



Klasifikasi Gerakan Bahasa Isyarat Indonesia (Bisindo) menggunakan Arsitektur *Transfer Learning Xception*

Meisya Vira Amelia ^{a,1}, Wahyu Syaifullah Jauharis Saputra ^{a,2,*}, Kartika Maulida Hindrayani ^{a,3}

^a Universitas Pembangunan Nasional "Veteran" Jawa Timur, Jl. Rungkut Madya, Gn. Anyar, Surabaya, Jawa Timur, Indonesia, 60294

¹ 21083010018@student.upnjatim.ac.id; ² wahyu.s.j.saputra.if@upnjatim.ac.id*;

³ kartika.maulida.ds@upnjatim.ac.id

INFO ARTIKEL

Histori Artikel

Pengajuan 2025-05-27

Diperbaiki 2025-06-25

Diterima 2025-08-05

Kata Kunci

Bahasa Isyarat,
BISINDO,
Convolutional Neural
Network,
Transfer Learning,
Xception

ABSTRAK

Komunikasi antar manusia pada umumnya dilakukan dengan mengandalkan suara. Namun, hal ini tidak berlaku bagi teman tuli yang bergantung pada bahasa isyarat dalam komunikasi sehari-hari. Ketika dalam lingkup Perguruan Tinggi tidak memiliki cukup orang yang memahami bahasa isyarat, hal tersebut akan memberi ketimpangan dalam proses ajar bagi mahasiswa Tuli. Kekurangan tersebut dapat diatasi dengan meningkatkan lingkungan yang lebih inklusif, salah satu caranya adalah dengan menerapkan sistem penerjemah bahasa isyarat. Maka dari itu, penelitian ini bertujuan untuk membentuk model *machine learning* yang mampu mengklasifikasikan gerakan abjad Bahasa Isyarat Indonesia. Dalam penelitian ini, dibuat sebuah model *machine learning* untuk mendeteksi gerakan Bahasa Isyarat Indonesia dengan memanfaatkan metode *transfer learning Xception* dari Convolutional Neural Network. Dataset terdiri dari 26 abjad Bahasa Isyarat Indonesia dengan total data sebanyak 650 gambar. Evaluasi model yang dilakukan melalui K-Folds cross validation mendapatkan F1-score 98% pada fase pengujian.

ABSTRACT

Keyword

BISINDO,
Convolutional Neural
Network,
Sign Language,
Transfer Learning,
Xception

Human communication generally relied on speech. However, this was not applicable to the deaf people, who depended on sign language for daily interactions. Unfortunately, not everyone had the ability to understand sign language. In higher education environments, the lack of individuals proficient in sign language often created inequality in the learning process for deaf students. This limitation could be addressed by fostering a more inclusive environment, one of which was through the implementation of a sign language translation system. Therefore, this study aimed to develop a machine learning model capable of detecting and translating Indonesian Sign Language (BISINDO) alphabet gestures. The model was built using the Xception transfer learning method from Convolutional Neural Networks (CNN). The dataset consisted of 26 BISINDO alphabet gestures with a total of 650 images. The model was evaluated using K-Fold cross-validation and achieved an F1-score of 98% during testing.

Ini adalah artikel akses terbuka di bawah lisensi [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/).



1. Pendahuluan

Kehidupan sehari-hari manusia tidak dapat terlepas dari kegiatan berkomunikasi dengan manusia lainnya. Hal ini dikarenakan komunikasi adalah kegiatan yang dilakukan oleh manusia untuk menerima dan menyampaikan pesan atau berita dengan tujuan pesan yang ingin disampaikan dapat dipahami [1]. Umumnya, manusia sehari-hari mengandalkan suara mereka untuk berbincang dengan orang lain. Namun, hal ini tidak berlaku bagi para penyandang tunarungu dan tunawicara.

Berdasarkan data yang dimiliki oleh Badan Pusat Statistik Nasional pada tahun 2022, terdapat sebanyak 4.956.814 penduduk Indonesia yang memiliki masalah kesulitan untuk mendengar dan 3.234.964 penduduk yang mengalami kesulitan berbicara [2], [3]. Menurut laman Layanan Mahasiswa Disabilitas yang dikelola oleh Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi hanya terdapat 27 dari 2.937 Perguruan Tinggi yang menerima mahasiswa dengan disabilitas rungu [4]. Jika dibandingkan dengan data jumlah Teman Tuli, tentunya angka tersebut masih sangat sedikit. Hal ini menunjukkan bahwa masih sangat sedikit perguruan tinggi yang menyediakan lingkungan yang inklusif, khususnya bagi mahasiswa tuna rungu.

Menciptakan lingkungan yang inklusif di lingkungan Perguruan Tinggi dapat dilakukan dengan berbagai cara. Salah satunya adalah dengan menyediakan penerjemah bahasa isyarat. Partisipasi aktif mahasiswa dalam kegiatan akademik dapat ditingkatkan dengan bantuan penerjemah [5]. Hal ini merupakan satu langkah maju dalam menciptakan lingkungan yang lebih inklusif di lingkungan perguruan tinggi.

Kemajuan teknologi dalam komputasi telah menyederhanakan banyak aspek kehidupan, termasuk pengembangan penerjemah bahasa isyarat [6], [7]. Penerjemah bahasa isyarat dapat dibuat dengan menggunakan *machine learning*. *Machine learning* adalah cabang dari ilmu kecerdasan buatan yang memberi komputer kemampuan untuk belajar dari data [8]. Model *machine learning* dibangun menggunakan algoritma. Salah satu algoritma yang sering digunakan adalah *Convolutional Neural Network* (CNN) [9]. CNN sering digunakan untuk memproses data gambar karena akurasi yang baik dalam mengklasifikasikan gambar [10]. Selain itu, CNN memiliki banyak model *pre-trained* yang dapat mengefisienkan proses pembuatan model baru.

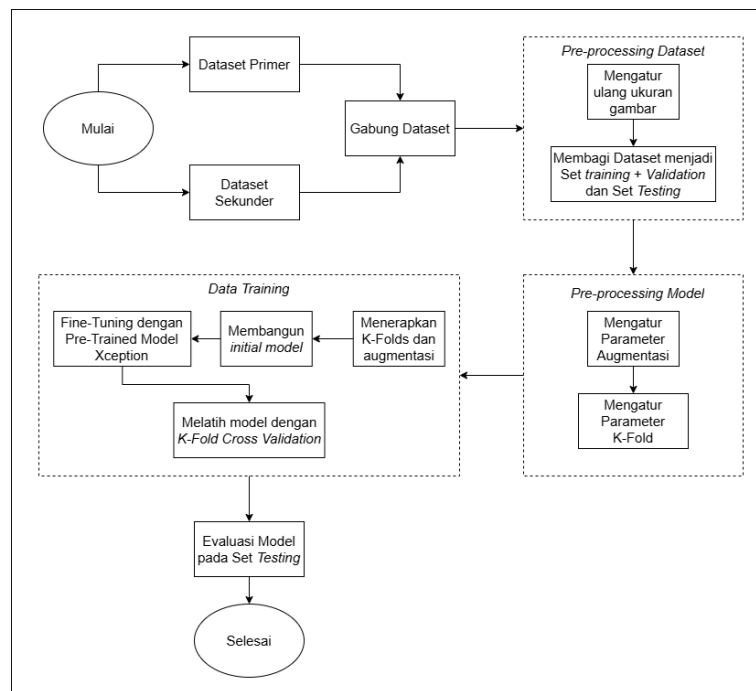
Penelitian terdahulu seperti penelitian Rachmawati et al [11] menggunakan 4 model *pre-trained* dari CNN, yaitu *ResNet50*, *EfficientNetB1*, *MobileNetV4*, dan *InceptionV3*. Data yang digunakan adalah gambar gerakan abjad BISINDO (Bahasa Isyarat Indonesia) yang berjumlah 1.727 gambar. *ResNet50* dan *MobileNetV4* menjadi model terbaik yang berhasil diperoleh dengan nilai akurasi pada fase pelatihan sebesar 99% dan 98% serta nilai *loss* sebesar 0,03 dan 0,06. Nilai tersebut lebih stabil dan unggul dibandingkan dengan *EfficientNetB1* dan *InceptionV3* yang memiliki nilai akurasi sebesar 8% dan 96% dengan nilai *loss* sebesar 3,40 dan 0,12. Penelitian lain yang dilakukan oleh Sari et al [12] menggunakan *Random Forest* sebagai model klasifikasi dan *InceptionV3* untuk proses *feature extraction*. Data yang digunakan adalah gambar gerakan abjad SIBI (Sistem Isyarat Bahasa Indonesia) dengan jumlah gambar sebanyak 5280 gambar. Model tersebut dievaluasi menggunakan *K-Fold Cross Validation* dan memperoleh nilai akurasi sebesar 85,40% dan *F1-score* sebesar 85,20%. Kedua penelitian tersebut menggunakan *InceptionV3* dalam membuat modelnya. Akan tetapi, kinerjanya masih kurang unggul dibandingkan dengan model *pre-trained* lainnya.

Francois Chollet menciptakan arsitektur baru dengan menggunakan basis hipotesis yang lebih kuat daripada hipotesis yang mendasari terciptanya arsitektur *Inception* [13]. Hipotesis yang digunakan adalah pemetaan korelasi *cross-channel* dan korelasi *spatial* pada peta fitur *Convolutional Neural Network* dapat sepenuhnya dipisahkan. Arsitektur ini kemudian disebut

Extreme Inception atau *Xception*. *Xception* mengungguli *InceptionV3* pada beberapa tugas klasifikasi, termasuk *ImageNet* dan dataset internal yang lebih besar dengan 17.000 kelas [10].

Berdasarkan hal tersebut, penelitian ini bertujuan untuk memperoleh performa terbaik dalam mengklasifikasikan gerakan abjad BISINDO dengan memanfaatkan metode *transfer learning* menggunakan arsitektur *Xception*. Dataset yang digunakan sama dengan dataset pada penelitian terdahulu [11] dengan menambahkan beberapa dataset yang diperoleh secara manual. Model yang ditampilkan merupakan model final. Penelitian ini menjelaskan tahapan yang dilakukan serta alasan pemilihan parameter yang digunakan.

2. Metode



Gambar 1. Alur Kerja Metodologi Penelitian

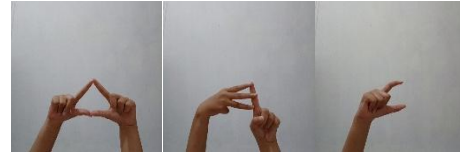
Metode yang digunakan dalam penelitian ini secara garis besar adalah pengumpulan dataset, *pre-processing* dataset, *pre-processing* model, pelatihan model, dan evaluasi model dengan set pengujian. Gambar 1 mengilustrasikan metode penelitian yang digunakan.

2.1. Mengumpulkan Dataset

Terdapat dua sumber data yang digunakan dalam penelitian ini. Data sekunder yang diambil dari situs Kaggle milik Achmad Noer [14]. Dataset ini terdiri dari abjad BISINDO dengan total gambar sebanyak 312 yang masing-masing kelasnya terdapat 12 gambar. Gambar memiliki 3 jenis latar belakang, yaitu kemeja putih polos, dinding putih, dan kemeja putih berbintik-bintik. Contoh gambar data sekunder ditampilkan pada Gambar 2. Kemudian data kedua, yaitu data primer. Data ini diambil secara manual dengan merekam gerakan abjad BISINDO, lalu dipotong setiap *frame* dan diseleksi gambar yang akan digunakan. Total gambar yang digunakan sebanyak 338 gambar dan setiap kelasnya terdapat 13 gambar. Contoh gambar data primer ditampilkan pada gambar 3.



Gambar 2. Gambar Data Sekunder



Gambar 3. Gambar Data Primer

Kedua dataset ini kemudian digabungkan menjadi satu, sehingga menghasilkan total 650 gambar dengan masing-masing kelas memiliki 25 gambar. Berdasarkan hal ini, dataset dikategorikan ke dalam klasifikasi multikelas. Klasifikasi multikelas mirip dengan klasifikasi biner, hanya saja kelas targetnya adalah non-biner [15].

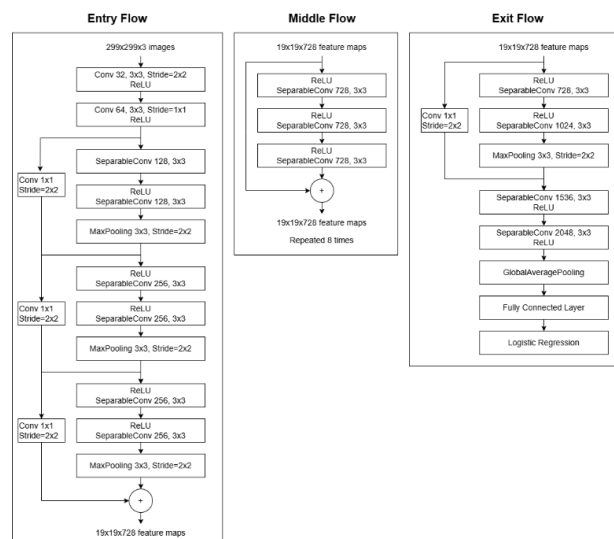
2.2. Pre-Processing Dataset

Dataset yang telah digabung selanjutnya diproses menjadi dalam bentuk *array* dengan ukuran 299×299. Nilai piksel dinormalisasi menggunakan fungsi *preprocess_input*, yang menskalakannya ke rentang [-1, 1] agar sesuai dengan format input model *pre-trained Xception*. Kemudian, label data ditransformasi menjadi *one-hot encoding*. Data akan dibagi dengan rasio 90:10, 90% dari data digunakan sebagai set pelatihan dan validasi, 10% sisanya digunakan sebagai set pengujian. Set pelatihan dan validasi tidak dipisah di awal, karena akan dibagi pada saat proses K-Fold. Sedangkan set pengujian dipisah terlebih dahulu untuk untuk menghindari terjadinya kebocoran data. Jumlah data yang diperoleh untuk set pelatihan dan validasi adalah 585 gambar dan set pengujian adalah 65 gambar.

2.3. Pre-Processing Model

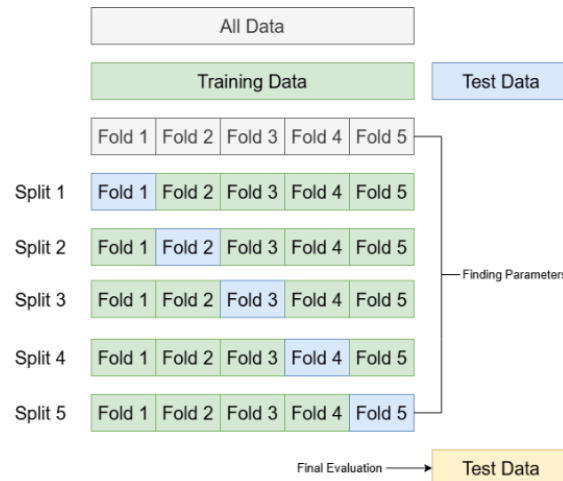
Parameter data augmentasi diletakkan sebelum model dilatih, namun augmentasi diproses secara *real-time* bersamaan dengan pelatihan model. Parameter data augmentasi yang digunakan adalah rentang rotasi, rentang pergeseran lebar, rentang pergeseran tinggi, rentang *zoom*, dan rentang kecerahan. Augmentasi ini dilakukan untuk menambah variasi data yang ada. Kemudian, nilai *fold* untuk proses *K-Fold* ditentukan menjadi 5. *Stratified K-Fold* diterapkan untuk membagi rata jumlah data pada masing-masing kelas di setiap *fold*. Proses pembagian data ini dilakukan sebelum pelatihan model. Nilai *epoch* yang diterapkan berjumlah 100 dengan 50 *epochs* untuk pelatihan *initial model* dan 50 *epochs* untuk pelatihan *fine-tuning*. Ukuran *batch* ditentukan menjadi 32.

2.4. Pelatihan Model



Gambar 4. Arsitektur Xception

Pada fase pelatihan model sesuai flowchart pada gambar 4, arsitektur *transfer learning* yang digunakan untuk pelatihan model adalah *Xception* dengan menggunakan *database ImageNet*. *Xception* merupakan salah satu arsitektur *deep learning* dan merupakan penyempurnaan dari arsitektur *Inception*. *Xception* dikembangkan berdasarkan hipotesis bahwa pemetaan korelasi *cross-channel* dan korelasi *spatial* pada peta fitur *Convolutional Neural Network* dapat sepenuhnya dipisahkan. Hipotesis ini merupakan versi yang lebih kuat dibandingkan hipotesis yang mendasari terciptanya arsitektur *Inception*, oleh karena itu mereka menamakan arsitektur yang tercipta dari hipotesis ini sebagai *Extreme Inception* atau *Xception* [13].



Gambar 5. *Workflow K-Fold Cross Validation*

Tabel 1 merupakan data pelatihan model, dengan menggunakan metode K-Fold Cross Validation. K-Fold Cross Validation dipilih sebagai teknik evaluasi dalam penelitian ini karena jumlah dataset yang sedikit dan untuk mengurangi risiko bias pada data selama pelatihan model [16]. K-Fold Cross Validation bekerja dengan memecah data dalam set pelatihan dan validasi menjadi 'k' bagian dari set data dengan jumlah data yang sama dibagi. Alur kerja K-Fold Cross Validation ditunjukkan pada Gambar 5.

Tabel 1. Parameter Pelatihan Model

Parameter	Detail
Arsitektur Transfer Learning	<i>Xception</i>
Jumlah Pelatihan Layer	4 layers terakhir sebelum <i>initial model</i> 70 layers ke atas setelah <i>initial model</i>
Optimizer	Adam
Loss Function	<i>Categorical Cross Entropy</i>
Epoch	100

Optimizer yang digunakan untuk mengurangi nilai *loss* pada model adalah Adam optimizer. *Learning rate* yang diterapkan pada *initial model* adalah 0,0001 (1e-4) dan saat *fine-tuning* 0,00001 (1e-5). *Categorical Cross Entropy* diterapkan sebagai *loss function*, karena sebelumnya model telah ditransformasi menjadi *one-hot encoding*.

Proses pelatihan model dimulai dari menjalankan proses augmentasi data pada set pelatihan. Kemudian melatih 4 layers terakhir dari model *pre-trained* dan dilanjutkan dengan *initial model* sebanyak 50 epochs. Setelah selesai pelatihan pada *initial model*, dilanjutkan dengan *fine-tune* yang melatih 70 layers ke atas pada model *pre-trained Xception* sebanyak 50 epochs. Model dievaluasi dan model *folders* terbaik akan digunakan sebagai model akhir. Model akhir dievaluasi dengan set pengujian, kemudian dicek performa model dengan *confusion matrix*.

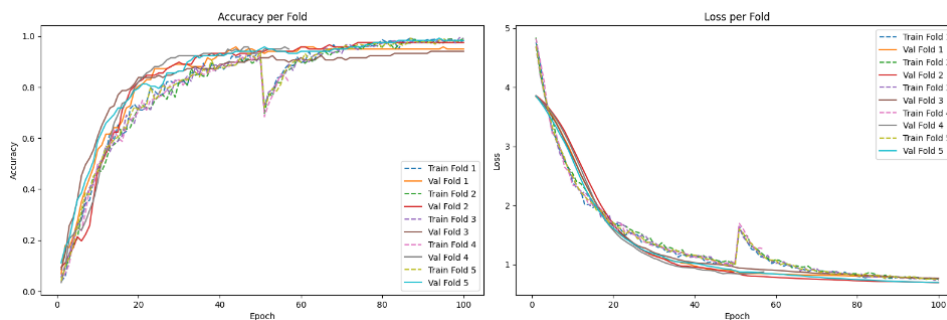
3. Hasil dan Analisis

Pada bagian ini akan dibahas terkait hasil yang didapatkan dari pelatihan model dan evaluasinya pada data uji. Hasil pelatihan model ditampilkan pada Tabel 2.

Tabel 2. Komparasi Performa Model tiap Fold

Fold	Akurasi Pelatihan	Akurasi Validasi	Loss Pelatihan	Loss Validasi	f1-score Validasi
1	99,25%	94,87%	0,7188	0,7604	95,11%
2	99,57%	97,44%	0,7442	0,6935	97,22%
3	98,64%	94,02%	0,7426	0,7629	93,38%
4	81,16%	94,87%	1,2579	0,8753	94,35%
5	98,18%	98,29%	0,7411	0,6944	98,45%

Pelatihan model dengan *K-Fold Cross Validation* dilakukan secara berulang (*looping*) sesuai dengan nilai k, yaitu 5 *folds*. Sehingga, setiap satu kali pelatihan model akan menghasilkan 5 model, seperti pada hasil Tabel 2. Penelitian ini menetapkan tiga metrik untuk mengevaluasi model yang akan digunakan, yaitu akurasi validasi, *loss* validasi, dan *f1-score* validasi. Berdasarkan performa dari kelima model, model dari *fold* 5 menampilkan hasil yang cukup stabil. Hal ini dapat dilihat dari nilai selisih akurasi pelatihan dan validasi yang tidak terlalu besar, artinya generalisasi model dalam memprediksi gambar sudah bagus. Begitu juga dengan nilai pada *loss* pelatihan dan validasi. Selain itu, secara keseluruhan *f1-score* milik model *fold* 5 adalah yang tertinggi dengan nilai 98,45%. Model pada *fold* 2 juga memiliki performa yang hampir sama bagusnya dengan *fold* 5. Namun, selisih nilai akurasi pada pelatihan dan validasinya lebih tinggi dibandingkan dengan model pada *fold* 5.



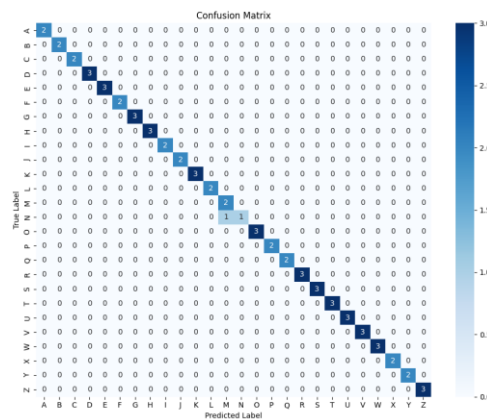
Gambar 6. Grafik Akurasi dan *Loss Model*

Ketika model memasuki fase *fine-tuning*, terdapat lonjakan negatif pada performa model. Lonjakan tersebut terjadi *epoch* 51 hingga 60, seperti yang ditampilkan pada Gambar 6. Lonjakan ini disebabkan oleh jumlah *layer* yang digunakan pada *fine-tuning* terlalu banyak, pada model ini yaitu 70 *layers* terakhir dari model *pre-trained Xception*. Seluruh *fold* model mencapai *epoch* 100, kecuali *fold* 4 yang berhenti pada *epoch* 57, karena mengalami *early stopping*.

Tabel 2. Evaluasi Performa pada Set Pengujian

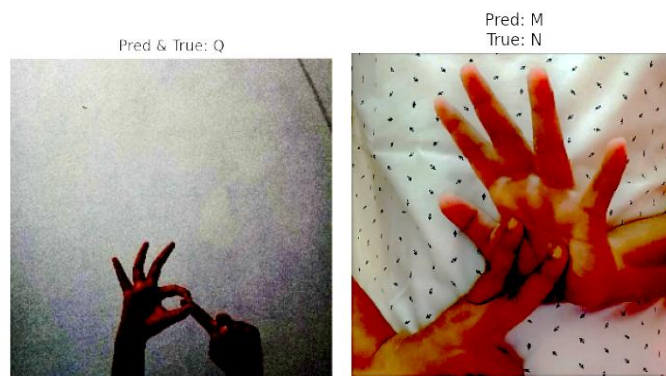
Keterangan	Nilai
Akurasi	98%
F1-score	98%
Presisi	99%
Recall	98%
Loss	0,0726

Performa model pada set pengujian menunjukkan hasil yang memuaskan dengan nilai akurasi sebesar 98%, *F1-score* 98%, presisi 99%, *recall* 98%, dan *loss* 0,0726.



Gambar 7. Grafik Confusion Matrix Model

Berdasarkan hasil *confusion matrix* pada gambar 7, dapat diketahui bahwa sampel pengujian setiap kelas memiliki 2-3 gambar. Model berhasil memprediksi gambar dengan benar sebagian besarnya. Namun, terdapat kesalahan klasifikasi pada abjad N yang diprediksi jadi M.



Gambar 8. Gambar Prediksi Benar dan Salah

Gambar 8 menampilkan prediksi yang benar dan salah. Kesalahan prediksi pada abjad N dapat dipengaruhi oleh gerakannya yang menyerupai dengan abjad M. dikarenakan, jumlah set pengujian cukup sedikit sehingga tidak bisa dikonfirmasi apakah model memang kesulitan dalam memprediksi abjad N atau ada faktor lain.

4. Simpulan

Berdasarkan hasil pengujian dan analisa yang telah dilakukan, berhasil diperoleh model terbaik yang bisa didapatkan pada penelitian ini untuk mengklasifikasikan gambar abjad BISINDO. Model dibangun dengan menerapkan metode *transfer learning Xception*, penerapannya adalah pelatihan 4 *layers* terakhir di awal model dan 70 *layers* terakhir di akhir model (*fase fine-tuning*). Model berhasil mendapatkan nilai akurasi sebesar 98%, *F1-score* 98%, presisi 99%, *recall* 98%, dan *loss* 0,0726 pada set pengujian, hal ini menunjukkan model memiliki performa yang sangat bagus.

Namun, terdapat beberapa hal yang mampu ditingkatkan dari penelitian ini. Pertama, menambahkan *dataset* karena *dataset* yang digunakan pada penelitian ini relatif sedikit sehingga sangat sensitif untuk terjadi *overfitting* ketika menggunakan *transfer learning* dalam pembuatan modelnya. Kedua, meminimalisir lonjakan negatif model ketika melakukan *fine-tuning* dengan mengurangi jumlah *layer* yang dilatih.

Referensi

- [1] T. M. Milyane and Dkk, *Pengantar Ilmu Komunikasi*, vol. 5, no. 3. 2022. [Online]. Available: <https://repository.penerbitwidina.com/media/publications/557082-pengantar-ilmu-komunikasi-22ec77af.pdf>
- [2] BPS, "Penduduk Menurut Wilayah dan Tingkat Kesulitan Mendengar," Badan Pusat Statistik. Accessed: Sep. 13, 2024. [Online]. Available: <https://sensus.bps.go.id/topik/tabular/sp2022/145/0/0>
- [3] BPS, "Jumlah Penduduk Kesulitan Berbicara," Badan Pusat Statistik. Accessed: Sep. 13, 2024. [Online]. Available: <https://sensus.bps.go.id/topik/tabular/sp2022/146/1/0>
- [4] LMD, "Sebaran Mahasiswa Disabilitas," Layanan Mahasiswa Disabilitas. Accessed: Sep. 13, 2024. [Online]. Available: <https://ptinklusif.kemdiktisaintek.go.id/s/5/sebaran-mahasiswa-disabilitas>
- [5] Effranzudeta, Yesinovitarsari, Herdi, and Dinafitriani, "Journal of Special Education Lectura Urgensi Bahasa Isyarat di Lingkungan Universitas," vol. 2, no. 1, pp. 65–72, 2024.
- [6] W. S. J. Saputra, M. H. Fardana, and M. A. R. Valentino, "Implementasi AI Pendeteksi Pola Gerak Tangan pada Game 'Pong Ball' dengan Menggunakan Algoritma Convolutional Neural Network," *J. Manajemen Inform. Jayakarta*, vol. 2, no. 3, p. 235, 2022, doi: 10.52362/jmijayakarta.v2i3.834.
- [7] D. A. Prasetya, A. P. Sari, P. A. Riyantoko, and T. M. Fahrudin, "The Effect of Information Quality and Service Quality on User Satisfaction of the Government of Kabupaten Malang," *TIERS Inf. Technol. J.*, vol. 4, no. 1, pp. 32–42, 2023, doi: 10.38043/tiers.v4i1.4328.
- [8] T. M. Fahrudin, P. A. Riyantoko, K. M. Hindrayani, and E. M. Safitri, "An Introduction To Machine Learning Games And Its Application For Kids In Fun Project," *Int. J. Comput. Netw. Secur. Inf. Syst.*, vol. 2, no. 1, pp. 26–30, 2020, [Online]. Available: <https://machinelearningforkids.co.uk>
- [9] M. Idhom, D. A. Prasetya, P. A. Riyantoko, T. M. Fahrudin, and A. P. Sari, "Pneumonia Classification Utilizing VGG-16 Architecture and Convolutional Neural Network Algorithm for Imbalanced Datasets," *TIERS Inf. Technol. J.*, vol. 4, no. 1, pp. 73–82, 2023, doi: 10.38043/tiers.v4i1.4380.
- [10] S. Shania, M. Farid Naufal, V. Riandaru Prasetyo, and M. S. Bin Azmi, "Translator of Indonesian Sign Language Video using Convolutional Neural Network with Transfer Learning," *Indones. J. Inf. Syst.*, vol. 5, no. 1, pp. 17–27, 2022, doi: 10.24002/ijis.v5i1.5865.
- [11] I. D. A. Rachmawati, R. Yunanda, M. F. Hidayat, and P. Wicaksono, "Deep Transfer Learning for Sign Language Image Classification: A Bisindo Dataset Study," *Eng. Math. Comput. Sci. J.*, vol. 5, no. 3, pp. 175–180, 2023, doi: 10.21512/emacsjournal.v5i3.10621.
- [12] M. Sari and E. R. Jamzuri, "Hand Sign Recognition of Indonesian Sign Language System (SIBI) Using," vol. 5, no. 158, pp. 258–265, 2026.
- [13] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," *Proc. IEEE Conf. Comput. Vis. pattern Recognit.*, vol. 7, no. 3, pp. 1251–1258, 2017, doi: <https://doi.org/10.48550/arXiv.1610.02357>.
- [14] A. Noer, "Bahasa Isyarat Indonesia (BISINDO) Alphabets." [Online]. Available: <https://www.kaggle.com/datasets/achmadnoer/alfabet-bisindo>
- [15] A. Muhaimin, W. Wibowo, and P. A. Riyantoko, "Multi-label Classification Using Vector Generalized Additive Model via Cross-Validation," *J. Inf. Commun. Technol.*, vol. 22, no. 4, pp.

657–673, 2023, doi: <https://doi.org/10.32890/jict2023.22.4.5>.

- [16] R. Tuntun, K. Kusrini, and K. Kusnawi, "Analisis Perbandingan Kinerja Algoritma Klasifikasi dengan Menggunakan Metode K-Fold Cross Validation," *J. Media Inform. Budidarma*, vol. 6, no. 4, p. 2111, 2022, doi: 10.30865/mib.v6i4.4681.