pooling	Global Average Pooling2D	(None, 512)
Dense Layer 1	Dense	(None, 256)
Dense Layer 2	Dense	(None, 256)
Output Layer	Softmax	(None, 256)

Bobot awal model ini diinisialisasi menggunakan dataset *ImageNet* untuk memanfaatkan pengetahuan yang telah dipelajari pada data skala besar yaitu *transfer learning*. Untuk meningkatkan efisiensi proses pelatihan, sebagian besar lapisan *MobileNetV2* dibekukan, sementara hanya 50 lapisan terakhir yang dibiarkan dapat dilatih. Model ini dioptimasi menggunakan algoritma *Adam* dengan laju pembelajaran sebesar $1e-4$, dan *loss function* yang digunakan adalah *categorical crossentropy*, sesuai dengan tugas klasifikasi multi-kelas. Untuk menjaga stabilitas pelatihan, diterapkan beberapa *callback* seperti *Early Stopping*, *Model Checkpoint*, dan *Reduce Learning Rate on Plateau*. Kombinasi ini memastikan pelatihan model berjalan efisien dan menghasilkan performa terbaik.

2. 4 Evaluasi Model

Penelitian ini menggunakan matrix accuracy, precision, recall, serta F1-score untuk mengevaluasi kinerja model dalam klasifikasi. Evaluasi dilakukan berdasarkan hasil dari confusion matrix, yang memberikan informasi mengenai jumlah *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). TP menggambarkan jumlah data positif yang diprediksi dengan tepat, sementara TN menggambarkan jumlah data negatif yang

diprediksi dengan tepat. FP merujuk pada jumlah data negatif yang salah diprediksi sebagai positif, sementara FN adalah jumlah data positif yang keliru diprediksi sebagai negatif..

Tabel 2 accuracy, precision, recall, dan F1-score

		Kelas Sebenarnya	
		Positive	Negative
Prediksi	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

Accuracy didefinisikan sebagai rasio antara jumlah prediksi yang benar dengan total keseluruhan data prediksi. Rumus untuk akurasi diberikan pada Persamaan (1).

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

Precision adalah perbandingan antara jumlah prediksi yang benar untuk suatu label dengan total keseluruhan prediksi, baik yang benar maupun salah, untuk label tersebut. *Precision* dapat dihitung dengan membagi jumlah *True Positive* oleh total *Predicted Positive*, sebagaimana ditunjukkan pada Persamaan (2).

$$Precision = \frac{TP}{FP+TP} \quad (2)$$

Recall digunakan untuk mengevaluasi sejauh mana model dapat menghindari *False Negative*, yang dihitung sebagai rasio *True Positive* terhadap *Actual Positive*. Rumus *recall* dapat ditemukan pada Persamaan (3).

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

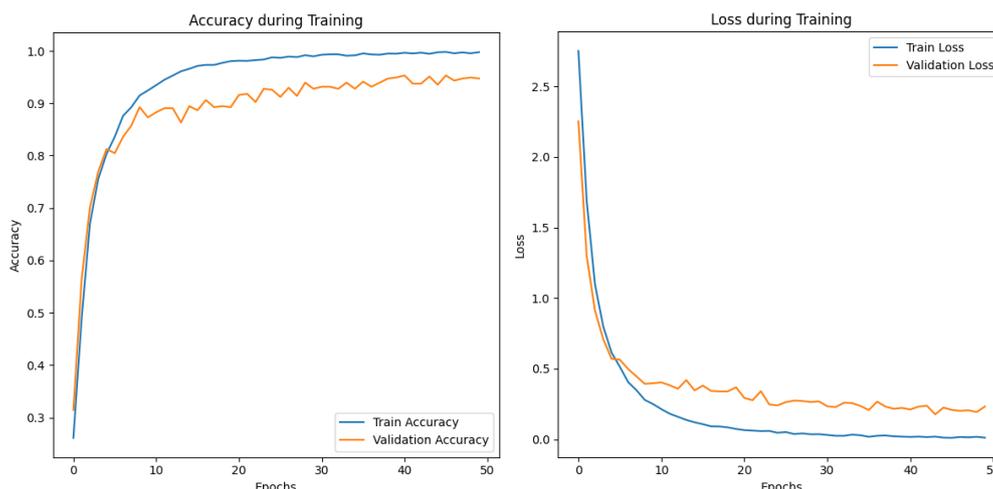
F1-score adalah rata-rata gabungan antara *precision* dan *recall*, yang memberikan keseimbangan antara kedua metrik tersebut. *F1-score* dirumuskan sebagaimana tertera pada Persamaan (4).

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

3. HASIL DAN PEMBAHASAN

3. 1 Hasil Pelatihan

Pada penelitian ini, proses pelatihan model dilakukan dengan menggunakan dataset alfabet BISINDO yang telah melalui tahapan pra-pemrosesan dan augmentasi. Pelatihan model menggunakan arsitektur *MobileNetV2* dengan konfigurasi *hyperparameter* seperti *learning rate* sebesar $1e-4$, *loss function categorical crossentropy*, dan *optimizer Adam*. Model dilatih selama 50 *epoch* dengan menggunakan *callback* untuk mencegah *overfitting*, seperti *Early Stopping* dan *Reduce Learning Rate on Plateau*.

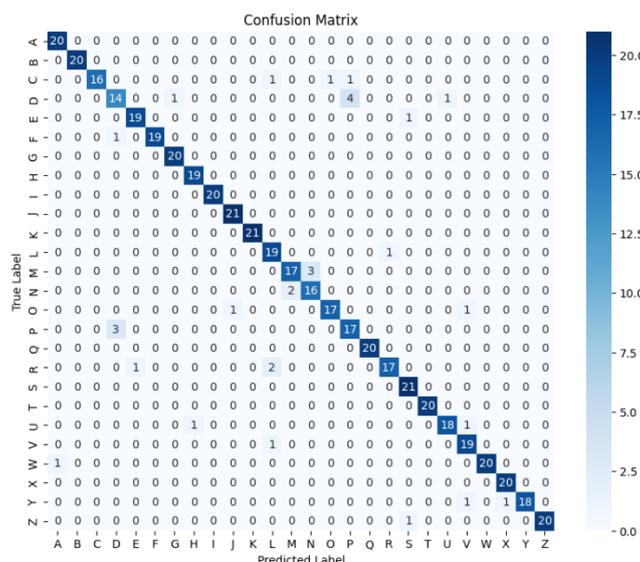


Gambar 3 Grafik Training Accuracy dan Loss

Grafik pertama pada gambar 3 menunjukkan akurasi pada data pelatihan terus meningkat secara konsisten hingga mendekati 100%, menandakan bahwa model mampu mempelajari pola dalam data pelatihan dengan baik. Akurasi pada data validasi juga menunjukkan peningkatan hingga mencapai sekitar 95%. Pola peningkatan yang stabil pada kedua metrik ini menunjukkan bahwa model tidak mengalami *overfitting* yang signifikan selama proses pelatihan. Pada grafik kedua pada gambar 3 memperlihatkan *train loss* terus menurun secara konsisten dengan bertambahnya *epoch*, menunjukkan bahwa model semakin baik dalam memprediksi label pada data pelatihan. *Validation loss* juga mengalami penurunan, meskipun terdapat sedikit fluktuasi pada beberapa titik.

3.2 Hasil Pengujian

Pada tahap pengujian, model dievaluasi menggunakan data uji untuk mengukur performa dalam mengklasifikasikan alfabet. Gambar di bawah ini menunjukkan *confusion matrix* yang dihasilkan dari proses pengujian model pada data uji:



Gambar 4 Hasil Pengujian *Confusion Matrix*

Confusion Matrix pada Gambar 4 terlihat sebagian besar nilai berada di diagonal utama, menunjukkan model berhasil memprediksi label dengan benar. Sebagai contoh, huruf A, B, dan C diprediksi sempurna untuk seluruh sampel. Namun, terdapat beberapa kesalahan, seperti huruf D yang hanya 14 dari 20 sampel diprediksi benar, sedangkan 4 lainnya salah sebagai huruf E, dan

huruf M yang 3 sampel salah diklasifikasikan sebagai huruf N, menunjukkan adanya kemiripan pola antar huruf. hasil pengujian juga ditampilkan dalam laporan *precision*, *recall*, *f1-score*, dan *support* untuk setiap kelas. Laporan tersebut dapat dilihat pada gambar berikut.

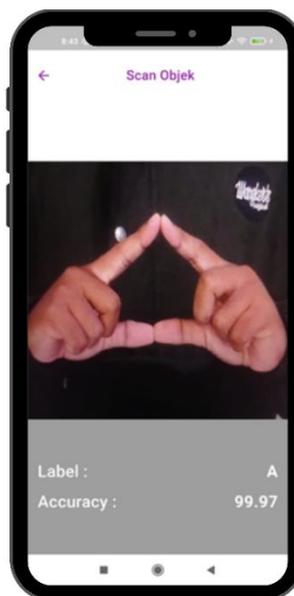
Classification Report:				
	precision	recall	f1-score	support
A	0.9524	1.0000	0.9756	20
B	1.0000	1.0000	1.0000	20
C	1.0000	0.8421	0.9143	19
D	0.7778	0.7000	0.7368	20
E	0.9500	0.9500	0.9500	20
F	1.0000	0.9500	0.9744	20
G	0.9524	1.0000	0.9756	20
H	0.9500	1.0000	0.9744	19
I	1.0000	1.0000	1.0000	20
J	0.9545	1.0000	0.9767	21
K	1.0000	1.0000	1.0000	21
L	0.8261	0.9500	0.8837	20
M	0.8947	0.8500	0.8718	20
N	0.8421	0.8889	0.8649	18
O	0.9444	0.8947	0.9189	19
P	0.7727	0.8500	0.8095	20
Q	1.0000	1.0000	1.0000	20
R	0.9444	0.8500	0.8947	20
S	0.9130	1.0000	0.9545	21
T	1.0000	1.0000	1.0000	20
U	0.9474	0.9000	0.9231	20
V	0.8636	0.9500	0.9048	20
W	1.0000	0.9524	0.9756	21
X	0.9524	1.0000	0.9756	20
Y	1.0000	0.9000	0.9474	20
Z	1.0000	0.9524	0.9756	21
accuracy			0.9385	520
macro avg	0.9399	0.9377	0.9376	520
weighted avg	0.9405	0.9385	0.9383	520

Gambar 5 Hasil Laporan *precision*, *recall*, dan *f1-score*

Dari laporan tersebut pada gambar 5, model mencapai akurasi total 93,85%, dengan rata-rata *precision*, *recall*, dan *f1-score* masing-masing sebesar 93,99%, 93,77%, dan 93,76%. Beberapa kelas memiliki nilai *f1-score* yang lebih rendah, menunjukkan perlunya perbaikan pada data atau metode pelatihan untuk meningkatkan performa model di kelas tertentu. Secara keseluruhan, hasil ini mengindikasikan bahwa model memiliki performa yang baik dalam mengklasifikasikan alfabet BISINDO..

3.3 Implementasi Model pada Perangkat Android

Model *CNN* yang telah dilatih kemudian diubah menjadi format *TensorFlow Lite* (*TFLite*) untuk dapat diimplementasikan pada aplikasi Android. Untuk meload model dalam aplikasi, digunakan library *flutter_flite*, yang menyediakan fungsi untuk membaca model *TFLite* beserta label yang digunakan untuk klasifikasi. Berikut adalah hasil tangkapan layar aplikasi, yang terdiri dari dua halaman yaitu halaman Home dan halaman Scan.



Gambar 6 Tampilan Halaman Home Gambar 7 Tampilan Halaman Scan

Pada Gambar 6 Tampilan Halaman Home, aplikasi memberikan penjelasan singkat tentang fungsinya, termasuk kemampuan mengenali bahasa isyarat Indonesia (BISINDO), dengan antarmuka sederhana dan ramah pengguna. Pada Gambar 7 Tampilan Halaman Scan menyediakan fitur utama, yaitu pengenalan bahasa isyarat secara *real-time* melalui kamera perangkat. Gestur yang dilakukan pengguna diproses oleh model untuk menampilkan label prediksi dan akurasi secara *real-time*.

Tabel 3 Hasil Pengujian pada Perangkat Android

Huruf Alfabet	Pengujian 1	Pengujian 2
A	99,88 %	99,97 %
B	98,87 %	99,34 %
C	98,73 %	93,99 %
D	70,44 %	90,20 %
E	99,98 %	99,55 %
F	97,99 %	97,91 %
G	99,98 %	99,99 %
H	99,92 %	91,47 %
I	93,05 %	99,84 %
J	99,18 %	99,98 %
K	99,83 %	99,66 %
L	99,96 %	99,98 %
M	99,86 %	99,51 %
N	98,64 %	94,16 %
O	98,96 %	100 %
P	98,65 %	98,53 %
Q	90,35 %	92,78 %
R	82,14 %	81,16 %
S	93,68 %	99,86 %
T	84,06 %	99,90 %
U	99,99 %	100 %
V	90,03 %	99,99 %
W	99,06 %	99,88 %

X	100 %	99,90 %
Y	99,81 %	99,49 %
Z	99,98 %	100 %

Tabel 3 menunjukkan hasil pengujian model pada perangkat Android untuk setiap huruf alfabet. Sebagian besar huruf mencapai akurasi tinggi di atas 99%, seperti A, G, L, dan Z. Namun, beberapa huruf seperti D, R, dan T memiliki akurasi lebih rendah, dengan nilai terendah 70,44% untuk huruf D pada Pengujian 1. Secara keseluruhan, model menunjukkan performa yang baik dan konsisten. Teknologi yang digunakan, yaitu MobileNetV2 dengan pendekatan transfer learning dan implementasi real-time menggunakan TensorFlow Lite, telah terbukti efektif dalam memberikan solusi untuk mengklasifikasikan alfabet BISINDO secara real-time pada perangkat Android.

4. KESIMPULAN

Penelitian ini menyimpulkan bahwa sistem klasifikasi alfabet Bahasa Isyarat Indonesia (BISINDO) menggunakan *Convolutional Neural Network (CNN)* dengan arsitektur *MobileNetV2* mampu menghasilkan model dengan performa yang sangat baik. Model ini berhasil mengenali 26 kelas alfabet BISINDO dengan akurasi total sebesar 93,85%, serta rata-rata *precision*, *recall*, dan *F1-score* masing-masing mencapai 93,99%, 93,77%, dan 93,76%. Implementasi model pada perangkat Android dapat membantu pengguna berinteraksi langsung melalui kamera perangkat, di mana aplikasi dapat mengenali gestur alfabet BISINDO secara real-time dengan tingkat akurasi yang tinggi.

5. SARAN

Untuk penelitian lebih lanjut, disarankan untuk memperluas dan memperkaya dataset dengan variasi gambar yang lebih beragam, termasuk gambar dengan berbagai kondisi pencahayaan, latar belakang, dan sudut pandang. Hal ini akan membantu model untuk lebih robust dalam mengenali gestur BISINDO di berbagai situasi. Selain itu, Membandingkan dengan arsitektur model *CNN* lain seperti *ResNet* atau *EfficientNet* dapat dilakukan untuk membandingkan performa dan efisiensi model.

UCAPAN TERIMA KASIH

Penulis mengucapkan rasa terima kasih kepada seluruh pihak yang telah memberikan bantuan dan dukungan selama proses penyelesaian penelitian ini.

DAFTAR PUSTAKA

- [1] A. Kinanti, M. Aly Afandi, I. Permatasari, and N. Yonara Tarigan, “Deteksi Objek Bahasa Isyarat Alfabet BISINDO Menggunakan Deep learning dan Arsitektur YOLO Bisindo Alphabet Sign Language Object Detection Using Deep Learning and YOLO Architecture.”
- [2] A. M. Ambarak and A. Z. Falani, “PENGEMBANGAN APLIKASI BAHASA ISYARAT INDONESIA BERBASIS REALTIME VIDEO MENGGUNAKAN MODEL MACHINE LEARNING,” *JIKA (Jurnal Informatika)*, vol. 7, no. 1, p. 89, Feb. 2023, doi: 10.31000/jika.v7i1.7277.
- [3] R. Fatmawati, R. Asmara, Y. R. Prayogi, and R. Y. Hakkun, “Aplikasi Pembelajaran Sistem Isyarat Bahasa Indonesia (SIBI) Berbasis Voice Menggunakan OpenSIBI,” *Technomedia Journal*, vol. 7, no. 1, pp. 22–39, Feb. 2022, doi: 10.33050/tmj.v7i1.1690.
- [4] D. Permana and J. Sutopo, “APLIKASI PENGENALAN ABJAD SISTEM ISYARAT BAHASA INDONESIA (SIBI) DENGAN ALGORITMA YOLOv5 MOBILE APPLICATION ALPHABET RECOGNITION OF INDONESIAN LANGUAGE SIGN SYSTEM (SIBI) USING YOLOv5 ALGORITHM,” vol. 11, no. 2, 2023.
- [5] I. Inayatul Arifah, F. Nur Fajri, and G. Qorik Oktagalu Pratamasunu, “Deteksi Tangan Otomatis Pada Video Percakapan Bahasa Isyarat Indonesia Menggunakan Metode YOLO Dan CNN,” 2022. [Online]. Available: <http://jurnal.polibatam.ac.id/index.php/JAIC>
- [6] L. N. Hayati, A. N. Handayani, W. S. G. Irianto, R. A. Asmara, D. Indra, and M. Fahmi, “Classifying BISINDO Alphabet using TensorFlow Object Detection API,” *ILKOM Jurnal Ilmiah*, vol. 15, no. 2, pp. 358–364, Aug. 2023, doi: 10.33096/ilkom.v15i2.1692.358-364.
- [7] M. F. Naufal and S. F. Kusuma, “Analisis Perbandingan Algoritma Machine Learning dan Deep Learning untuk Klasifikasi Citra Sistem Isyarat Bahasa Indonesia (SIBI),” *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 10, no. 4, pp. 873–882, Aug. 2023, doi: 10.25126/jtiik.20241046823.
- [8] A. Hanjarwati, J. Suprihatiningrum, and S. Aminah, “Survei Aksesibilitas Fisik dan Sosial sebagai Dasar Penyusunan Grand Design Desa Inklusif,” *JPPM (Jurnal Pendidikan dan Pemberdayaan Masyarakat)*, vol. 8, no. 1, pp. 1–17, Apr. 2021, doi: 10.21831/jppm.v8i1.24266.
- [9] I. G. A. O. Aryananda and F. Samopa, “Comparison of the Accuracy of The Bahasa Isyarat Indonesia (BISINDO) Detection System Using CNN and RNN Algorithm for Implementation on Android,” *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 4, no. 3, pp. 1111–1119, Jul. 2024, doi: 10.57152/malcom.v4i3.1465.
- [10] R. Sutjiadi, “Android-Based Application for Real-Time Indonesian Sign Language Recognition Using Convolutional Neural Network,” *TEM Journal*, pp. 1541–1549, Aug. 2023, doi: 10.18421/TEM123-35.
- [11] J. Pratama, S. Nurmaini, and M. Fachrurrozi, “Deteksi Struktur Jantung ... 635.”
- [12] O. Dwi Nurhayati, D. Eridani, and M. Hafiz Tsalavin, “SISTEM ISYARAT BAHASA INDONESIA (SIBI) METODE CONVOLUTIONAL NEURAL NETWORK SEQUENTIAL SECARA REAL TIME A REAL-TIME INDONESIAN LANGUAGE SIGN SYSTEM USING THE CONVOLUTION NEURAL NETWORK METHOD”, doi: 10.25126/jtiik.202294787.
- [13] S. Nur, A. N. Assyifa, and H. Nurjannah, “PENGEMBANGAN APLIKASI PENERJEMAH BAHASA ISYARAT INDONESIA (BISINDO) MENGGUNAKAN METODE LONG-SHORT TERM MEMORY,” *EDUSAINTEK: Jurnal Pendidikan, Sains dan Teknologi*, vol. 11, no. 1, pp. 13–30, Jul. 2023, doi: 10.47668/edusaintek.v11i1.898.

- [14] A. Arrozaq and Latipah, "Implementasi Computer Vision untuk Terjemahkan Abjad Bahasa Isyarat SIBI pada Aplikasi Android," *Jurnal Teknologi Dan Sistem Informasi Bisnis*, vol. 6, no. 2, Apr. 2024, doi: 10.47233/jteksis.v6i2.1261.
- [15] Z. Fadhilah and N. L. Marpaung, "Pengenalan Alfabet SIBI Menggunakan Convolutional Neural Network sebagai Media Pembelajaran Bagi Masyarakat Umum," vol. 8, no. 2, 2023.
- [16] L. Arisandi Barka Satya, "Sistem Klarifikasi Bahasa Isyarat Indonesia (Bisindo) Dengan Menggunakan Algoritma Convolutional Neural Network," 2022.
- [17] M. Murinto, M. Rosyda, and M. Melany, "Klasifikasi Jenis Biji Kopi Menggunakan Convolutional Neural Network dan Transfer Learning pada Model VGG16 dan MobileNetV2," *JRST (Jurnal Riset Sains dan Teknologi)*, vol. 7, no. 2, p. 183, Sep. 2023, doi: 10.30595/jrst.v7i2.16788.
- [18] S. Arnandito and T. B. Sasongko, "Comparison of EfficientNetB7 and MobileNetV2 in Herbal Plant Species Classification Using Convolutional Neural Networks," *Journal of Applied Informatics and Computing*, vol. 8, no. 1, pp. 176–185, Jul. 2024, doi: 10.30871/jaic.v8i1.7927.
- [19] M. F. Supriadi, E. Rachmawati, and A. Arifianto, "Pembangunan Aplikasi Mobile Pengenalan Objek Untuk Pendidikan Anak Usia Dini," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 8, no. 2, pp. 357–364, Mar. 2021, doi: 10.25126/jtiik.2021824363.
- [20] Samuel Ady Sanjaya, "BISINDO Indonesian Sign Language: Alphabet Image Data," Mendeley Data.
- [21] M. Zahir and R. Adi Saputra, "DETEKSI PENYAKIT RETINOPATI DIABETES MENGGUNAKAN CITRA MATA DENGAN IMPLEMENTASI DEEP LEARNING CNN," 2024. [Online]. Available: <https://www.kaggle.com/datasets/gunavenkatdoddi/eye-diseases-classification>