

Pengenalan Pola Huruf Hijaiyyah dengan Metode CNN untuk Bahasa Isyarat Arab

Siska Khoirunnisa^{*1}, Aris Rakhmadi²

^{1,2}Informatika, Fakultas Komunikasi dan Informatika, Universitas Muhammadiyah Surakarta,
Indonesia
Email: ¹siskakhnnisa@gmail.com, ²aris.rakhmadi@ums.ac.id

Abstrak

Bahasa Isyarat Arab (*Arabic Sign Language/ArSL*) merupakan sarana komunikasi utama bagi penyandang tunarungu, termasuk dalam pembelajaran Al-Qur'an. Namun, keterbatasan teknologi dalam mengenali bahasa isyarat secara otomatis menjadi hambatan serius terhadap akses pendidikan agama yang inklusif. Penelitian ini bertujuan untuk mengenali pola huruf hijaiyyah dalam ArSL dengan memanfaatkan metode *Convolutional Neural Network* (CNN) melalui pendekatan transfer learning dan fine-tuning pada empat arsitektur pralatih, yaitu MobileNetV2, EfficientNetB0, VGG16, dan ResNet50. Dataset yang digunakan terdiri dari 7.856 citra RGB tangan yang mewakili 31 huruf hijaiyyah, yang dibagi menjadi data pelatihan, validasi, serta pengujian. Evaluasi dilakukan menggunakan metrik *accuracy*, *precision*, *recall*, *F1-score*, serta efisiensi komputasi berdasarkan ukuran model dan waktu inferensi. Hasil penelitian memperlihatkan bahwa ResNet50 memperoleh akurasi tertinggi sebesar 98,35%, diikuti MobileNetV2 (97,84%), EfficientNetB0 (97,71%), dan VGG16 (97,07%). Meskipun demikian, MobileNetV2 memiliki ukuran model terkecil dan kecepatan inferensi tercepat, sehingga paling sesuai untuk implementasi pada perangkat dengan keterbatasan sumber daya. Analisis *confusion matrix* juga menunjukkan kesalahan klasifikasi terutama pada huruf yang memiliki kemiripan visual, seperti dal-dzal dan ta-tha. Penelitian ini menegaskan efektivitas CNN berbasis *transfer learning* dalam pengenalan huruf hijaiyyah bahasa isyarat Arab serta memberikan kontribusi nyata terhadap pengembangan sistem pembelajaran agama yang lebih inklusif bagi penyandang tunarungu.

Kata kunci: Bahasa Isyarat Arab, Convolutional Neural Network, Hijaiyyah, Transfer Learning, Tunarungu

Recognition of Hijaiyyah Characters in Arabic Sign Language Using Convolutional Neural Networks

Abstract

Arabic Sign Language (ArSL) serves as a primary communication medium for individuals with hearing impairments, particularly in Qur'anic education. However, the lack of technology capable of automatically recognizing sign language poses a major barrier to inclusive access to religious learning. This study aims to recognize hijaiyyah characters in ArSL by applying Convolutional Neural Networks (CNNs) through a transfer learning and fine-tuning approach on four pretrained architectures: MobileNetV2, EfficientNetB0, VGG16, and ResNet50. The dataset comprises 7,856 RGB images of hand gestures representing 31 hijaiyyah letters, divided into training, validation, and testing sets. The evaluation employed metrics such as accuracy, precision, recall, F1-score, as well as computational efficiency in terms of model size and inference speed. The results demonstrate that ResNet50 achieved the highest test accuracy of 98.35%, followed by MobileNetV2 (97.84%), EfficientNetB0 (97.71%), and VGG16 (97.07%). Nevertheless, MobileNetV2 showed the smallest model size and the fastest inference time, making it the most suitable option for deployment on resource-constrained devices. Confusion matrix analysis revealed that most misclassifications occurred between visually similar letters, such as dal-dzal and ta-tha. This study highlights the effectiveness of transfer learning-based CNNs in recognizing hijaiyyah characters in Arabic Sign Language and provides a significant contribution to the development of inclusive religious education systems for the deaf community.

Keywords: Arab Sign Language, Convolutional Neural Network, Hijaiyyah, Transfer Learning, Visual Impairment

1. PENDAHULUAN

Bahasa isyarat merupakan sarana komunikasi utama bagi individu tunarungu dan tunawicara dalam berinteraksi dengan lingkungan sekitarnya. Salah satu bentuk bahasa isyarat yang digunakan adalah Bahasa Isyarat Arab (*Arab Sign Language/ArSL*), yang tidak hanya berfungsi dalam komunikasi sehari-hari, tetapi juga berperan penting dalam pendidikan, termasuk dalam pembelajaran Al-Qur'an. Bagi penyandang tunarungu, memahami Al-Qur'an tidak hanya bergantung pada teks tertulis, tetapi juga membutuhkan metode visual yang mendukung pemahaman tajwid, makna, dan pelafalan. Namun, penyandang tunarungu menghadapi kesulitan dalam memahami aspek fonetik dan artikulasi Al-Qur'an secara komprehensif. Dalam konteks tersebut, bahasa isyarat menjadi metode yang efektif untuk menyampaikan informasi visual. Penelitian menunjukkan bahwa penggunaan bahasa isyarat dapat membantu siswa tunarungu dalam mengenal huruf hijaiyah pada pembelajaran Al-Qur'an [1], [2]. Akan tetapi, keterbatasan teknologi pendukung menyebabkan proses pembelajaran menjadi kurang inklusif dan membatasi akses komunikasi, yang berdampak pada rendahnya partisipasi sosial serta aksesibilitas penyandang disabilitas [3]. Di sisi lain, rendahnya pemahaman masyarakat terhadap ArSL memperburuk hambatan komunikasi [4]-[6]. Kondisi tersebut menegaskan perlunya pengembangan teknologi yang inklusif, adaptif, dan berbasis pembelajaran visual untuk mendukung komunikasi dua arah yang efektif bagi penyandang disabilitas.

Perkembangan kecerdasan buatan (*Artificial Intelligence/AI*), khususnya dalam bidang pengolahan citra digital (*Computer Vision*), membuka peluang besar untuk mengembangkan sistem penerjemahan otomatis bahasa isyarat. Salah satu pendekatan yang paling efektif adalah *Convolutional Neural Network* (CNN), yaitu arsitektur jaringan saraf dalam yang unggul dalam klasifikasi citra dan pengenalan pola visual kompleks [7], [8]. CNN mampu mengekstraksi fitur visual secara otomatis tanpa memerlukan teknik ekstraksi manual [9], sehingga sangat relevan untuk pengenalan bahasa isyarat yang membutuhkan pemrosesan pola visual secara efisien dan presisi tinggi. Meskipun teknologi *deep learning* telah menunjukkan kemajuan pesat, penerjemahan otomatis Bahasa Isyarat Arab masih menghadapi kendala utama berupa keterbatasan dataset ArSL yang representatif dan beragam, yang berdampak langsung pada kualitas pelatihan model *deep learning* [10]. Mengatasi hal tersebut, pendekatan *transfer learning* dengan teknik *fine-tuning* pada model prelatih menjadi solusi yang menjanjikan karena memungkinkan pemanfaatan pengetahuan dari domain lain (misalnya, ImageNet) untuk meningkatkan akurasi pada domain dengan data terbatas.

Sejumlah penelitian telah melaporkan keberhasilan CNN dalam pengenalan bahasa isyarat melalui penerapan *transfer learning*, *fine-tuning*, dan *data augmentation*. Studi pada Bahasa Isyarat Amerika (*American Sign Language/ASL*) menunjukkan bahwa arsitektur seperti InceptionV3, EfficientNetB0, MobileNetV2, dan ResNet50 mampu mencapai akurasi sangat tinggi, bahkan mendekati 100% [11]-[13]. Penelitian lain pada Bahasa Isyarat Arab (ArSL) juga menunjukkan hasil yang menjanjikan dengan akurasi mencapai 99,6% menggunakan ResNet50, MobileNetV2, dan EfficientNetB7 [14], [15]. Selain itu, studi pengenalan huruf hijaiyah berbasis CNN menunjukkan bahwa arsitektur seperti MobileNetV2, VGG16, dan Xception mampu mencapai akurasi tinggi dengan waktu pelatihan relatif singkat [16], sedangkan penelitian lain menyoroti potensi penerapan model CNN pada perangkat bergerak melalui TensorFlow Lite [17]. Hasil-hasil tersebut menunjukkan bahwa *transfer learning* menjadi pendekatan dominan yang efektif dalam meningkatkan performa pengenalan bahasa isyarat pada berbagai konteks.

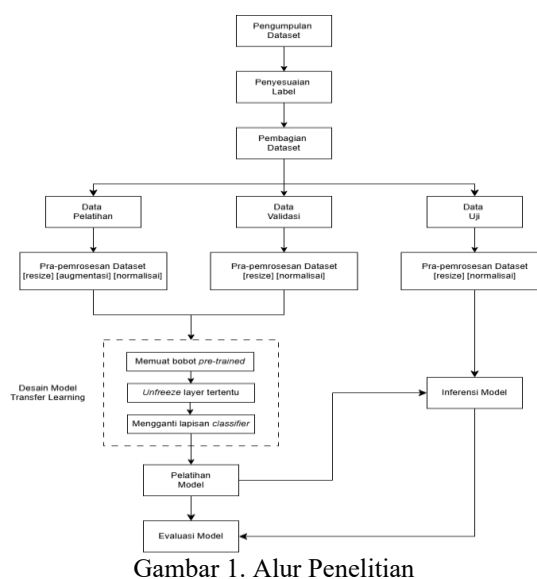
Walaupun demikian, penerapan model-model tersebut dalam kondisi dunia nyata masih menghadapi tantangan signifikan. Variasi latar belakang citra, pencahayaan, serta sudut pengambilan gambar yang tidak seragam sering kali menurunkan akurasi model dan mengurangi kemampuan generalisasi sistem. Untuk mengatasi hal ini, sejumlah penelitian telah menerapkan teknik augmentasi dan preprocessing lanjutan guna meningkatkan *robustness* terhadap kondisi lingkungan yang dinamis [18]-[21]. Penelitian lain juga menegaskan bahwa pemilihan arsitektur dan strategi *fine-tuning* yang tepat memiliki pengaruh besar terhadap performa klasifikasi pada dataset kompleks, termasuk dalam pengenalan bahasa isyarat [22], [23]. Hal ini menunjukkan pentingnya evaluasi komparatif antar-model untuk memahami keseimbangan antara akurasi, efisiensi, dan kompleksitas arsitektur.

Berdasarkan tinjauan tersebut, dapat diidentifikasi adanya celah penelitian (*research gap*) berupa terbatasnya kajian komparatif yang secara sistematis membandingkan performa berbagai arsitektur CNN prelatih dalam klasifikasi huruf hijaiyah Bahasa Isyarat Arab, terutama dalam konteks keseimbangan antara akurasi dan efisiensi komputasi. Oleh karena itu, penelitian ini bertujuan untuk menganalisis dan membandingkan performa empat arsitektur CNN prelatih, yaitu MobileNetV2, EfficientNetB0, VGG16, dan ResNet50, melalui penerapan strategi *fine-tuning*, augmentasi data, serta optimasi *hyperparameter* dalam proses klasifikasi huruf hijaiyah pada Bahasa Isyarat Arab. Hasil penelitian ini diharapkan dapat memberikan kontribusi terhadap pengembangan sistem pengenalan bahasa isyarat yang lebih efisien, akurat, dan aplikatif bagi penyandang tunarungu dalam konteks pembelajaran Al-Qur'an maupun komunikasi visual yang lebih inklusif.

2. METODE PENELITIAN

Penelitian ini menggunakan pendekatan *transfer learning* dengan teknik *fine-tuning* pada arsitektur *Convolutional Neural Network* (CNN). *Transfer learning* merupakan metode dalam *machine learning* yang memanfaatkan pengetahuan dari model yang telah dilatih pada tugas tertentu dan menerapkannya pada tugas baru dengan data terbatas, yang terbukti efektif dalam pengenalan pola citra [24]. Teknik *fine-tuning* sendiri dilakukan dengan melatih ulang sebagian lapisan dari model pra-latih agar dapat menyesuaikan dengan karakteristik dataset target [25]. Pendekatan ini sangat relevan ketika jumlah data yang tersedia relatif terbatas, seperti pada kasus klasifikasi bahasa isyarat huruf hijaiyah. Dalam konteks tersebut, memulai pelatihan model dari awal (*from scratch*) dinilai tidak efisien dan berisiko menghasilkan model dengan performa generalisasi yang rendah. Oleh karena itu, pemanfaatan model pralatih sebagai titik awal pelatihan yang kemudian disesuaikan melalui *fine-tuning* menjadi strategi yang sangat efektif [26].

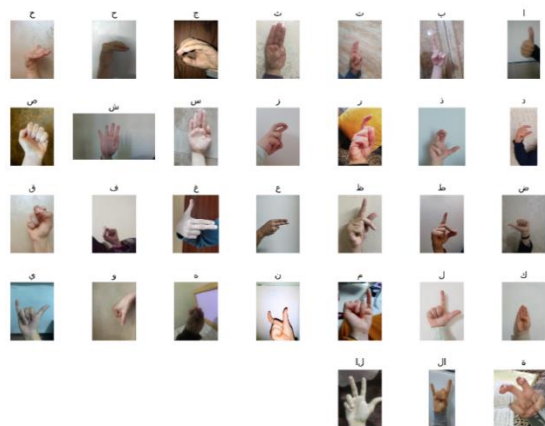
Proses penelitian ini dilaksanakan melalui serangkaian tahapan yang sistematis, meliputi: pengumpulan data, penyesuaian label, pembagian dataset, pra-pemrosesan, perancangan model *transfer learning*, pelatihan model, dan evaluasi kinerja model. Rangkaian tahapan tersebut divisualisasikan pada Gambar 1.



Gambar 1. Alur Penelitian

2.1 Pengumpulan Dataset

Dataset yang digunakan merupakan RGB Arabic Alphabets Sign Language Dataset dari Kaggle yang dikembangkan oleh Muhammad Albrham (2023). Dataset ini terdiri atas 7.856 citra RGB berformat JPG/JPEG yang merepresentasikan 31 huruf hijaiyah [27]. Setiap kelas huruf direpresentasikan oleh variasi bentuk tangan dari beberapa individu. Contoh citra dataset ditampilkan pada Gambar 2. Struktur folder dan label dataset disimpan secara konsisten untuk mendukung replikasi eksperimen.



Gambar 2. Contoh Citra Bahasa Isyarat Huruf Hijaiyah

2.2 Penyesuaian Label

Label asli dataset berupa nama folder dalam bentuk transliterasi Arab. Penyesuaian label dilakukan secara manual untuk menyeragamkan penamaan menjadi format huruf hijaiyyah yang umum, seperti Alif, Ba, Ta, dan seterusnya, sebagaimana tercantum pada Tabel 1. Langkah ini dilakukan untuk memudahkan interpretasi hasil klasifikasi serta analisis kesalahan pada tahap evaluasi.

Tabel 1. Penyesuaian label

No	Label Asli	Label Baru	Huruf Hijaiyyah
1	alef	alif	ا
2	beh	ba	ب
3	teh	ta	ت
4	theh	tsa	ث
5	jeem	jim	ج
6	hah	ha	ح
7	khah	kha	خ
8	dal	dal	د
9	thal	dzal	ذ
10	rah	ra	ر
11	zain	zay	ز
12	seen	sin	س
13	sheen	syin	ش
14	sad	shad	ص
15	dad	dhad	ض
16	tah	tha	ط
17	zah	zha	ظ
18	ain	ain	ع
19	ghain	ghain	غ
20	feh	fa	ف
21	qaf	qaf	ق
22	kaf	kaf	ك
23	lam	lam	ل
24	meem	mim	م
25	noon	nun	ن
26	heh	Ha	هـ
27	waw	wau	و
28	yeh	ya	ي
29	teh marbuta	ta marbuta	ة
30	la	lam alif	لا
31	al	alif lam	ال

2.3 Pembagian Dataset

Dataset dibagi menjadi tiga subset, yaitu 80% data pelatihan, 10% data validasi, dan 10% data pengujian, sebagaimana ditunjukkan pada Tabel 2. Pembagian dilakukan menggunakan teknik *stratified sampling* untuk memastikan proporsi setiap kelas tetap seimbang di ketiga subset. *Stratified train-test split* merupakan teknik yang memastikan bahwa setiap kelas atau kelompok dalam dataset terwakili secara proporsional pada set pelatihan dan pengujian [28]. Pendekatan ini penting untuk menghindari bias distribusi kelas yang dapat memengaruhi performa model. Proses pembagian data dilakukan dengan nilai *random seed* tetap (*random_state=42*) untuk menjaga reproduktibilitas hasil eksperimen. Seluruh hasil pembagian disimpan dalam format *parquet* untuk memudahkan proses pemuatan dan pra-pemrosesan data.

Tabel 2. Pembagian Dataset

No	Data	Presentase	Jumlah
1	Data Latih	80%	6284
2	Data Validasi	10%	786

No	Data	Presentase	Jumlah
3	Data Uji	10%	786

2.4 Pra-pemrosesan Dataset

Proses pra-pemrosesan dilakukan untuk menstandarkan ukuran dan kualitas citra sebelum masuk ke tahap pelatihan. Setiap citra diubah ukurannya menjadi 256x256 piksel dan dipotong acak menjadi 224x224 piksel menggunakan *RandomResizedCrop*. Augmentasi data diterapkan pada subset pelatihan untuk meningkatkan variasi citra dan ketahanan model terhadap kondisi dunia nyata. Teknik augmentasi yang digunakan meliputi *horizontal flip*, rotasi acak, distorsi perspektif, *gaussian blur*, dan *color jittering*. Seluruh citra kemudian dinormalisasi berdasarkan nilai rata-rata (*mean*) dan standar deviasi (*standard deviation*) dari dataset ImageNet. Adapun data validasi dan data uji hanya dilakukan *resize* dan normalisasi tanpa augmentasi agar distribusi data tetap konsisten.

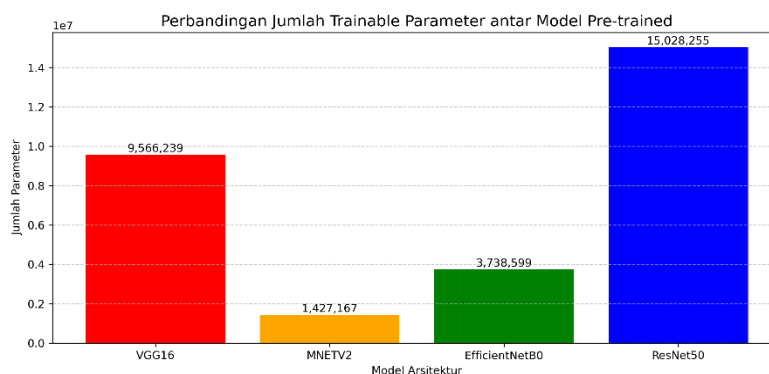
2.5 Desain Model *Transfer learning*

Empat arsitektur CNN pralatih digunakan, yaitu MobileNetV2, EfficientNetB0, ResNet50, dan VGG16. Model-model ini dipilih karena memiliki karakteristik yang berbeda dalam hal jumlah parameter, kecepatan inferensi, dan akurasi. Pemanfaatan model pralatih terbukti dapat meningkatkan akurasi dalam tugas klasifikasi visual, seperti pada klasifikasi gangguan retina menggunakan ResNet50 yang menunjukkan peningkatan akurasi dari 86% menjadi 92% [29], serta pada pengenalan huruf hijaiyyah menggunakan CNN dengan dataset AASL [8]. *Fine-tuning* dilakukan dengan membekukan sebagian besar lapisan awal dan hanya melatih ulang beberapa lapisan akhir serta *classifier*. Tabel 3 menyajikan konfigurasi *fine-tuning* untuk masing-masing arsitektur CNN, termasuk lapisan-lapisan yang dibuka kembali (*unfreeze*) selama proses pelatihan ulang.

Tabel 3. Konfigurasi *Fine-tuning* Model

No	Model	Layer yang di- <i>unfreeze</i>
1	MobileNetV2	features[13] hingga features[17] serta <i>classifier</i> .
2	EfficientNetB0	features[5] hingga features[8] dan <i>classifier</i> .
3	VGG16	features[24] hingga akhir dan <i>classifier</i> .
4	ResNet50	layer4 serta fc (<i>fully connected</i>)

Gambar 3 menunjukkan perbandingan jumlah parameter yang dilatih (*trainable parameters*) pada empat arsitektur CNN setelah *fine-tuning*. MobileNetV2 memiliki parameter paling sedikit, yaitu 1.427.167, menjadikannya ringan dan cepat untuk dilatih, cocok untuk implementasi di perangkat dengan sumber daya terbatas. Sementara itu, ResNet50 dan VGG16 menawarkan kapasitas pembelajaran yang lebih dalam, yang berpotensi menghasilkan akurasi lebih tinggi, namun memerlukan sumber daya komputasi yang lebih besar. Perbedaan jumlah parameter yang dilatih pada tiap model secara langsung dipengaruhi oleh banyaknya lapisan (*layer*) yang dibuka kembali selama proses *fine-tuning* [30], di mana semakin banyak lapisan yang dibuka untuk pelatihan ulang, maka jumlah *trainable parameters* akan semakin besar. Hal ini menjelaskan variasi dalam kapasitas pembelajaran dan kebutuhan komputasi di antara model-model tersebut. Perbedaan ini memengaruhi durasi pelatihan, kebutuhan perangkat keras, dan *trade-off* antara kecepatan dan akurasi dalam klasifikasi huruf hijaiyyah bahasa isyarat [31], [32].



Gambar 3. Perbandingan Jumlah Trainable Parameters pada Model Pralatih

2.6 Pelatihan Model

Proses pelatihan model dilakukan menggunakan konfigurasi hiperparameter sebagaimana ditunjukkan pada Tabel 4. Parameter utama yang digunakan meliputi *batch size* 32, *loss function* CrossEntropyLoss, *optimizer* Adam dengan learning rate awal 0.0005, dan *scheduler* ReduceLROnPlateau dengan factor 0,5 dan patience 5. Model disimpan secara otomatis menggunakan mekanisme *checkpoint* berdasarkan nilai akurasi validasi tertinggi untuk mencegah *overfitting*. Setiap *epoch* terdiri dari dua fase, yaitu pelatihan dan validasi. Metrik validasi dipantau setiap *epoch*, dan apabila tidak terjadi peningkatan selama beberapa iterasi, *learning rate* secara adaptif dikurangi oleh scheduler.

Tabel 4. Konfigurasi *Hyperparameter* Pelatihan

No	Hyperparameter	Nilai
1	Batch Size	32
2	Optimizer	Adam
3	Learning rate	0.0005
4	Loss function	CrossEntropyLoss
5	Epoch	30
6	Scheduler	ReduceLROnPlateau (mode= 'min', factor=0.5, patience=5)

2.7 Evaluasi Kinerja Model

Evaluasi kinerja model dilakukan untuk menilai kemampuan sistem dalam mengenali citra huruf hijaiyah Bahasa Isyarat Arab. Pengujian menggunakan beberapa metrik, yaitu *accuracy*, *precision*, *recall*, dan *F1-score*.

Accuracy menunjukkan persentase prediksi benar terhadap seluruh data uji, sebagaimana ditunjukkan pada Persamaan (1)

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

Precision dan *recall* digunakan untuk mengukur ketepatan serta kemampuan model dalam mengenali kelas positif, seperti pada Persamaan (2) dan (3).

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

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

Untuk menyeimbangkan keduanya, digunakan *F1-score* yang merupakan rata-rata harmonik antara *precision* dan *recall*, sebagaimana Persamaan (4).

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

Selain metrik tersebut, confusion matrix digunakan untuk melihat distribusi kesalahan prediksi, sedangkan ukuran model dan waktu inferensi dianalisis untuk menilai efisiensi komputasi.

2.8 Lingkungan Eksperimen

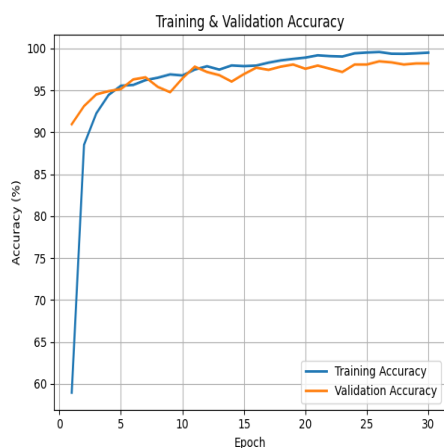
Eksperimen dilaksanakan menggunakan bahasa pemrograman Python 3.10 dengan kerangka kerja PyTorch 2.1.0 dan Torchvision 0.15.2 pada platform Kaggle Notebook. Proses pelatihan model dijalankan pada CPU Intel Xeon dengan kapasitas RAM 16 GB tanpa dukungan akselerasi GPU. Selain itu, pustaka Torchprofile 0.0.4 digunakan untuk menghitung kompleksitas model dan waktu inferensi. Lingkungan ini dipilih karena mendukung integrasi pustaka *deep learning* secara optimal serta memberikan stabilitas dan efisiensi dalam pelaksanaan eksperimen.

3. HASIL DAN PEMBAHASAN

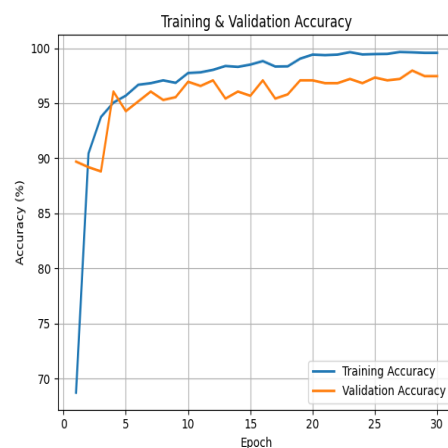
3.1. Hasil Pelatihan dan Validasi

Pelatihan model dilakukan menggunakan pendekatan *transfer learning* dengan teknik *fine-tuning* pada empat arsitektur CNN pralatih, yaitu VGG16, MobileNetV2, EfficientNetB0, dan ResNet50. Pemilihan keempat model ini didasarkan pada keragaman kompleksitas arsitektur dan efisiensi komputasi, sehingga representatif untuk kebutuhan sistem klasifikasi berbasis citra yang akurat dan efisien. Dalam proses *fine-tuning*, seluruh bobot awal model dibekukan terlebih dahulu, kemudian beberapa lapisan akhir diaktifkan kembali untuk dilatih ulang. Selain itu, lapisan *classifier* pada masing-masing model dimodifikasi agar dapat menghasilkan keluaran sebanyak 31 kelas, sesuai dengan jumlah huruf dalam dataset huruf hijaiyyah Bahasa Isyarat Arab. Rincian lapisan yang diaktifkan kembali dapat dilihat pada Tabel 3, sementara konfigurasi *hyperparameter* pelatihan disajikan pada Tabel 4. Proses pelatihan dilakukan selama 30 *epoch*, dengan pemilihan model terbaik berdasarkan akurasi validasi tertinggi yang diperoleh pada setiap *checkpoint*.

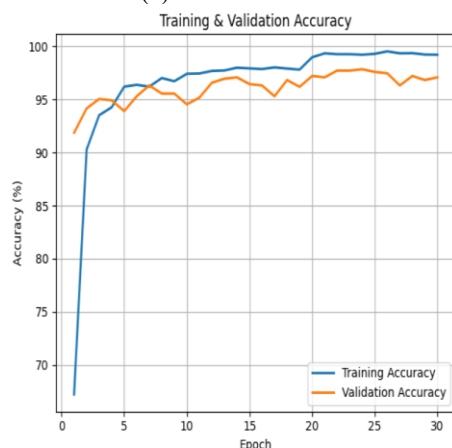
Gambar 4 menampilkan kurva pelatihan dan validasi dari masing-masing model. Secara umum, seluruh model menunjukkan performa tinggi, dengan akurasi validasi di atas 97%. MobileNetV2 (Gambar 4a) menunjukkan performa terbaik dengan akurasi validasi tertinggi sebesar 0,9847 pada *epoch* ke-26 dan akurasi pelatihan sebesar 0,9959. ResNet50 (Gambar 4b) menyusul dengan akurasi validasi 0,9796 pada *epoch* ke-28 dan akurasi pelatihan 0,9962. EfficientNetB0 (Gambar 4c) mencatat akurasi validasi 0,9784 dan pelatihan 0,9920 pada *epoch* ke-24, sementara VGG16 (Gambar 4d) memperoleh akurasi validasi 0,9733 dan pelatihan 0,9839 pada *epoch* ke-26. Rangkuman nilai akurasi pelatihan dan validasi dari keempat model disajikan pada Tabel 5. Perbedaan akurasi antar model relatif kecil, namun tetap signifikan dalam konteks efisiensi komputasi dan kestabilan performa model.



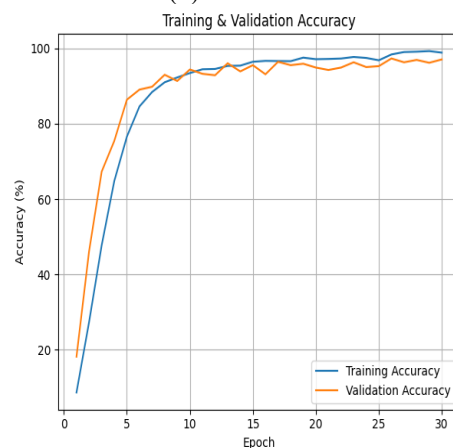
(a) MobileNetV2



(b) ResNet50



(c) EfficientNetB0



(d) VGG16

Gambar 4. Kurva pelatihan dan validasi dari masing-masing model: (a) MobileNetV2, (b) ResNet50, (c) EfficientNetB0, dan (d) VGG16.

Tabel 5 menyajikan ringkasan nilai akurasi pelatihan dan validasi dari keempat model.

Tabel 5. Nilai Akurasi Pelatihan dan Validasi				
No	Model	Latih	Validasi	Epoch
1	MobileNetV2	0.9959	0.9847	26
2	EfficientNetB0	0.9920	0.9784	24
3	VGG16	0.9839	0.9733	26
4	ResNet50	0.9962	0.9796	28

3.2. Hasil Pengujian

Pengujian dilakukan terhadap data uji yang tidak pernah dilihat oleh model selama proses pelatihan, sebagaimana dijelaskan pada Tabel 6. Tujuan utama pengujian ini adalah untuk mengevaluasi kemampuan generalisasi model terhadap data baru. Evaluasi dilakukan menggunakan metrik *accuracy*, *precision*, *recall*, dan *F1-score*, sebagaimana ditunjukkan pada Tabel 6. Hasil pengujian menunjukkan bahwa ResNet50 memberikan performa terbaik dengan akurasi 0,9835, *precision* 0,98, *recall* 0,98, dan *F1-score* 0,98. MobileNetV2 berada di posisi kedua dengan akurasi 0,9784 dan nilai *precision*, *recall*, serta *F1-score* sebesar 0,98. Selanjutnya, EfficientNetB0 mencatat akurasi 0,9771 dan VGG16 memperoleh akurasi 0,9707. Seluruh model berhasil mempertahankan akurasi di atas 97% pada data uji, menegaskan efektivitas pendekatan *transfer learning* dalam proses klasifikasi huruf hijaiyyah pada Bahasa Isyarat Arab.

Tabel 6. Hasil Kinerja Model pada Data Uji					
No	Model	Precision	Recall	F1-score	Accuracy
1	MobileNetV2	0.98	0.98	0.98	0.9784
2	EfficientNetB0	0.98	0.98	0.98	0.9771
3	VGG16	0.97	0.97	0.97	0.9707
4	ResNet50	0.98	0.98	0.98	0.9835

3.3 Evaluasi Komputasi

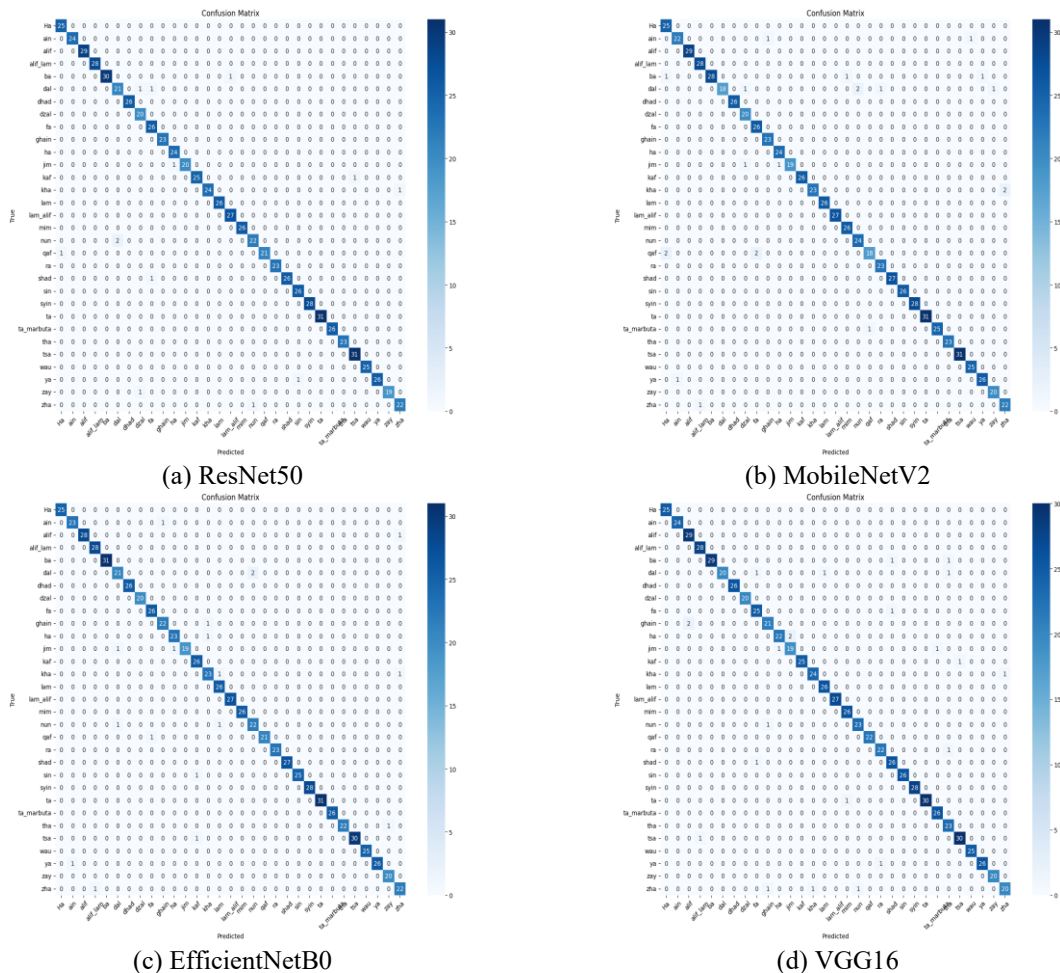
Analisis efisiensi komputasi dilakukan untuk menilai kesesuaian model terhadap implementasi pada perangkat dengan keterbatasan sumber daya atau aplikasi *real-time*. Parameter yang dievaluasi meliputi model dan waktu inferensi rata-rata per citra, sebagaimana ditampilkan pada Tabel 7. MobileNetV2 menampilkan keseimbangan terbaik antara ukuran dan kecepatan dengan ukuran model sebesar 8,86 MB dan waktu inferensi 30,63 ms per citra. Hal ini menjadikannya arsitektur yang sangat sesuai untuk aplikasi real time pada perangkat bergerak. Sebaliknya, VGG16 menunjukkan ukuran dan latensi terbesar yaitu 516 MB dengan kecepatan inferensi 345,04 ms per citra, sehingga kurang ideal untuk penggunaan di lingkungan dengan keterbatasan memori.

ResNet50 dan EfficientNetB0 menempati posisi Tengah, menawarkan kompromi antara akurasi tinggi dan efisiensi komputasi yang masih dapat diterima. Hasil ini menunjukkan bahwa pemilihan arsitektur CNN perlu mempertimbangkan ujian implementasi, apakah untuk mendapatkan akurasi maksimum atau efisiensi dalam penggunaan sumber daya.

Tabel 7. Ukuran dan Kecepatan Model			
No	Model	Ukuran Model (Mb)	Kecepatan Inferensi (ms/sample)
1	MobileNetV2	8,86	30,63
2	EfficientNetB0	15,72	36,40
3	VGG16	512,66	345,04
4	ResNet50	90,22	131,62

3.4 Analisis Confusion Matrix

Analisis *confusion matrix* ditunjukkan pada Gambar 5, yang memperlihatkan distribusi hasil klasifikasi keempat arsitektur CNN pralatih terhadap 31 kelas huruf hijaiyyah. Gambar 5a–d masing-masing menunjukkan hasil dari ResNet50, MobileNetV2, EfficientNetB0, dan VGG16. Visualisasi tersebut memberikan gambaran kuantitatif mengenai kemampuan model dalam mengenali kelas secara benar (*true positive*) serta pola kesalahan prediksi (*misclassification*). Dari matriks ini terlihat variasi tingkat akurasi antar kelas dan huruf yang paling sering mengalami kekeliruan prediksi.



Gambar 5. Confusion Matrix hasil pengujian model ResNet50 (a), MobileNetV2 (b), EfficientNetB0 (c), dan VGG16 (d)

Berdasarkan Berdasarkan hasil pada Gambar 5, kesalahan klasifikasi paling banyak terjadi pada huruf dengan kemiripan morfologis tinggi, seperti dal-dzal, ta-tha, dan kha-ghain. Pola ini menunjukkan bahwa model masih menghadapi kesulitan dalam membedakan fitur visual halus yang melibatkan jumlah titik, orientasi jari, atau posisi tangan. Faktor eksternal seperti variasi pencahayaan, sudut pengambilan gambar, dan latar belakang juga turut memengaruhi tingkat ambiguitas antar kelas.

Mengurangi kesalahan serupa, penelitian selanjutnya dapat menerapkan augmentasi morfologis adaptif, seperti rotasi kecil, translasi halus, dan *local zoom* untuk menekankan fitur detail. Selain itu, penggunaan *focal loss* dapat membantu menyeimbangkan pembelajaran antar kelas, sedangkan integrasi fitur landmark tangan berbasis MediaPipe berpotensi memperkuat representasi spasial dan meningkatkan diskriminasi antar huruf yang memiliki bentuk serupa.

3.5 Diskusi dan Temuan

Hasil penelitian menunjukkan bahwa seluruh model CNN pralatih mencapai kinerja klasifikasi yang tinggi dengan akurasi di atas 97%. Temuan ini menegaskan efektivitas pendekatan *transfer learning* dengan teknik *fine-tuning* dalam klasifikasi huruf hijaiyah Bahasa Isyarat Arab. Di antara model yang diuji, ResNet50 mencatat akurasi tertinggi sebesar 98,35%, sementara MobileNetV2 menampilkan efisiensi komputasi terbaik dengan ukuran model terkecil dan waktu inferensi tercepat. Hasil tersebut menunjukkan adanya *trade-off* antara kompleksitas arsitektur dan efisiensi komputasi: model yang lebih dalam memiliki kemampuan representasi fitur yang lebih kuat, namun memerlukan sumber daya pelatihan dan inferensi yang lebih besar.

Perbandingan hasil penelitian ini dengan studi-studi terdahulu disajikan pada Tabel 8. Secara umum, capaian akurasi yang diperoleh sejalan dengan studi Balat et al. [14] dan Ismail et al. [15], yang juga melaporkan performa unggul arsitektur ResNet50 dalam pengenalan huruf Arab berbasis citra tangan. Namun, variasi hasil akurasi antar

penelitian dapat disebabkan oleh perbedaan karakteristik dataset, jumlah kelas, teknik augmentasi, serta parameter pelatihan yang digunakan.

Tabel 8. Perbandingan hasil akurasi dengan penelitian terdahulu

Referensi	Dataset	Model	Pendekatan	Akurasi (%)
[7]	Bahasa Isyarat Arab (ASL-DS-III)	ResNet50, EfficientNet, MobileNetV2	Transfer learning	96 - 97
[11]	ArSL2018	InceptionV3, EfficientNetB0	Fine-tuning	99,0
[16]	Huruf Hijaiyah Isyarat (31 kelas)	MobileNetV2, VGG16, Xception, ResNet50	Transfer learning dan fine-tuning	82 – 99,85
[19]	Bahasa Isyarat Umum (Real-time)	MobileNetV2	Transfer learning dan augmentasi data	97,0
Penelitian in	Huruf Hijaiyah Isyarat (31 kelas)	ResNet50, MobileNetV2, EfficientNetB0, VGG16	Fine-tuning dan optimasi hyperparameter	97,07 – 98,35

Berdasarkan tabel tersebut, terlihat bahwa hasil penelitian ini berada pada rentang akurasi yang kompetitif dengan studi terdahulu, namun menawarkan keunggulan dalam efisiensi dan stabilitas performa antar model. Pendekatan *fine-tuning* yang selektif serta penerapan augmentasi moderat terbukti mampu menjaga keseimbangan antara akurasi dan efisiensi komputasi tanpa menyebabkan overfitting.

Analisis *confusion matrix* mengungkap bahwa kesalahan klasifikasi paling sering terjadi pada huruf dengan kemiripan morfologis tinggi, seperti dal-dzal dan ta-tha. Kesalahan tersebut terutama disebabkan oleh variasi pencahayaan, orientasi tangan, serta perbedaan kecil pada posisi jari yang sulit dibedakan secara visual. Untuk mengurangi hal tersebut, penelitian lanjutan dapat mengeksplorasi augmentasi morfologis adaptif dan pra-pemrosesan berbasis kontur tangan guna meningkatkan sensitivitas model terhadap perbedaan fitur spasial yang halus.

Secara keseluruhan, hasil penelitian ini memperkuat bukti empiris bahwa kombinasi *transfer learning* dan *fine-tuning* efektif untuk pengenalan huruf hijaiyah berbasis bahasa isyarat. Selain itu, efisiensi MobileNetV2 menjadikannya kandidat potensial untuk diterapkan pada sistem pembelajaran visual interaktif berbasis perangkat bergerak, khususnya dalam mendukung komunikasi inklusif bagi penyandang tunarungu.

4. KESIMPULAN

Penelitian ini menunjukkan bahwa pendekatan *transfer learning* dengan teknik *fine-tuning* pada arsitektur CNN pralatih efektif dalam mengenali huruf hijaiyah pada Bahasa Isyarat Arab (ArSL). Keempat model yang diuji, yaitu ResNet50, MobileNetV2, EfficientNetB0, dan VGG1, seluruhnya menunjukkan performa tinggi dengan akurasi di atas 97%. Model ResNet50 mencapai akurasi tertinggi sebesar 98,35%, sedangkan MobileNetV2 menampilkan efisiensi terbaik melalui ukuran model terkecil dan waktu inferensi tercepat. Temuan ini menegaskan adanya keseimbangan antara kompleksitas arsitektur dan efisiensi komputasi yang perlu dipertimbangkan dalam pengembangan sistem berbasis pembelajaran visual.

Secara akademik, penelitian ini memperkuat bukti empiris bahwa *fine-tuning* selektif pada model CNN pralatih dapat menghasilkan generalisasi yang optimal, bahkan pada dataset dengan jumlah data terbatas. Hasil ini turut memperkaya literatur terkait penerapan *deep learning* untuk pengenalan huruf hijaiyah berbasis isyarat, khususnya dalam konteks pendidikan agama bagi penyandang tunarungu.

Dari sisi praktis, model MobileNetV2 dengan efisiensi tinggi berpotensi diintegrasikan ke dalam aplikasi pembelajaran interaktif berbasis Android secara *real-time*. Implementasi tersebut dapat mendukung proses pembelajaran huruf hijaiyah secara visual dan inklusif. Selain itu, konsistensi performa antar model menunjukkan bahwa sistem pengenalan huruf isyarat ini layak dijadikan dasar bagi pengembangan platform pembelajaran berbasis visi komputer di lingkungan pendidikan khusus.

Sebagai tindak lanjut, penelitian selanjutnya disarankan untuk melakukan validasi sistem bersama pengguna tunarungu guna menilai aspek akurasi fungsional dan pengalaman pengguna (*usability*). Selain itu, integrasi

dengan sistem pelacakan tangan seperti MediaPipe dapat meningkatkan ketepatan pengenalan gerakan dinamis tanpa mengorbankan efisiensi komputasi.

DAFTAR PUSTAKA

- [1] B. Pamungkas and H. Hermanto, "Tahapan Belajar Al Qur'an Menggunakan Huruf Hijaiyah Isyarat bagi Anak dengan Hambatan Pendengaran," *Jurnal Pendidikan Kebutuhan Khusus*, vol. 6, no. 1, pp. 34–41, Apr. 2022, doi: 10.24036/jpkk.v6i1.621.
- [2] H. A. Abdelghfar et al., "QSLRS-CNN: Qur'anic sign language recognition system based on convolutional neural networks," *The Imaging Science Journal*, vol. 72, no. 2, pp. 254–266, Feb. 2024, doi: 10.1080/13682199.2023.2202576.
- [3] A. Setiawan, "Intelligent Systems For Disabilities Technology For A More Inclusive Life," 2025. doi: 10.2139/ssrn.5062273.
- [4] N. M. Alharthi and S. M. Alzahrani, "Vision Transformers and Transfer learning Approaches for Arabic Sign Language Recognition," *Applied Sciences*, vol. 13, no. 21, p. 11625, Oct. 2023, doi: 10.3390/app132111625.
- [5] J. Sánchez, J. Reyes-Rojas, and J. Alé-Silva, "What Is Known about Assistive Technologies in Distance and Digital Education for Learners with Disabilities?" *Education Sciences*, vol. 14, no. 6, p. 595, May 2024, doi: 10.3390/educsci14060595.
- [6] S. A. Alhulays, "The Role Of Assistive Technology in Supporting Communication and Academic Access For Deaf Students: A Qualitative Study At Gallaudet University," *Baileo: Jurnal Sosial Humaniora*, vol. 2, no. 1, pp. 97–106, Sep. 2024, doi: 10.30598/baileofisipvol2iss1pp97-106.
- [7] S. Al Ahmadi, F. Muhammad, and H. Al Dawsari, "Enhancing Arabic Sign Language Interpretation: Leveraging Convolutional Neural Networks and Transfer learning," *Mathematics*, vol. 12, no. 6, p. 823, Mar. 2024, doi: 10.3390/math12060823.
- [8] R. El Kharoua and X. Jiang, "Deep learning Recognition for Arabic Alphabet Sign Language RGB Dataset," *Journal of Computer and Communications*, vol. 12, no. 03, pp. 32–51, 2024, doi: 10.4236/jcc.2024.123003.
- [9] I. D. Mienye, T. G. Swart, G. Obaido, M. Jordan, and P. Ilono, "Deep Convolutional Neural Networks in Medical Image Analysis: A Review," *Information*, vol. 16, no. 3, p. 195, Mar. 2025, doi: 10.3390/info16030195.
- [10] Z. S. Bilal, A. Gargouri, H. F. Mahmood, and H. Mnif, "Comparison of Collective Diverse Arabic Sign Language Dataset," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 15, no. 4, pp. 133–150, Dec. 2024, doi: 10.58346/JOWUA.2024.14.009.
- [11] A. Hussain, S. Ul Amin, M. Fayaz, and S. Seo, "An Efficient and Robust Hand Gesture Recognition System of Sign Language Employing Finetuned Inception-V3 and Efficientnet-B0 Network," *Computer Systems Science and Engineering*, vol. 46, no. 3, pp. 3509–3525, 2023, doi: 10.32604/csse.2023.037258.
- [12] P. Poladiya, D. Suresh, P. Gulhane, M. A. Ajmal, and P. Kosamkar, "Sign Language Detection Using Deep learning," in *Proc. 3rd Int. Conf. for Innovation in Technology (INOCON)*, IEEE, Mar. 2024, pp. 1–6, doi: 10.1109/INOCON60754.2024.10512307.
- [13] B. Alsharif, A. S. Altaher, A. Altaher, M. Ilyas, and E. Alalwany, "Deep learning Technology to Recognize American Sign Language Alphabet," *Sensors*, vol. 23, no. 18, p. 7970, Sep. 2023, doi: 10.3390/s23187970.
- [14] M. Balat, R. Awaad, H. Adel, A. B. Zaky, and S. A. Aly, "Advanced Arabic Alphabet Sign Language Recognition Using Transfer learning and Transformer Models," *arXiv preprint*, Oct. 2024. [Online]. Available: <https://arxiv.org/abs/2410.00681>
- [15] M. H. Ismail, S. A. Dawwd, and F. H. Ali, "Static hand gesture recognition of Arabic sign language by using deep CNNs," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 24, no. 1, pp. 178–188, Oct. 2021, doi: 10.11591/ijeecs.v24.i1.pp178-188.
- [16] Y. Brianorman and R. Munir, "Perbandingan Pre-Trained CNN: Klasifikasi Pengenalan Bahasa Isyarat Huruf Hijaiyah," *Jurnal Sistem Informasi Bisnis*, vol. 13, no. 1, pp. 52–59, Jul. 2023, doi: 10.21456/vol13iss1pp52-59.
- [17] U. Fadlilah, A. K. Mahamad, and B. Handaga, "The Development of Android for Indonesian Sign Language Using Tensorflow Lite and CNN: An Initial Study," *Journal of Physics: Conference Series*, vol. 1858, no. 1, p. 012085, Apr. 2021, doi: 10.1088/1742-6596/1858/1/012085.

-
- [18] M. F. Nurnoby, E.-S. M. El-Alfy, and H. Luqman, "Evaluation of CNN Models with Transfer learning for Recognition of Sign Language Alphabets with Complex Background," in *Proc. Int. Conf. on Innovation and Intelligence for Informatics, Computing and Technologies (3ICT)*, IEEE, Dec. 2020, pp. 1–6. doi: 10.1109/3ICT51146.2020.9311989.
- [19] R. Temkar, S. Jagtap, K. Jadhav, and M. Deshmukh, "Real-time Sign Language Recognition Using MobileNetV2 and Transfer learning."
- [20] R. Poojary, R. Raina, and A. Kumar Mondal, "Effect of data-augmentation on fine-tuned CNN model performance," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 10, no. 1, p. 84, Mar. 2021, doi: 10.11591/ijai.v10.i1.pp84-92.
- [21] H. Imaduddin, F. Yusfida Ala, A. Fatmawati, and B. A. Hermansyah, "Comparison of transfer learning method for COVID-19 detection using convolution neural network," *Bulletin of Electrical Engineering and Informatics*, vol. 11, no. 2, pp. 1091–1099, Apr. 2022, doi: 10.11591/eei.v11i2.3525.
- [22] M. R. Sadik et al., "Computer Vision Based Bangla Sign Language Recognition Using Transfer learning," in *Proc. 2nd Int. Conf. on Data Science and Information System (ICDSIS)*, IEEE, May 2024, pp. 1–7. doi: 10.1109/ICDSIS61070.2024.10594269.
- [23] M. A. A. Mosleh, A. Assiri, A. H. Gumaiei, B. F. Alkhamees, and M. Al-Qahtani, "A Bidirectional Arabic Sign Language Framework Using Deep learning and Fuzzy Matching Score," *Mathematics*, vol. 12, no. 8, p. 1155, Apr. 2024, doi: 10.3390/math12081155.
- [24] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. Cambridge, MA: MIT Press, 2016.
- [25] F. Chollet, *Deep learning with Python*, 2nd ed. Shelter Island, NY: Manning Publications, 2021.
- [26] E. Stevens, L. Antiga, and T. Viehmann, *Deep learning with PyTorch*. Shelter Island, NY: Manning Publications, 2020.
- [27] M. Al-Barham, A. Alsharkawi, M. Al-Yaman, M. Al-Fetyani, A. Elnagar, A. A. SaAleek, and M. Al-Odat, "RGB Arabic Alphabets Sign Language Dataset," *arXiv preprint arXiv:2301.11932*, Jan. 2023, doi: 10.48550/arXiv.2301.11932.
- [28] N. Trung Tuyen, N. Thanh Dang, D. Hon Minh, and T. Q. Nguyen, "Enhancing human activity recognition model performance through KMeans-based stratified data splitting," *Measurement and Control*, Jan. 2025, doi: 10.1177/00202940241312873.
- [29] H. Imaduddin, I. C. Utomo, and D. A. Anggoro, "Fine-tuning ResNet-50 for the classification of visual impairments from retinal fundus images," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 14, no. 4, p. 4175, Aug. 2024, doi: 10.11591/ijece.v14i4.pp4175-4182.
- [30] D. Griebßhaber, J. Maucher, and N. T. Vu, "Fine-Tuning BERT for Low-Resource Natural Language Understanding via Active Learning," in *Proc. 28th Int. Conf. on Computational Linguistics*, Stroudsburg, PA: ICCL, 2020, pp. 1158–1171, doi: 10.18653/v1/2020.coling-main.100.
- [31] W. N. Ismail, H. A. Alsalamah, and E. A. Mohamed, "Genetic-Efficient Fine-Tuning with Layer Pruning on Multimodal Covid-19 Medical Imaging," *Neural Computing and Applications*, vol. 36, no. 6, pp. 3215–3237, Feb. 2024, doi: 10.1007/s00521-023-09194-5.
- [32] A. Davila, J. Colan, and Y. Hasegawa, "Comparison of Fine-Tuning Strategies for Transfer Learning in Medical Image Classification," *Image and Vision Computing*, vol. 146, p. 105012, Jun. 2024, doi: 10.1016/j.imavis.2024.105012.