



Perbandingan *Pre-Trained* CNN: Klasifikasi Pengenalan Bahasa Isyarat Huruf Hijaiyah

Yulrio Brianorman^{a,*}, Rinaldi Munir^b

^a Universitas Muhammadiyah Pontianak

^b Institut Teknologi Bandung

Naskah Diterima: 18 Februari 2023; Diterima Publikasi: 05 Juli 2023

DOI: 10.21456/vol13iss1pp52-59

Abstract

The number of documented deaf people continues to increase. To communicate with each other, the deaf use sign language. The problem arises when Muslims with hearing impairment or deafness need to recite the Al-Quran. Muslims recite Al-Quran using their voice, but for the deaf, there are no available means to do the reciting. Thus, learning hijaiyah letters using finger gestures is considered important to develop. In this study, we use the recognition of hijaiyah letters based on pictures as the learning model. The real-time-based recognition then uses the learning model. This study uses 4 CNN pre-trained models, namely MnetV2, VGG16, ResNet50, and Xception. The learning process shows that MnetV2, VGG16, and Xception reach the accuracy limit of 99.85% in 2, 3, and 11 s, respectively. Meanwhile, ResNet50 cannot reach the accuracy limit after processing 100 s. ResNet50 achieves 82.12% accuracy. The testing process shows that MnetV2, VGG16, and ResNet50 achieve 100% precision, recall, f1-score, and accuracy. ResNet50 shows figures 81.55%, 86.04%, 82.04%, and 82.58%. The implementing process of the learning outcomes from MnetV2 shows good performance for recognizing finger shapes in real-time.

Keywords : Sign Language; Hijaiyah Letters; Pre-Trained Model; CNN

Abstrak

Jumlah tuna rungu yang didokumentasikan terus meningkat. Untuk berkomunikasi satu sama lain, kaum tuli menggunakan bahasa isyarat. Masalah muncul ketika umat Islam yang tuli perlu membaca Al-Quran. Muslim membaca Al-Quran dengan suara mereka, tetapi untuk kaum tuli, tidak ada sarana yang tersedia untuk membaca. Dengan demikian, pembelajaran huruf hijaiyah dengan menggunakan gestur jari dianggap penting untuk dikembangkan. Dalam penelitian ini, kami menggunakan pengenalan huruf hijaiyah berdasarkan gambar sebagai model pembelajaran. Pengenalan berbasis waktu nyata kemudian menggunakan model pembelajaran. Penelitian ini menggunakan 4 *pre-trained* model CNN yaitu MnetV2, VGG16, ResNet50, dan Xception. Proses pembelajaran menunjukkan bahwa MnetV2, VGG16, dan Xception mencapai batas akurasi 99,85% masing-masing dalam 2, 3, dan 11 *epoch*. Sementara itu, ResNet50 tidak dapat mencapai batas akurasi setelah memproses 100 *epoch*. ResNet50 mencapai akurasi 82,12%. Proses pengujian menunjukkan bahwa MnetV2, VGG16, dan ResNet50 mencapai 100% *precision*, *recall*, *f1-score*, dan *accuracy*. ResNet50 menunjukkan angka 81,55%, 86,04%, 82,04%, dan 82,58%. Proses implementasi hasil pembelajaran dari MnetV2 menunjukkan kinerja yang baik untuk pengenalan bentuk jari secara waktu nyata.

Kata Kunci : Bahasa Isyarat; Huruf Hijaiyah; *Pre-Trained* Model; CNN

1. Pendahuluan

Sejak awal umat manusia telah menggunakan gerakan untuk berkomunikasi, menjelaskan, dan melengkapi pemahaman ekspresi mereka. Kaum tuna rungu menggunakan bahasa isyarat sebagai alternatif bahasa lisan. Mereka cenderung menggunakan bahasa isyarat seperti isyarat tangan, isyarat tubuh, dan ekspresi wajah sebagai media komunikasi. Bahasa isyarat memiliki seperangkat gerakan yang

terstruktur di mana setiap gerakan memiliki makna tertentu.

Selain itu, terdapat pula bahasa isyarat dalam bentuk karakter. Misalkan, untuk huruf latin ada bentuk jari tangan dari abjad “a” sampai dengan “z”. Sedangkan untuk huruf arab juga terdapat bentuk jari tangan dari huruf “alif” sampai dengan “ya” dan ditambah beberapa huruf lainnya seperti alif lam dan ta marbutah yang memiliki makna tertentu, ini ditunjukkan pada Gambar 1.

*) Penulis korespondensi: y.brianorman@unmuhpnk.ac.id

Pada penelitian ini akan dilakukan klasifikasi bahasa isyarat huruf hijaiyah menggunakan model pre-trained CNN. Hal ini penting dilakukan dikarenakan kaum tuli masih belum banyak mengetahui huruf hijaiyah dalam bentuk bahasa isyarat. Bahasa isyarat dalam bentuk huruf hijaiyah diperlukan oleh mereka untuk membaca Al Quran.

Alur penelitian ini secara singkat diawali dengan pembuatan dataset huruf hijaiyah yang digunakan pada Al-Quran seperti yang terlihat pada Gambar 1. Kemudian dilakukan *pre-processing* pada dataset tersebut, selanjutnya persiapan dataset untuk proses pelatihan, validasi dan pengujian. Terakhir, desain model *pre-trained* CNN, evaluasi kinerja dan implementasi model.



Gambar 1 Bahasa Isyarat Huruf Hijaiyah

Proses klasifikasi pada bidang *computer vision* menggunakan metode CNN. Penelitian yang memanfaatkan CNN untuk melakukan klasifikasi dilakukan oleh Yudistira *et al.* (2020). Penelitian ini dilakukan untuk mendeteksi COVID-19 dari photo thorax pasien. Hasil klasifikasinya adalah COVID-19, Pneumonia dan normal. Penelitian ini membandingkan model *pre-trained* CNN yang telah dikembangkan yaitu ShuffleNet, ResNet50, EfficientNet dan FullConv. Hasil menunjukkan bahwa ResNet memiliki akurasi tertinggi yaitu 90.8%. Peneliti Swastika (2020) juga melakukan pendeteksian COVID-19 menggunakan citra thorax. Hasil akurasi yang ditunjukkan sebesar 96,86% untuk klasifikasi infeksi COVID-19 dan normal.

Penelitian yang membandingkan model *pre-trained* CNN untuk menentukan tingkat kesegaran seekor ikan bandeng berdasarkan citra mata dilakukan oleh Prasetyo *et al.* (2021). Pada penelitian ini digunakan 4 model CNN yaitu Xception, MobileNet V2, ResNet50 dan VGG16. Hasil menunjukkan bahwa VGG16 memiliki akurasi terbaik yaitu 97%.

Analisis performa dari model *pre-trained* CNN untuk mendeteksi penyakit TBC telah dilakukan oleh Rochmawanti *et al.* (2021). Pada penelitian dianalisis 5 pre-train model yaitu ResNet50, DenseNet121, MobileNet, Xception, InceptionV3, dan

InceptionResNetV2. Hasil pengujian menunjukkan bahwa DenseNet121 memberikan hasil terbaik yaitu 91,57%. Sedangkan MobileNet merupakan model yang memiliki komputasi tercepat.

Penelitian terkait pengenalan bahasa isyarat dilakukan oleh Nurhayati *et al.* (2022). Penelitian ini menggunakan data berupa bahasa isyarat sebanyak 29 citra, yang terdiri dari 26 huruf alfabet dan 3 isyarat tambahan yaitu spasi, hapus dan tidak terklasifikasi. Akurasi yang didapat sebesar 97,2%.

Penelitian lain juga dilakukan oleh Barbhuiya *et al.* (2021) untuk melakukan klasifikasi dari bahasa isyarat Amerika. Peneliti melakukan modifikasi pada *pre-trained* model AlexNet dan VGG16 yang dipadukan dengan SVM. Hasil penelitian menunjukkan bahwa AlexNet dan VGG16 menghasilkan akurasi yang sama yaitu 70%. Penelitian lain terkait bahasa isyarat juga dilakukan oleh Ameur *et al.* (2020). Penelitian ini melakukan klasifikasi pada bahasa isyarat. Namun pada penelitian klafiksikasi menggunakan model bidirectional unidirectional LSTM. Akurasi yang ditunjukkan pada penelitian ini sebesar 90,13% dengan sebanyak 200.

Penelitian mengenai pengenalan bentuk jari tangan bahasa isyarat dilakukan oleh Islam *et al.* (2018). Penelitian ini kumpulan data yang digunakan adalah American Sign Language (ASL). Dua tahapan proses dilakukan pada penelitian ini yaitu tahapan ekstraksi fitur menggunakan CNN dan pengenalan bahasa isyarat dengan menggunakan metode Support Vector Machine (SVM). Hasil dari penelitian ini menunjukkan akurasi klasifikasi sebesar 94,57%. Penelitian serupa (Aly *et al.*, 2019; Ameen dan Vadera, 2017; Bin Makhshen *et al.*, 2019; Chevtchenko *et al.*, 2018; Hosoe *et al.*, 2017; Tao *et al.*, 2018) juga telah dilakukan oleh para peneliti lainnya.

Pengembangan model *pre-trained* CNN memberikan hasil yang baik untuk klasifikasi citra. Hal ini yang mendorong peneliti untuk melakukan kajian pada model *pre-trained* untuk klasifikasi huruf hijaiyah. Penelitian ini menggunakan metode *transfer learning* pada beberapa model *pre-trained* CNN yang populer seperti MnetV2, VGG16, ResNet50 dan Xception. Proses ekstraksi fitur dari konvolusi dilakukan secara langsung oleh CNN sesuai dengan rancangan model. Layer klasifikasi dilakukan dengan mengintegrasikan *fully connected multi layer perceptron*. Ujicoba ini dilakukan dengan penerapan *transfer learning* pada pengenalan bahasa isyarat huruf hijaiyah pada *pre-trained* CNN yang tersedia. Percobaan pada berbagai model CNN adalah menentukan model *pre-trained* CNN yang tepat untuk digunakan untuk klasifikasi tersebut hal ini dilakukan karena tidak semua model *pre-train* bisa belajar dengan baik untuk semua jenis datasets. Sehingga perlu dilakukan eksperimen model *pre-train* CNN mana yang tepat untuk sebuah dataset.

Adapun kebaruan dari penelitian ini sebagai berikut:

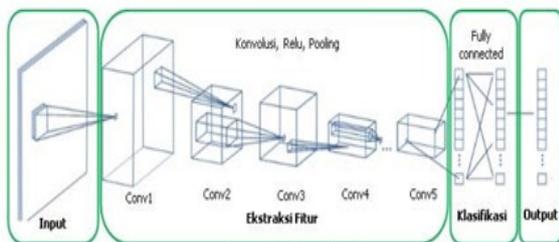
1. Pembuatan dataset dari berupa citra untuk bahasa isyarat huruf hijaiyah. Ini dilakukan dikarenakan ada terdapat 3 huruf yang tidak masuk dalam standar bahasa isyarat Arab yaitu ta marbuta, lam alif dan alif lam.
2. Desain model *pre-trained* CNN yang akan digunakan untuk proses klasifikasi. Kemudian, melakukan perbandingan model *pre-trained* CNN untuk mengetahui *pre-trained* yang terbaik untuk proses pengenalan dari dataset yang dibentuk.
3. Implementasi hasil pelatihan pada aplikasi pengenalan bahasa isyarat huruf hijaiyah.

2. Kerangka Teori

2.1. Convolutional Neural Network

Convolutional Neural Network, disingkat CNN atau ConvNet, merupakan metode yang banyak diaplikasikan untuk analisis citra. Peran utama dari CNN adalah melakukan ekstraksi fitur dari sebuah citra. Hasil dari ekstraksi inilah yang akan diproses pada multilayer perceptron (Alom *et al.*, 2018)

Metode CNN merupakan salah satu metode yang terdapat pada *deep learning*. Metode ini dapat melakukan proses pembelajaran mandiri agar dapat mengenali objek, mengekstraksi objek dan mengklasifikasi serta dapat diterapkan pada citra. Ilustrasi model dari CNN seperti yang terlihat pada Gambar 2.



Gambar 2. Model CNN secara Umum

Transfer learning, fungsinya pada pembelajaran mesin, adalah menggunakan hasil dari model yang telah dilatih sebelumnya pada permasalahan lain. Dalam *transfer learning*, mesin mengekstraksi pengetahuan yang diperoleh dari pembelajaran sebelumnya untuk meningkatkan generalisasi tentang hal lainnya. Misalnya, dalam melatih proses klasifikasi untuk mengenali apakah suatu citra berisi buah-buahan, maka dapat menggunakan hasil pembelajaran yang diperoleh selama pelatihan untuk mengenali sayuran dari sebuah citra lainnya. Ilustrasi *transfer learning* dapat juga dilihat pada Gambar 2. Namun, proses ekstraksi fitur dilakukan oleh model *pre-trained* CNN yang sudah ditentukan, yaitu MnetV2, VGG16, ResNet50 dan Xception.

2.2 Perhitungan Performa

Perhitungan performa dari proses pembelajaran yang akan dihitung adalah *precision*, *recall*, *accuracy* dan *F1 score*. Persamaan (1), (2), (3) dan (4) menunjukkan rumus perhitungan *precision*, *recall*, *accuracy* dan *F1 score* secara berurutan.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

$$F1\ score = \frac{Precision \times Recall}{Precision + Recall} \tag{4}$$

Penjelasan mengenai variabel *TP*, *FP*, *FN* dan *TN* ditunjukkan oleh *confusion matrix*, seperti yang terlihat pada Gambar 3. Pada gambar tersebut terdapat 4 kemungkinan yang dihasilkan. Kotak berwarna hijau mempresentasikan prediksi yang benar sedangkan kotak abu-abu mempresentasikan prediksi yang salah. Variabel *True Positive* (TP) menyatakan bahwa antara hasil prediksi dengan keadaan sebenarnya adalah benar. Variabel *True Negative* (TN) menyatakan menyatakan bahwa antara hasil prediksi dengan keadaan sebenarnya adalah salah. Variabel *False Positive* (FP) menyatakan bahwa hasil prediksi menyatakan benar sedangkan keadaan sebenarnya salah. Variabel *False Negative* (FN) menyatakan bahwa hasil prediksi menyatakan salah sedangkan keadaan sebenarnya benar.

		True/Actual Class	
		Positive(P)	Negative(N)
Predicted Class	True (T)	True Positive (TP)	False Positive (FP)
	False (F)	False Negative (FN)	True Negative (TN)
		$P=TP+FN$	$N=FP+TN$

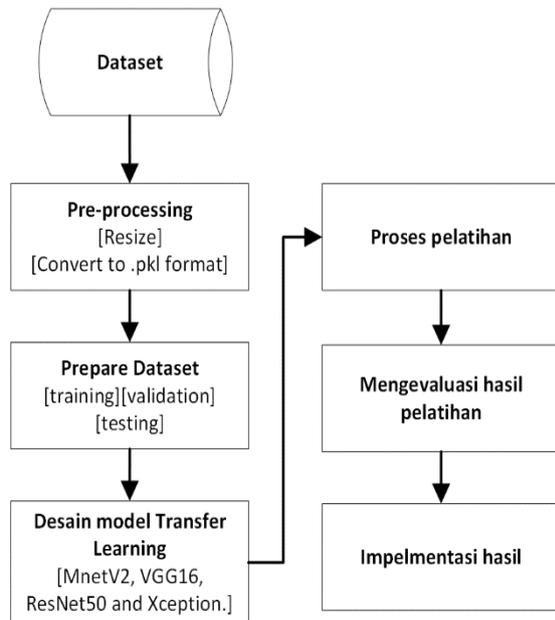
Gambar 3. Confusion Matrix

3. Metode

Pada bagian ini menjelaskan mengenai langkah-langkah yang dilakukan untuk mencapai tujuan penelitian. Penelitian ini melakukan langkah-langkah seperti yang ditunjukkan pada Gambar 4. Adapun langkah-langkah tersebut adalah:

1. Pengumpulan dataset
2. *Pre-processing*.
3. *Prepare* dataset.
4. Desain model *transfer learning*.

5. Proses pelatihan
6. Mengevaluasi hasil pelatihan
7. Implementasi hasil.



Gambar 4. Alur dari Metode Penelitian

3.1. Pengumpulan Dataset

Pada penelitian ini menggunakan dataset citra bahasa isyarat hijaiyah yang diambil dengan menggunakan kamera ponsel iPhone 5c. Huruf hijaiyah terdiri dari 31 huruf masing-masing terdiri dari 200 citra seperti yang terlihat pada Tabel 1.

Tabel 1. Citra Dataset Huruf Hijaiyah

No	Latin	Hijaiyah	No	Latin	Hijaiyah
1.	alif	ا	17.	zha	ظ
2.	ba	ب	18.	ain	ع
3.	ta	ت	19.	ghain	غ
4.	tsha	ث	20.	fa	ف
5.	jim	ج	21.	qaf	ق
6.	ha	ح	22.	kaf	ك
7.	kha	خ	23.	lam	ل
8.	dal	د	24.	mim	م
9.	dzal	ذ	25.	nun	ن
10.	ra	ر	26.	wau	و
11.	zay	ز	27.	Ha	ه
12.	sin	س	28.	ya	ي
13.	syin	ش	29.	ta marbuttha	ة
14.	shad	ص	30.	lam alif	لا
15.	dhad	ض	31.	alif lam	ال
16.	tha	ط			

3.2. Pre-processing

Resolusi citra yang diambil berukuran 400x400 piksel. Selanjutnya, dilakukan penyesuaian ukuran

menjadi berukuran 224x224. Hal ini dilakukan karena model pre-train yang akan digunakan menerima input pada ukuran tersebut.

Data citra ini disimpan dalam bentuk numpy array. Hal ini dilakukan agar pada proses pembelajaran tidak diperlukan lagi migrasi dari gambar ke numpy array. Ekstensi dari file tersebut adalah *.npy. Dataset dapat diakses pada tautan berikut <http://bit.ly/40sWUHW>.

3.3. Prepare Dataset

Pada tahapan ini, semua data citra yang digunakan memiliki ukuran 224x224 sesuai pada saat pengumpulan data citra. Total data citra yang dimiliki adalah 6200 citra yang terdiri dari 31 huruf hijaiyah yang masing-masing memiliki 200 citra.

Langkah selanjutnya membagi data citra menjadi 3 bagian yaitu data latih, data validasi dan data uji. Proses pembagian melalui 2 tahap yaitu semua data diacak terlebih dahulu. Selanjutnya, data dibagi menjadi data latih, data validasi dan data uji. Komposisi pembagian data terlihat pada Tabel 2.

Tabel 2. Pembagian Data Latih, Validasi dan Uji

No	Data	Presentase	Jumlah
1.	Data Latih	80%	4.960
2.	Data Validasi	10%	620
3.	Data Uji	10%	620

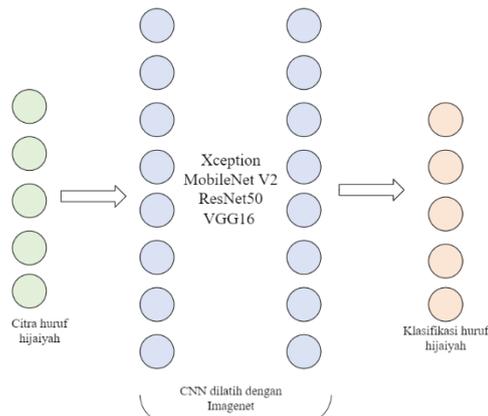
3.4. Desain model transfer learning

Pada tahapan ini akan dirancang model CNN yang akan digunakan dalam proses training untuk klasifikasi citra bahasa isyarat huruf hijaiyah. Data citra yang telah disiapkan akan menjadi masukkan untuk proses pelatihan menggunakan model CNN dengan teknik *transfer learning*. Ada 4 model *pre-trained* CNN yang digunakan pada penelitian ini yaitu MnetV2, VGG16, ResNet50 dan Xception.

Penelitian ini menggunakan metode transfer learning. Adapun metode yang digunakan adalah dengan tidak mengikut sertakan bagian output dari setiap model *pre-trained* CNN. Kemudian, bagian *output* tersebut digantikan *output* layer yang terdiri dari 31 *neuron* yang digunakan untuk proses klasifikasi dengan aktivasi softmax. Ilustrasi desain model CNN menggunakan *pre-trained* Imagenet terlihat pada Gambar 5.

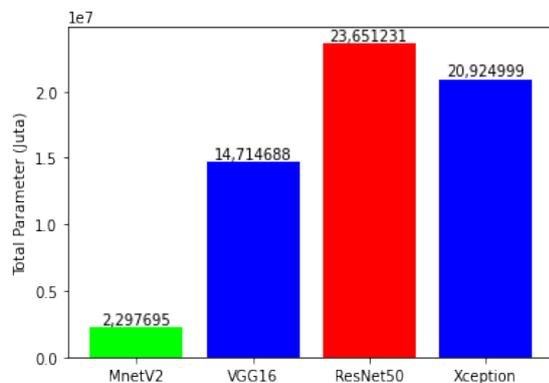
Jumlah parameter yang dilatih pada setiap model *pre-trained* CNN yang digunakan seperti terlihat pada Gambar 6. Pada grafik tersebut terlihat bahwa MnetV2 memiliki total parameter paling sedikit yaitu sebanyak 2.297.695 parameter. Total parameter tertinggi dimiliki oleh ResNet50 yaitu sebanyak 23.651.231 parameter. Jumlah parameter ini akan mempengaruhi lamanya proses pelatihan. Namun, jumlah parameter tidak dapat dijadikan patokan

untuk mendapatkan akurasi yang baik. Jumlah parameter yang lebih banyak belum tentu menghasilkan akurasi yang lebih baik dari yang lebih sedikit.



Gambar 5. Input dari Model Berupa Citra dari Huruf Hijaiyah

Kemudian, input dilatih oleh model *pre-trained* CNN yang pada bagian outputnya diganti dengan klasifikasi 31 huruf hijaiyah.



Gambar 6. Grafik dari Total Parameter yang Digunakan pada Setiap *Pre-Trained*

3.5. Proses Pelatihan

Pengaturan hyperparameter pada model CNN seperti yang terlihat pada Tabel 3. Penyesuaian *hyperparameter* pada proses pelatihan sebagai berikut: *optimizer* menggunakan Adam, *learning rate* 0,001, *loss function* menggunakan *categorical cross entropy*, sebanyak 100 kali, *step per epoch* 10 dan *batch size* diatur sebanyak 32. Selain itu, pembelajaran juga akan diberhentikan ketika sudah mencapai akurasi 0,998.

Tabel 3. *Hyperparameter* Model *Pre-Trained* CNN

No.	Hyperparameter	Nilai
1.	Optimizer	Adam
2.	Learning Rate	0,001
3.	Loss function	Categorical Cross Entropy
4.	Epoch	100
5.	Step per epoch	10
6.	Batch size	32

3.4. Evaluasi Hasil Pelatihan

Evaluasi hasil pelatihan menggunakan data citra huruf hijaiyah ini akan menunjukkan kinerja klasifikasi pada proses pelatihan, validasi dan pengujian. Adapun ukuran kinerja yang digunakan adalah *precision*, *recall*, *accuracy* dan *f1-score*. Pengujian ini akan dilakukan pada setiap model *pre-trained* CNN yang digunakan.

3.6. Implementasi

Pada tahapan terakhir, model yang dihasilkan dari proses pelatihan akan digunakan untuk mengenali bentuk tangan dari kamera secara langsung. Peraga akan membuat bentuk jari tangan seperti bahasa isyarat hijaiyah. Kemudian, sistem akan mencoba untuk mengenalinya.

4. Hasil dan Pembahasan

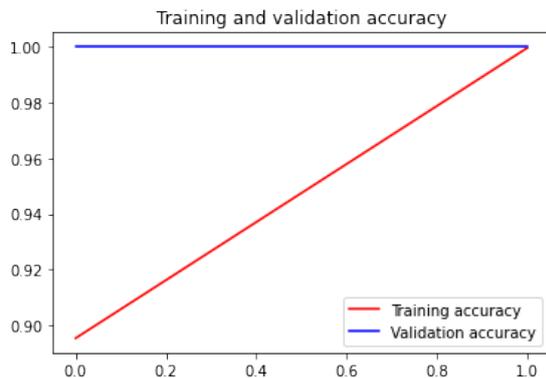
Pada bagian ini akan terbagi menjadi 3 pembahasan yaitu pembahasan hasil pelatihan dan validasi, hasil pengujian dan hasil implementasi. Untuk proses pelatihan, validasi dan pengujian latihan dilakukan dengan menggunakan Google Colab Pro yang disewa dengan harga 11,09\$ per bulan. Sedangkan untuk proses implementasi menggunakan laptop dengan spesifikasi prosesor Intel i5-11300H @ 3.10Ghz (8 CPU), RAM 16GB dan kartu grafis NVIDIA GeForce RTX 3050. Pada proses ini menggunakan latar belakang layar hijau sesuai dengan data gambar yang digunakan pada proses pelatihan, validasi dan uji.

4.1. Hasil Pelatihan dan Validasi

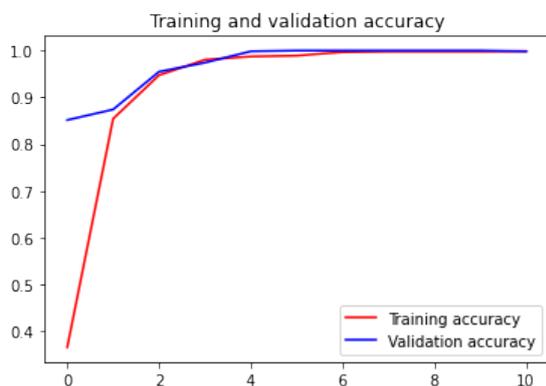
Proses pelatihan yang dilakukan menggunakan konsep transfer learning dari setiap *pre-trained* model CNN yaitu MnetV2, VGG16, ResNet50 dan Xception. Pada pelatihan ini dilakukan sebanyak 100 *epoch* dengan hyperparameter seperti yang terlihat pada Tabel 2. Namun, pelatihan akan dihentikan apabila nilai akurasi telah mencapai 0,998.

Hasil dari pelatihan ditunjukkan pada Gambar 7. Dari gambar tersebut terlihat bahwa untuk mencapai nilai akurasi 98.8% MnetV2 membutuhkan 2 *epoch*, disusul oleh Xception dengan 3, selanjutnya VGG16 dengan 11 *epoch* sedangkan ResNet50 gagal mencapai nilai akurasi 0,998 walaupun sudah melakukan 100 *epoch*. Nilai akurasi tertinggi yang dicapai oleh ResNet50 adalah 0,812. Secara umum akurasi yang dicapai ResNet50 memang menunjukkan nilai akurasi sekitar 0,8 (Shen & Savvides, 2020). ResNet50 memiliki jumlah parameter terbanyak antara model *pre-trained* yang digunakan, seperti yang terlihat pada Gambar 6. Jumlah parameter yang banyak menunjukkan bahwa model *pre-trained* CNN ResNet50 sangat kompleks. Selain itu, tidak semua model *pre-trained* memiliki nilai akurasi yang sama pada semua dataset. Ini

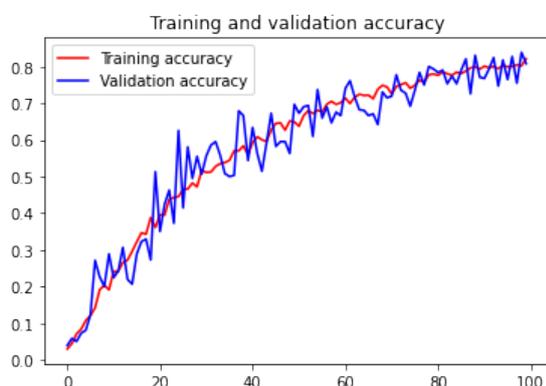
terjadi dikarenakan setiap model pre-trained belum tentu cocok pada setiap dataset tersebut.



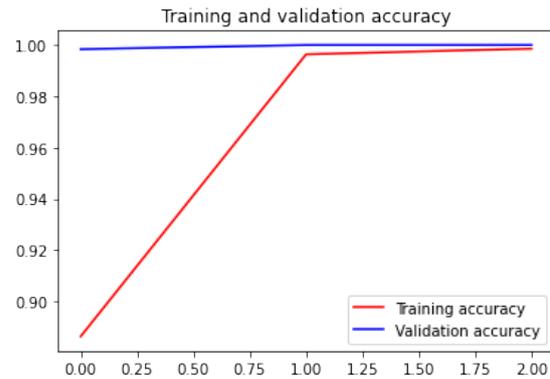
a. MnetV2 mencapai nilai akurasi 0,9996 dalam 1 epoch pada proses pelatihan. Nilai akurasi validasi juga langsung menunjukkan nilai akurasi 1 diawal proses validasi.



b. VGG16 mencapai akurasi 0,9988 dalam 11 epoch saat proses pelatihan. Pada awal proses validasi akurasi mencapai nilai akurasi 0,8515. Namun, setelah 4 epoch validasi mencapai nilai akurasi 0,9984.



c. ResNet50 mencapai nilai akurasi 0,8212 dalam 100 epoch pada proses pelatihan. Proses validasi menunjukkan nilai akurasi 0,8081. Selain itu, grafik juga menunjukkan terjadinya *overfitting* antara akurasi pelatihan dan validasi.



d. Xception mencapai nilai akurasi 0,9986 dalam 2 epoch pada proses pelatihan. Akurasi validasi juga langsung menunjukkan nilai akurasi 1 pada epoch ke-1.

Gambar 7. Grafik Pelatihan pada Setiap *Pre-Trained*

Hasil pelatihan dan validasi secara utuh ditunjukkan pada Tabel 4.

Tabel 4. Hasil Akurasi Pelatihan dan Validasi

No	Pre-trained Model	Latih	Validasi	Epoch
1	MnetV2	0,9996	1,0000	1
2	VGG16	0,9988	0,9984	10
3	ResNet50	0,8212	0,8081	100
4	Xception	0,9986	1,0000	2

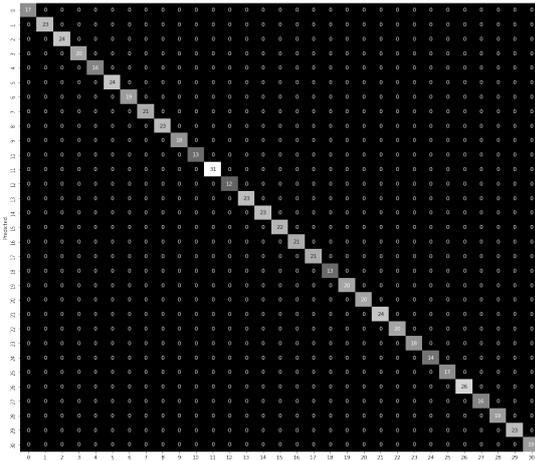
4.2. Hasil Pengujian

Pemeriksaan kinerja pada proses pelatihan menggunakan data validasi di setiap *epoch*. Namun, data uji merupakan data yang belum pernah dilihat *pre-trained* model selama proses pelatihan. Dengan kata lain, data ini dapat digunakan sebagai benchmark untuk implementasi sistem yang sebenarnya. Suatu hasil model pelatihan dapat bekerja dengan baik dalam eksperimen, tetapi tidak menunjukkan akurasi yang baik pada data yang belum pernah dilihat selama proses pelatihan. Ini menunjukkan betapa pentingnya suatu model diuji menggunakan data uji. Adapun pembagian kumpulan data seperti yang terlihat pada Tabel 2.

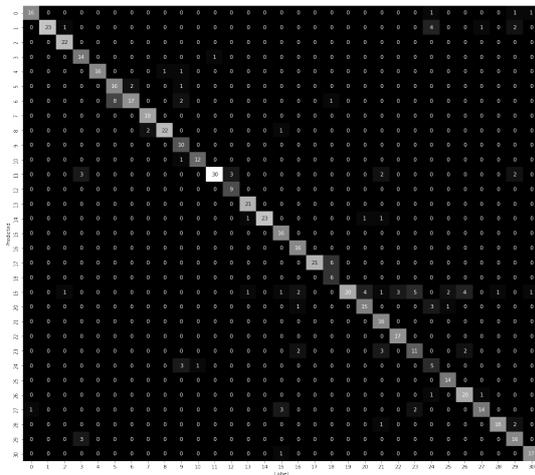
Hasil akurasi pengujian pada data uji ditunjukkan pada Tabel 5. Sedangkan *confusion matrix* terlihat pada Gambar 8.

Tabel 1. Hasil Akurasi Pengujian

Pre-Trained Model	Precision	Recall	F1-Score	Accuracy
MnetV2	1	1	1	1
VGG16	1	1	1	1
ResNet50	0,81	0,86	0,82	0,82
Xception	1	1	1	1



a. MnetV2, VGG16 dan Xception menunjukkan nilai akurasi 1 pada proses pengujian. Ini artinya ketiga pre-trained tersebut selalu tepat melakukan klasifikasi pada data uji.



b. ResNet50 masih melakukan kesalahan pada proses klasifikasi saat menggunakan data uji.

Gambar 8. Confusion Matrix pada Setiap Pre-Trained Model

Matrik confusion pada *pre-trained* model MnetV2, VGG16 dan Xception menunjukkan akurasi yang tepat. Artinya ketiga model ini tidak pernah salah dalam melakukan klasifikasi pada data uji. Sementara itu, *pre-trained* model RetNet50 masih melakukan kesalahan pada saat klasifikasi pada data uji. ResNet50 melakukan kesalahan klasifikasi sebanyak 108 data dan tepat mengklasifikasikan sebanyak 512 data.

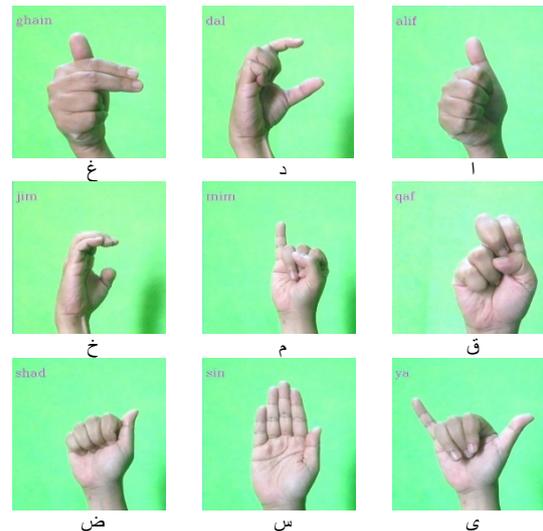
Pada Gambar 9 ditunjukkan beberapa data yang mengalami kesalahan klasifikasi. Terlihat pada gambar huruf syin diprediksi dengan huruf sin, huruf nun diprediksi dengan huruf ba, huruf ra diprediksi dengan huruf jim, huruf tsa diprediksi dengan huruf lam alif, huruf shad diprediksi dengan huruf dhad dan seterusnya.



Gambar 9. Contoh Data Salah Klasifikasi

4.3 Implementasi

Model hasil pembelajaran digunakan untuk proses pengenalan secara langsung bahasa isyarat melalui media video. Pada implementasi ini digunakan model dari hasil pembelajaran model MnetV2. Hasil menunjukkan proses pengenalan bahasa isyarat berlangsung dengan baik. Beberapa hasil uji coba terlihat pada Gambar 10.



Gambar 10. Implementasi Pengenalan Bahasa Isyarat

5. Kesimpulan

Penelitian ini menghasilkan sebuah dataset berupa citra huruf hijaiyah yang dapat digunakan pada penelitian berikutnya. Adapun hasil perbandingan model pre-trained CNN menunjukkan bahwa MnetV2, Xception, VGG16 berhasil mencapai batas akurasi yang diinginkan yaitu 99,85%, sedangkan ResNet50 tidak mencapai batas tersebut. Jumlah *epoch* yang dibutuhkan untuk mencapai batas akurasi tersebut MnetV2, Xception, VGG16 adalah 2, 3, 11 *epoch* secara berurutan. ResNet50 mencapai akurasi

82,12% pada *epoch* ke 100. Ini terjadi karena ResNet memang memiliki akurasi sekitar 80%.

Untuk proses pengujian menggunakan data uji menunjukkan bahwa MnetV2, Xception, VGG16 mendapatkan hasil 100% untuk *precision*, *recall*, *f1-score*, dan *accuracy*. Sedangkan, pada pengujian ResNet50 dihasilkan 81,55%, 86,04%, 82,04%, 82,58% untuk *precision*, *recall*, *f1-score*, dan *accuracy* secara berurutan. Hasil pembelajaran cukup menunjukkan kinerja yang baik saat proses implementasi pengenalan bahasa isyarat secara waktu nyata.

Adapun limitasi dari penelitian ini diantaranya adalah tidak melakukan proses *preprocessing* citra sebelum dilakukan pelatihan. Proses *preprocessing* citra yang dapat dilakukan diantaranya adalah segmentasi dan augmentasi. Selain itu, pemberi bahasa isyarat masih bersumber pada 1 orang saja. Variasi pemberi bahasa isyarat dapat dilakukan untuk menghindari terjadinya *overfitting*.

Daftar Pustaka

- Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., Van Esesn, B. C., Awwal, A. A. S., Asari, V. K., 2018. The history began from alexnet: a comprehensive survey on deep learning approaches. <http://arxiv.org/abs/1803.01164>
- Aly, W., Aly, S., Almotairi, S., 2019. User-independent american sign language alphabet recognition based on depth image and PCANet features. *IEEE Acces* 7, 123138–123150. <https://doi.org/10.1109/ACCESS.2019.2938829>
- Ameen, S., Vadera, S., 2017. A convolutional neural network to classify American Sign Language fingerspelling from depth and colour images. *Expert Systems* 34(3). <https://doi.org/https://doi.org/10.1111/exsy.12197>
- Ameur, S., Ben Khalifa, A., Bouhleb, M. S., 2020. A novel hybrid bidirectional unidirectional LSTM network for dynamic hand gesture recognition with Leap Motion. *Entertainment Computing* 35, 1–10. <https://doi.org/10.1016/j.entcom.2020.100373>
- Barbhuiya, A. A., Karsh, R. K., Jain, R., 2021. CNN based feature extraction and classification for sign language. *Multimedia Tools and Applications* 80(2), 3051–3069. <https://doi.org/10.1007/s11042-020-09829-y>
- Bin Makhshen, G. M., Luqman, H. A., El-Alfy, E.-S. M., 2019. Using gabor filter bank with downsampling and SVM for visual sign language alphabet recognition. 2nd smart cities symposium (SCS 2019) 15–21. <https://doi.org/10.1049/cp.2019.0188>
- Chevtchenko, S. F., Vale, R. F., Macario, V., 2018. Multi-objective optimization for hand posture recognition. *Expert Systems with Applications* 92, 170–181. <https://doi.org/10.1016/j.eswa.2017.09.046>
- Hosoe, H., Sako, S., Kwolek, B., 2017. Recognition of JSL finger spelling using convolutional neural networks. 2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA) 85–88. <https://doi.org/10.23919/MVA.2017.7986796>
- Islam, M. R., Mitu, U. K., Bhuiyan, R. A., Shin, J., 2018. Hand gesture feature extraction using deep convolutional neural network for recognizing American sign language. 2018 4th International Conference on Frontiers of Signal Processing (ICFSP) 115–119. <https://doi.org/10.1109/ICFSP.2018.8552044>
- Nurhayati, O. D., Eridani, D., Tsalavin, M. H., 2022. Sistem isyarat bahasa Indonesia (Sibi) metode convolutional neural network sequential secara real time. *Jurnal Teknologi Informasi Dan Ilmu Komputer* 9(4), 819–828. <https://doi.org/10.25126/jtiik.202294787>
- Prasetyo, E., Purbaningtyas, R., Adityo, R. D., Prabowo, E. T., Ferdiansyah, A. I., 2021. Perbandingan convolution neural network untuk klasifikasi kesegaran ikan bandeng pada citra mata. *Jurnal Teknologi Informasi Dan Ilmu Komputer* 8(3), 601–608. <https://doi.org/10.25126/jtiik.202184369>
- Rochmawanti, O., Utaminigrum, F., Bachtar, F. A., 2021. Analisis performa pre-trained model convolutional neural network dalam mendeteksi penyakit tuberculosis. *Jurnal Teknologi Informasi Dan Ilmu Komputer* 8(4), 805–814. <https://doi.org/10.25126/jtiik.202184441>
- Shen, Z., Savvides, M., 2020. Meal v2: boosting vanilla resnet-50 to 80%+ top-1 accuracy on imagenet without tricks. <http://arxiv.org/abs/2009.08453>
- Swastika, W., 2020. Studi awal deteksi covid-19 menggunakan citra ct berbasis deep learning. *Jurnal Teknologi Informasi dan Ilmu Komputer* 7(3), 629–634. <https://doi.org/10.25126/jtiik.202073399>
- Tao, W., Leu, M. C., Yin, Z., 2018. American sign language alphabet recognition using convolutional neural networks with multiview augmentation and inference fusion. *Engineering Applications of Artificial Intelligence* 76, 202–213. <https://doi.org/10.1016/j.engappai.2018.09.006>
- Yudistira, N., Widodo, A. W., Rahayudi, B., 2020. Deteksi covid-19 pada citra sinar-x dada menggunakan deep learning yang efisien. *Jurnal Teknologi Informasi dan Ilmu Komputer* 7(6), 1289–1296. <https://doi.org/10.25126/jtiik.202073651>