

MODEL PENERJEMAH BAHASA ISYARAT HURUF HIJAIYAH SECARA REALTIME MENGGUNAKAN CNN

Rif'at Chusnul Ma'fi, Wahyudi Agustiono

Sistem Informasi, Universitas Trunojoyo Madura

Jl. Raya Telang, Perumahan Telang Inda, Telang, Kamal, Kabupaten Bangkalan

Aat17agustus@gmail.com

ABSTRAK

Bahasa isyarat membantu dalam keterbatasan komunikasi pada penyandang tunarungu di Indonesia dengan bahasa isyarat indonesia (BISINDO)-nya, terlebih lagi pada umat muslim yang hanya bisa mengandalkan bahasa isyarat indonesia saja dalam memahami dan membaca Al Qur'an. Kesulitan didapat terlebih anak-anak yang belajar Al Qur'an dalam memahami huruf hijaiyah memiliki kesulitan belajar. Dengan berkembangnya teknologi akan memberikan solusi dalam hal tersebut dengan membuat pengenalan citra digital dalam bentuk pendeteksian dan pengenalan gerakan tangan. dengan memanfaatkan metode Convolutional Neural Network dapat mengklasifikasi bahasa isyarat pada huruf hijaiyah secara *realtime*. Hasil akhir pengujian beberapa parameter pada metode Convolutional Neural Network untuk klasifikasi citra menunjukkan performa yang relatif rendah, dengan akurasi sebesar 76.92%.

Kata kunci : Bahasa Isyarat, Hijaiyah, Deep Learning, Model Penerjemah , Convolutional Neural Network

1. PENDAHULUAN

Seiring dengan perkembangan teknologi yang semakin pesat, banyak inovasi yang dikembangkan untuk meningkatkan kualitas hidup penyandang disabilitas, termasuk mereka yang mengalami gangguan pendengaran atau tunarungu. Teknologi ini tidak hanya membantu dalam aspek komunikasi, tetapi juga memperluas akses mereka terhadap informasi dan kesempatan yang sebelumnya sulit dijangkau. Selain itu, teknologi seperti penerjemah bahasa isyarat dan transkripsi otomatis juga menjadi solusi efektif dalam mengatasi keterbatasan komunikasi sehari-hari [1].

Bahasa isyarat (*sign language*) sebagai bahasa yang berkembang secara alami yang digunakan di dalam komunitas tunarungu. Beberapa penyandang tunarungu menggunakan bahasa sehari-hari dengan bahasa isyarat Indonesia (BISINDO). Bahasa isyarat tidak hanya digunakan dalam kehidupan sehari-hari, tetapi juga diperlukan untuk memahami dan membaca Al Qur'an, termasuk untuk memahami huruf-huruf hijaiyah. Dengan kesulitan tersebut yang dirasakan penyandang tunarungu terlebih pada anak-anak yang merasa kesulitan memahami guru dengan mendengar belajar Al-Qur'an di lingkungan umum, karena keterbatasan pendengaran mereka [2].

Terdapat permasalahan antara guru pengajar yang sering menggunakan bahasa lisan dalam menjelaskan ke anak-anak tunarungu hal yang terjadi anak-anak tunarungu mengalami kesulitan untuk mengenali dan memahami terlebih pada huruf-huruf hijaiyah [2]. Seharusnya anak-anak mendapatkan hak yang sama, hal tersebut membatasi penyandang tunarungu dari pemahaman tentang ajaran agama islam. Hal tersebut yang menjadikan muslim yang tunarungu tidak memiliki kemampuan, misalnya, untuk mempelajari Al-Qur'an dan maknanya[3].

Dengan permasalahan utama adalah keterbatasan anak-anak tunarungu dalam memahami

huruf hijaiyah dan ajaran agama Islam melalui metode pembelajaran tradisional yang berbasis lisan [2]. Dengan berkembangnya teknologi saat ini dapat menyediakan solusi atas permasalahan dan membantu mengatasinya, salah satu bentuk solusi tersebut adalah pengenalan citra digital dalam bentuk pendeteksian dan pengenalan gerakan tangan yang dapat menampilkan informasi berupa bahasa isyarat yang nantinya dapat dikenali oleh penyandang tunarungu[4].

Terdapat banyak metode yang dapat digunakan untuk pengenalan citra digital, salah satunya yang populer adalah dengan menggunakan Artificial Intelligence (kecerdasan buatan) dalam bentuk Convolution neural network (CNN). *Convolutional Neural Network* (CNN) adalah salah satu representasi data dalam jaringan saraf tiruan *Artificial Neural Network* (ANN) yang dapat digunakan untuk mewakili sebuah objek yang terlihat secara visual spasial. *Convolutional Neural Network* (CNN) dapat digunakan untuk mengekstrak sebuah karakteristik/fitur dari sebuah gambar atau data visual lainnya dengan cara melatihnya untuk mengenali pola tertentu berdasarkan data visual yang diberikan. Salah satu penerapan CNN di dunia nyata adalah sistem pendeteksian pelanggaran lalu lintas yang dapat dipasang pada kamera CCTV pada lampu lalu lintas[5].

Adapun seperti penelitian sebelumnya tentang bahasa isyarat, seperti yang dilakukan oleh [6] menggunakan metode *Convolutional Neural Network* (CNN) untuk klasifikasi bahasa isyarat SIBI (Sistem Isyarat Bahasa Indonesia) dan memperoleh akurasi sebesar 90,05%. Penelitian lain oleh [7] menggunakan metode CNN untuk klasifikasi bahasa isyarat Amerika (ASL) dengan akurasi 73%.. Selain itu, pada penelitian oleh [8] menggunakan metode CNN untuk melakukan klasifikasi bahasa isyarat SIBI (Sistem Isyarat Bahasa

Indonesia) dan tingkat akurasi mencapai 54,8% dalam mengklasifikasi 30 bahasa isyarat.

Berdasarkan penelitian-penelitian terdahulu, banyak peneliti menggunakan metode *Convolutional Neural Network* (CNN) untuk pengenalan bahasa isyarat, terutama untuk bahasa isyarat sehari-hari. Oleh karena itu, penelitian ini bertujuan untuk menerapkan teknologi citra digital dengan menggunakan metode *Convolutional Neural Network* secara realtime pada bahasa isyarat huruf hijaiyah. Dengan cara pengambilan data visual dari gestur tangan yang kemudian memprosesnya dengan model arsitektur CNN sehingga dapat menghasilkan output informasi berupa huruf apa yang ditunjukkan dari gestur tangan tersebut.

2. TINJAUAN PUSTAKA

2.1. Bahasa Isyarat

Bahasa isyarat merupakan gerakan tangan yang melibatkan gerakan visual dan tanda-tanda. Bahasa isyarat digunakan sebagai sistem komunikasi untuk membantu komunitas penyandang tunarungu untuk interaksi sehari-hari. Bahasa isyarat meliputi penggunaan bagian tubuh yang berbeda, seperti jari, tangan, lengan, kepala, tubuh, dan ekspresi wajah [9].

2.2. Google Colaboratory

Sebuah produk dari Goggle yang disebut kolaboratif, atau hanya "Colab". Google Colab dapat menjalankan kode Python apa pun melalui browser web, biasa *digunakan* untuk analisis data, pembelajaran mesin, dan pengajaran. Google Colab adalah layanan notebook Jupyter yang menawarkan akses gratis ke kemampuan komputasi yang berat pada *graphic processing unit*(GPU) dan dapat digunakan tanpa konfigurasi. [10].

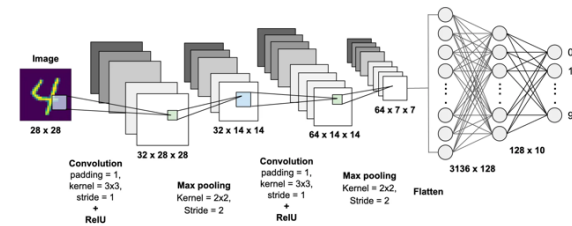
2.3. Deep Learning

Deep learning merupakan metode yang menggunakan algoritma dalam pembelajarannya dengan melewati data yang bertujuan meningkatkan akurasi dan efisiensi prediksi atau keputusan. *Deep learning* menggunakan jaringan saraf untuk mempelajari dataset. *Deep learning* memiliki fitur dalam menganalisis dan memproses data, banyak dampak yang signifikan dengan menggunakan teknologi ini di bidang seperti analisis pencitraan medis, pemrosesan bahasa alami, pengenalan bicara, dan mengemudi. *Deep learning* berpotensi untuk memungkinkan kita untuk memahami sistem yang kompleks, membuat keputusan yang tepat, dan menciptakan solusi yang lebih efisien dan efektif untuk masalah dunia nyata [11].

2.4. Convolution Neural Network (CNN)

Convolution neural network merupakan teknik deep learning dalam pengenalan objek, seperti menganalisa, memodelkan objek. *Convolution neural network* berguna dalam pencitraan dalam bidang medis, dalam mendeteksi hama tanaman di bidang

algikultur, dan lainnya. CNN memiliki keuntungan dalam hal melakukan secara otomatis dan tanpa pengawasan manusia yang membuat teknik ini populer dari pendahulunya [5].

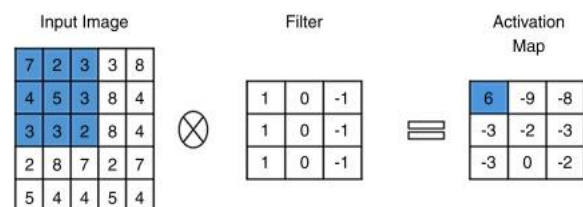


Gambar 1. Arsitektur *convolutional neural network*

Pada Gambar 1. Terdapat ilustrasi arsitektur metode *Convolutional Neural Network* terdapat tiga bagian, yaitu *input*, *feature extraction* dan *classification*. Pada bagian pertama *layer feature extraction* memiliki beberapa layer antara lain:

2.5. Convolutional layer

Lapisan *convolutional layer* terdiri dari neuron-neuron sehingga membentuk filter dengan panjang dan lebar dalam satuan piksel. Proses konvolusi dilakukan dengan menggunakan kernel dan stride, Proses konvolusi terjadi dengan melakukan pergeseran filter sehingga mencakup keseluruhan bagian citra. Jumlah pergeseran filter disebut juga dengan *stride*. Konvolusi dalam pengolahan gambar berarti menambahkan kernel (kotak biru) ke gambar di seluruh offset yang memungkinkan, seperti yang ditunjukkan pada gambar 2 di bawah ini [12].



Gambar 2. Perhitungan konvolusi

Dalam menghitung dimensi dari *feature map* digunakan rumus sebagai berikut [6].

$$\text{Output feature map} = \frac{W - N + 2P}{S} + 1 \quad (1)$$

Keterangan:

W = ukuran input N = Ukuran filter

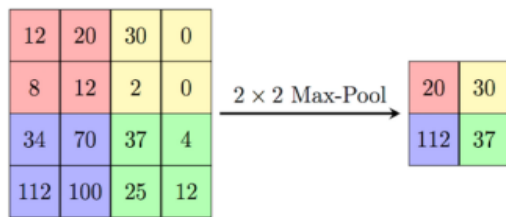
P = Padding S = Stride

Fungsi aktivasi ReLu akan digunakan untuk menormalisasi hasil *feature map*, dengan rumus berikut:

$$\text{Relu}(x) = \max(0, x) \quad (2)$$

2.6. MaxPooling layer

Pada gambar 3 merupakan ilustrasi lapisan *maxpooling* untuk mengurangi dimensi matriks dari hasil konvolusi. Dengan cara tersebut mempercepat proses komputasi. [6].

Gambar 3. Ilustrasi *max pooling*

Layer classification memiliki beberapa lapisan antara lain:

a. *Flatten*

Merupakan proses merubah matriks hasil *feature learning* menjadi matriks 2D menjadi 1D [6].

b. *Dropout*

Merupakan proses menghilangkan neuron yang memiliki noise secara acak untuk mencegah overfitting [6].

c. *Fully Connected Layer*

Proses menyatukan seluruh neuron menjadi satu dimensi dari data proses *feature learning* yang telah diubah menjadi vektor melalui *flattening* [6].

d. *Softmax*

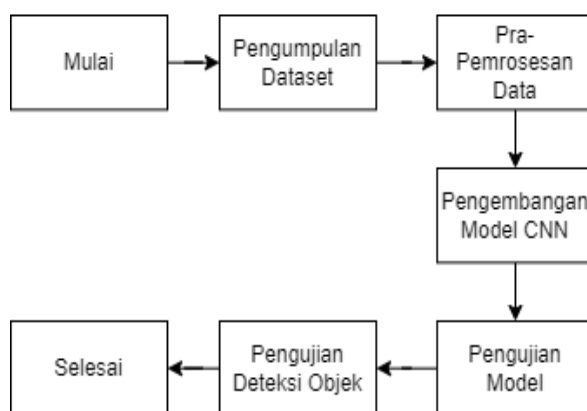
Merupakan proses menghitung nilai probabilitas pada setiap kelas untuk semua kelas target. Nilai probabilitas yang diperoleh dari hasil perhitungan softmax berkisar antara 0 dan 1[6].

2.7. Evaluasi Model

Evaluasi model atau evaluasi performa model dilakukan setelah suatu model telah dilatih dengan evaluasi model. Evaluasi model menggunakan data uji yang sudah dipersiapkan untuk mengetahui performa dari model tertentu terhadap citra di luar dataset latih. Evaluasi model menggunakan confusion matrix dalam mengukur performa model [13].

3. METODE PENELITIAN

Dalam alur penelitian ini, penulis menjelaskan segala sesuatu yang telah dan akan dilakukan selama penelitian. Berikut alur penelitian pada penelitian ini.



Gambar 4. Diagram alur penelitian

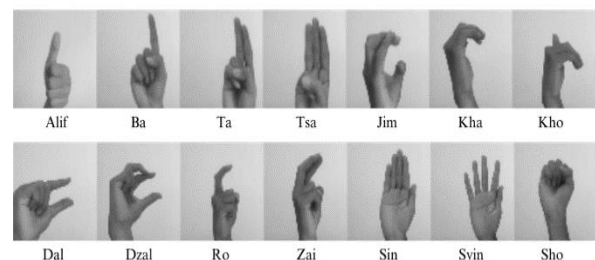
Pada gambar 4 merupakan gambaran alur penelitian dengan beberapa tahapan yang dilakukan

pada penelitian ini dan diketahui bahwa penelitian ini melewati lima tahapan. Berikut langkah -langkah tahapan penelitian:

3.1. Pengumpulan Dataset

Pada tahapan pertama dilakukan Pengumpulan data, data yang digunakan penelitian ini adalah citra gestur tangan, karena keterbatasan dataset peneliti melakukan pengumpulan data dengan standar bahasa isyarat hijaiyah dari lembaga Lajnah Pentashihan Mushaf Al-Qur'an, Badan Litbang dan Diklat Kementerian Agama RI [14].

Berdasarkan hal diatas peneliti mengumpulkan dataset menggunakan kamera smartphone sebagai alat pengambilan data. Dengan menggunakan kode python dalam mengumpulkan data diperoleh dataset gestur tangan yang memiliki total gambar 938 dengan 30 kelas huruf hijaiyah dan 1 kelas kosong. Berikut beberapa contoh tampilan citra gestur tangan huruf hijaiyah pada dataset yang digunakan:



Gambar 5. Foto gestur tangan

Pada gambar 5 dataset dipotret menggunakan kamera smartphone dengan resolusi kamera 48 megapixel. Dengan Jenis dataset berupa gambar dengan ekstensi file .jpg dengan resolusi 64x64 piksel.

3.2. Pra-Pemrosesan Data

Pada tahapan kedua dilakukan pengolahan data (*Pre-processing*). Tahapan ini memiliki tiga tahapan yaitu: pembagian dataset, augmentasi dan normalisasi data. Pada umumnya pembagian dataset dibagi menjadi 80% data training, 10% validasi, dan 10% data uji [15].

Tabel 1 Pembagian Data

Total Gambar	Data Latih (80%)	Data Validasi (10%)	Data Uji(10%)
938	746	101	91

Pada tabel 1 menggambarkan skenario pembagian dataset yang akan digunakan. Dari total 938 gambar, 743 akan digunakan untuk data latihan (training), 101 akan digunakan untuk data validasi (validation), dan 101 akan digunakan untuk data uji (test). Kemudian dilakukan proses augmentasi pada data training seperti gambar di bawah ini.



Gambar 6. Hasil augmentasi

Pada gambar 6 gambar telah melalui proses *zoom in* dan *zoom out* hingga 20%, kemudian dilakukan pencerahan gambar rentang 0.8 sampai 1 untuk mengatur pencahayaan dan proses terakhir menggunakan *fill mode* untuk mengisi *pixel* yang hilang setelah augmentasi. Setelah augmentasi, normalisasi dilakukan pada dataset yang sudah dibagi. Proses ini akan menormalisasi data menggunakan library keras *ImageDataGenerator*. Ini juga memberikan label pada data. Untuk meningkatkan hasil *learning*, semua nilai piksel dinormalisasi menjadi 0 hingga 1.

3.3. Implementasi Model CNN

Dalam pemilihan model peneliti menggunakan CNN model yang dibangun dan dijalankan menggunakan *Python programming language*, serta Keras *libraries* dan TensorFlow. TensorFlow adalah *framework* untuk membuat model machine learning, khususnya neural networks, dan dibuat oleh Google pada tahun 2015 [16]. Untuk Keras memungkinkan pengguna untuk melatih dan menganalisis neural network dan berfungsi dengan TensorFlow [17].

Berikut lapisan model yang akan dijabarkan pada tabel sebagai berikut:

Tabel 2. Tabel Arsitektur CNN

Type Layer	Output Shape	Keterangan
conv2d	(None, 62, 62, 64)	Filter size= 3x3 Act = ReLu
max_pooling2d	(None, 31, 31, 64)	Pool_Size = 2x2
conv2d_1	(None, 29, 29, 128)	Filter size= 3x3 Act = ReLu
max_pooling2d_1	(None, 14, 14, 128)	Pool_Size = 2x2
flatten	(None, 25088)	-
dense	(None, 128)	Act = ReLu
dropout	(None, 128)	Dropout_Rate = 0.5
dense_1	(None, 31)	Act = Softmax

Pada tabel 2 dijelaskan lapisan pertama, data pelatihan akan digunakan sebagai data masukan. Selanjutnya, proses konvolusi pertama akan dilakukan dengan menggunakan fungsi aktivasi ReLU dan max pooling. Kemudian luaran hasil layer pertama akan digunakan sebagai input untuk konvolusi kedua. Setelah itu, hasil konvolusi akan melewati proses flattening, yang menghasilkan vektor satu baris. Proses flattening ini akan diterima pada lapisan yang terhubung sepenuhnya dengan aktivasi ReLU, yang membuat semua neuron pada lapisan tersebut terhubung secara menyeluruh. Kemudian pada lapisan *dropout* akan dilakukan pembuangan neuron-neuron yang tidak terpakai untuk mengatasi *overfitting*[18]. Hasil semua proses tadi akan dikumpulkan ke dalam lapisan *fully-connected* akhir. Tahap ini akan mengidentifikasi fitur yang terkait dengan kelas tertentu. Selanjutnya, hasil akhir dari proses adalah fitur yang dikategorikan dalam 31 kelas.

Setelah pembuatan model arsitektur, Berikutnya masuk dalam proses training dengan menggunakan acuan nilai parameter yang ditunjukkan pada tabel 2.

Tabel 3. Parameter Training

Parameter	Nilai
Epoch	100
Batch size	32
Learning Rate	0.001
Loss function	Categorical Cross Entropy
Optimizer	Adam
Optimizer metrics	Accuracy

Pada tabel 3 ada beberapa parameter yang digunakan pada implementasi model. *Epoch* digunakan ketika semua dataset yang telah melewati proses training akan diulang kembali ke awal untuk sekali putaran. Kemudian *batch size* merupakan jumlah sampel data yang disebarkan ke dalam *neural network*. Selanjutnya *learning rate* yang menentukan seberapa besar langkah yang diambil dalam setiap iterasi ketika menyesuaikan nilai parameter. Untuk *loss function* mengukur seberapa buruk atau bagus performa model dalam memprediksi output yang benar dari *input* yang diberikan. *Optimizer* Adam adalah Optimasi yang digunakan untuk meningkatkan akurasi dari model yang sudah dibuat. *Optimizer metrics* merupakan daftar parameter yang diteruskan ke model Keras. *Metrics* berfungsi untuk mengevaluasi kinerja model pembelajaran mendalam [19].

3.4. Evaluasi Model

Evaluasi model dilakukan untuk mengukur performa model CNN dalam mengklasifikasikan citra yang terprediksi dengan benar dari keseluruhan data uji. Pada evaluasi ini menggunakan pustaka *numpy* berfungsi memanipulasi data umum dan melakukan banyak tugas dengan Python [20]. *Skylearn* pustaka yang berfungsi untuk membangun model machine learning, seperti regresi, klasifikasi, dll dan *Matplotlib* untuk memvisualisasikan data dalam 2D dan 3D [21]. Beberapa parameter dikaitkan dalam evaluasi kinerja

model yang menggunakan metode confusion matrix [15].

a. *True Positive (TP)*

Nilai yang diprediksi oleh model bahwa data prediksi berada dalam kelas positif dan data sebenarnya berada dalam kelas positif.

b. *False Positive (FP)*

Nilai yang diprediksi oleh model bahwa data prediksi berada dalam kelas negatif tetapi data sebenarnya berada dalam kelas positif.

c. *False Negative*

Nilai yang diprediksi oleh model bahwa data prediksi berada dalam kelas negatif tetapi data sebenarnya berada dalam kelas positif.

d. *Accuracy*

Akurasi dapat dihitung dengan membandingkan jumlah klasifikasi yang benar dengan jumlah klasifikasi total. Klasifikasi yang tepat dianggap sebagai Positif Asli untuk setiap jenis atau kelas. Rumus matematis *accuracy* dapat dilihat pada persamaan (3).

$$Accuracy = \frac{TP}{Total\ Seluruh\ Data} \quad (3)$$

e. *Precision*

Precision mengukur akurasi prediksi positif model terhadap prediksi kelas tertentu disebut *precision*, dan rumus matematis dapat dilihat dalam persamaan (4).

$$Precision = \frac{TP}{(TP + FP)} \quad (4)$$

f. *Recall*

Recall mengukur rasio prediksi *true positif* dibandingkan dengan keseluruhan data *true positif* dengan nilai mendekati 1. Penulisan kembali matematis, seperti yang ditunjukkan dalam persamaan (5).

$$Recall = \frac{TP}{(TP + FN)} \quad (5)$$

g. *F1-Score*

Merupakan rata-rata dari perhitungan *precision* dan *recall* dan perhitungan kombinasi nilai untuk dijadikan sebagai nilai pengukuran. Berikut penulisan matematis pada persamaan (6).

$$F1 - Score = 2 \times \frac{precision \times recall}{recall + precision} \quad (6)$$

3.5. Pengujian Deteksi Objek

Pada tahapan akhir pengujian deteksi objek dengan melakukan pengambilan citra secara terus menerus (*real time*) untuk melakukan pendeteksian objek dan akan diimplementasikan melalui bantuan bahasa pemrograman python [12]. Model yang sudah dibuat akan digunakan dalam menghasilkan objek deteksi dengan bantuan *TensorFlow* dan akan menampilkan akurasi yang didapat pada tangkapan gestur tangan.

4. HASIL DAN PEMBAHASAN

4.1. Hasil Pengujian Model Training Data

Proses pelatihan akan menggunakan dataset yang sudah melalui proses *preprocessing* untuk menghasilkan model klasifikasi. Dataset yang memiliki 31 kelas dan memiliki jumlah 938 data citra digital yang sudah melewati proses *preprocessing*. Kemudian dilakukan proses training sampai nilai akurasi yang diperoleh cukup baik agar model dikatakan cukup optimal. Berikut hasil nilai training model yang didapat dari setiap *epoch* ditampilkan pada tabel 4.

Tabel 4. Hasil Training Model

Epoch ke-	Accuracy	Loss	Val_acc	Val_Loss
20	0.4062	1.9225	0.3960	2.1801
40	0.4688	1.2941	0.6535	1.5177
60	0.7188	0.9762	0.7228	1.3641
80	0.7812	0.4622	0.7030	1.3781
100	0.9375	0.2292	0.7426	1.4815

Keterangan:

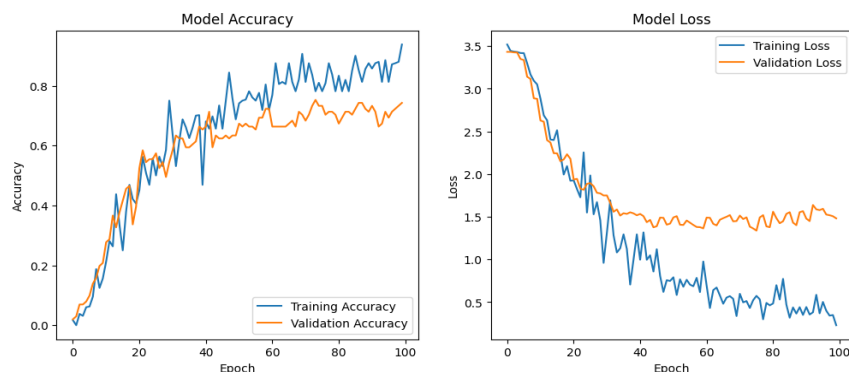
Accuracy: nilai akurasi dari training dataset latih.

Loss: nilai *loss function* dari dataset latih.

Val_acc: nilai akurasi dari *validation* dataset.

Val_Loss: nilai *loss function* dari *validation* dataset.

Berdasarkan Tabel 4, dapat disimpulkan bahwa data validasi mengalami peningkatan akurasi yang signifikan, mencapai 65,35% pada epoch ke-40, sedangkan data latih mencapai akurasi 71,88% pada epoch ke-60. Selain itu, nilai *loss* pada data validasi menurun secara drastis pada epoch ke-40, dengan nilai 1,5177, dan nilai *loss* pada data latih mencapai 1,2941 pada epoch yang sama. Untuk tampilan grafik ditunjukkan pada gambar dibawah.

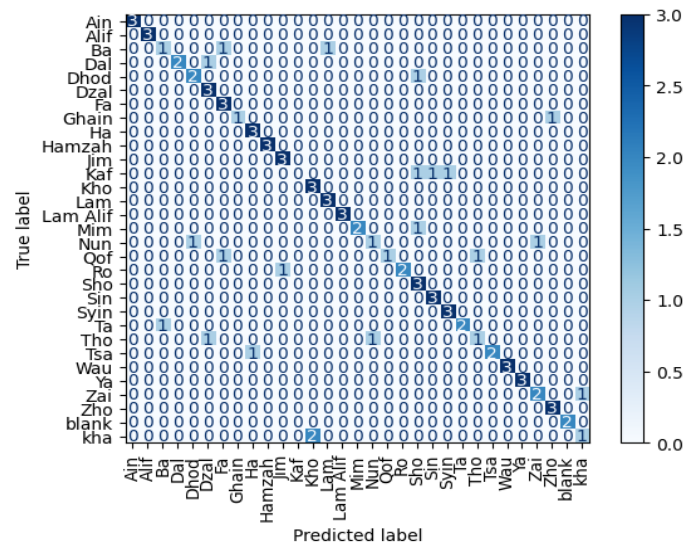


Gambar 7. Grafik akurasi hasil training model

Gambar 7 menunjukkan hasil dari proses training, bahwa model optimal ditandai dengan akurasi yang meningkat dan loss yang menurun. Loss berguna mengukur seberapa baik prediksi dengan mengukur jarak antara hasil output prediksi dan nilai target. Hasil menunjukkan pelatihan data train didapat akurasi sebesar 93.75% dan tahap pelatihan data validasi menghasilkan akurasi sebesar 74,26%.

4.2. Pengujian Evaluasi Model

Dalam pengujian evaluasi model dilakukan untuk melihat hasil akurasi yang diperoleh dari beberapa kelas yang nantinya dengan melakukan evaluasi dan prediksi menggunakan data uji untuk mengetahui keakuratannya. selanjutnya, akurasi model *Convolutional Neural Network* dalam bentuk tabel *confusion matrix*, hasil dapat dilihat dibawah ini;



Gambar 8. Hasil *Confusion Matrix*

Pada gambar 8 merupakan hasil proses confusion matrix terlihat beberapa kelas cenderung melakukan prediksi dengan benar karena elemen diagonalnya sangat terlihat. Warna biru tua pada diagonal menunjukkan prediksi yang bernilai tinggi, sedangkan warna lebih muda diluar diagonal menunjukkan kesalahan prediksi yang rendah. Hasil performa model akan ditampilkan pada tabel dengan nilai dari tingkat akurasi model, *precision* dan *recall* dalam mengklasifikasikan keseluruhan sampel data.

Tabel 5. Hasil performa model

Performa	Hasil
Accuracy	76.92%
Precision	79.56%
Recall	78,57%
F1-Score	76,35%

Hasil pada performa *accuracy* menunjukkan bahwa model memprediksi dari 91 gambar pada tiap kelas, 70 dapat diidentifikasi true positif sebagai gambar gestur tangan oleh klasifikasi *Convolutional Neural Network* (CNN). Dalam menghitung tingkat akurasi CNN untuk mengidentifikasi gambar dengan rumus pada persamaan (3).

$$Akurasi = \frac{\text{Jumlah Prediksi Benar}}{\text{Jumlah Total Prediksi}} \times 100\%$$

$$Akurasi = \frac{70}{91} \times 100\% = 76.92\%$$

Pada performa *precision* perhitungan untuk mengukur seberapa banyak prediksi positif yang benar dibandingkan dengan total prediksi positif. Berikut rumus pada persamaan (4) dengan penambahan rumus *weighted average*.

$$Precisio = \frac{\sum_{i=1}^n (\text{Precision}_i \times \text{Data uji}_i)}{\sum_{i=1}^n (\text{Data uji}_i)} \times 100\%$$

$$Precisio = \frac{72,40}{91} \times 100\% = 79.56\%$$

Hasil menunjukkan seberapa akurat model dalam memprediksi label positif dengan hasil *precision* yang sebesar 79.56%. Model masih membuat sekitar 20.44% kesalahan dalam memprediksi suatu kelas, yang artinya ada sejumlah *false positive*.

Pada performa *recall* perhitungan untuk mengukur seberapa banyak prediksi positif yang benar dibandingkan dengan total kasus positif yang sebenarnya. Berikut perhitungan dengan rumus pada persamaan (5) dengan penambahan rumus *weighted average*

$$Recall = \frac{\sum_{i=1}^n (\text{Recall}_i \times \text{Data uji}_i)}{\sum_{i=1}^n (\text{Data uji}_i)} \times 100\%$$

$$Recall = \frac{71,50}{91} \times 100\% = 78.57\%$$

Hasil menunjukkan nilai yang sama seperti *precision*. Dengan *recall* sebesar 78.57% yang benar-benar positif di dataset, model berhasil menemukan

sekitar 78.57% di antaranya dan model gagal mendeteksi sekitar 21,43% dari kasus positif *actual*, atau masih ada *false negative*.

Dalam performa *f1-score* dengan rata-rata dari *precision* dan *recall* untuk menghitung tingkat akurasi CNN dalam mengidentifikasi gambar dengan rumus pada persamaan (6) dengan rumus *weighted average*.

$$F1 - Score = 2 \times \frac{\sum_{i=1}^n (F1_i \times Data\ uji_i)}{\sum_{i=1}^n (Data\ uji_i)} \times 100\%$$

$$F1 - Score = 2 \times \frac{69.48}{91} = 76.35\%$$

Hasil akurasi akhir metode *Convolutional Neural Network* (CNN) menunjukkan dalam pengklasifikasian bahasa isyarat hijaiyah sebesar 76.35%. Meskipun hasil ini menunjukkan bahwa model dapat mengenali beberapa gestur dengan baik, terdapat faktor mungkin berkontribusi pada akurasi yang tidak mencapai 80%. Di antaranya adalah keterbatasan jumlah dataset yang hanya 938 citra, yang berpotensi menyebabkan *overfitting* dan mengurangi kemampuan generalisasi model terhadap data baru.

4.3. Pengujian Deteksi Objek



Gambar 7. Hasil akurasi beberapa kelas

Pada gambar 7 merupakan hasil luaran pengujian pada citra bahasa isyarat beberapa kelas sesuai dengan kelasnya dengan probabilitas prediksi beberapa class memiliki nilai akurasi diatas 50% dan beberapa huruf tidak konsisten dalam performa akurasi.

5. KESIMPULAN DAN SARAN

Berdasarkan hasil dari penelitian yang telah dilakukan, metode *Convolutional Neural Network* dapat melakukan klasifikasi terhadap objek yang menunjukkan kinerja yang baik dengan hasil proses latih model mendapatkan akurasi yang baik sebesar 93.75% pada model dan pada pengujian evaluasi model untuk performa klasifikasi citra bahasa isyarat hijaiyah yaitu hanya dengan rata-rata akurasi 76.92%, *precision* 79,56%, *recall* 78,57%, dan *f1-score* 76.35%. Pada pengujian deteksi objek beberapa huruf tidak konsisten dalam akurasi deteksi. Untuk penelitian selanjutnya, disarankan untuk

meningkatkan jumlah data pelatihan dan kualitas gambar dengan *background* yang beragam, menambahkan teknik augmentasi yang lebih agresif, seperti rotasi dan *flipping*, serta mengeksplorasi arsitektur model lain agar lebih baik.

DAFTAR PUSTAKA

- [1] A. Núñez-Marcos, O. Perez-de-Viñaspre, and G. Labaka, "A survey on Sign Language machine translation," *Expert Syst. Appl.*, vol. 213, p. 118993, 2023, doi: <https://doi.org/10.1016/j.eswa.2022.118993>.
- [2] N. Nurdyansyah and N. Pujiati, "Penerapan Isyarat Huruf Hijaiyyah dalam meningkatkan kemampuan membaca Al-Qur'an bagi anak tunarungu," *LITERAL Disabil. Stud.*, vol. 1, no. 01, pp. 32–44, 2023, doi: 10.62385/literal.v1i01.25.
- [3] G. Tharwat, A. M. Ahmed, and B. Bouallegue, "Arabic Sign Language Recognition System for Alphabets Using Machine Learning Techniques," *J. Electr. Comput. Eng.*, vol. 2021, no. 1, p. 2995851, 2021, doi: <https://doi.org/10.1155/2021/2995851>.
- [4] A. P. Saputri, A. Taqwa, and S. Soim, "Analisis Deteksi Objek Citra Digital Menggunakan Algoritma Yolo dan CNN Dengan Arsitektur REPVG pada Sistem Pendeteksian dan Pengenalan Ekspresi Wajah," *Syntax Lit. J. Ilm. Indones.*, vol. 7, no. 9, pp. 13068–13080, 2022.
- [5] M. M. Taye, "Theoretical Understanding of Convolutional Neural Network: Concepts, Architectures, Applications, Future Directions," *Computation*, vol. 11, no. 3, 2023, doi: 10.3390/computation11030052.
- [6] M. Sholawati, K. Auliasari, and F. Ariwibisono, "Pengembangan Aplikasi Pengenalan Bahasa Isyarat Abjad Sibi Menggunakan Metode Convolutional Neural Network (Cnn)," *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 6, no. 1, pp. 134–144, 2022, doi: 10.36040/jati.v6i1.4507.
- [7] R. H. Alfikri, M. S. Utomo, H. Februariyanti, and E. Nurwahyudi, "Pembangunan Aplikasi Penerjemah Bahasa Isyarat Dengan Metode Cnn Berbasis Android," *J. Teknoinfo*, vol. 16, no. 2, p. 183, 2022, doi: 10.33365/jti.v16i2.1752.
- [8] Nasha Hikmatia A.E. and M. I. Zul, "Aplikasi Penerjemah Bahasa Isyarat Indonesia menjadi Suara berbasis Android menggunakan Tensorflow," *J. Komput. Terap.*, vol. 7, no. 1, pp. 74–83, 2021, doi: 10.35143/jkt.v7i1.4629.
- [9] R. Rastgoo, K. Kiani, and S. Escalera, "Sign Language Recognition: A Deep Survey," *Expert Syst. Appl.*, vol. 164, p. 113794, 2021, doi: <https://doi.org/10.1016/j.eswa.2020.113794>.
- [10] G. I. E. Soen, M. Marlina, and R. Renny, "Implementasi Cloud Computing dengan Google Colaboratory pada Aplikasi Pengolah Data Zoom Participants," *JITU J. Inform. Technol. Commun.*, vol. 6, no. 1, pp. 24–30, 2022.

- [11] K. Sharifani and M. Amini, "Machine learning and deep learning: A review of methods and applications," *World Inf. Technol. Eng. J.*, vol. 10, no. 07, pp. 3897–3904, 2023.
- [12] K. Azmi and S. Defit, "Implementasi Convolutional Neural Network (CNN) Untuk Klasifikasi Batik Tanah Liat Sumatera Barat," vol. 16, no. 1, pp. 2580–2582, 2023.
- [13] Kade Bramasta Vikana Putra, I Putu Agung Bayupati, and Dewa Made Sri Arsa, "Klasifikasi Citra Daging Menggunakan Deep Learning dengan Optimisasi Hard Voting," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 4, pp. 656–662, 2021, doi: 10.29207/resti.v5i4.3247.
- [14] A. Manap, "Mengenal Al-Qur'an dengan Bahasa Isyarat, Legacy Kemenag untuk Sahabat Disabilitas," *Jabar NU*, 2023. <https://jabar.nu.or.id/nasional/mengenal-al-quran-dengan-bahasa-isyarat-legacy-kemenag-untuk-sahabat-disabilitas-RnMkC> (accessed Sep. 01, 2024).
- [15] E. Christianto, AlwinM.Sambu, and Feisy D. Kambey, "Implementation of Convolutional Neural Network on Images for Starlings Classification," *J. Tek. Inform.*, 2021, [Online]. Available: <https://ejournal.unsrat.ac.id/index.php/informatika>
- [16] J. R. S. Iruela, L. G. B. Ruiz, M. I. Capel, and M. C. Pegalajar, "A TensorFlow Approach to Data Analysis for Time Series Forecasting in the Energy-Efficiency Realm," *Energies*, vol. 14, no. 13, 2021, doi: 10.3390/en14134038.
- [17] B. T. Chicho and A. B. Sallow, "A Comprehensive Survey of Deep Learning Models Based on Keras Framework," *J. Soft Comput. Data Min.*, vol. 2, no. 2, pp. 49–62, 2021, doi: 10.30880/jscdm.2021.02.02.005.
- [18] H. D. P. Alfendio Alif Faudisyah, Kristoko Dwi Hartomo, "View of Deteksi Cacat pada Isolasi Trafo Secara Visual menggunakan Algoritma Convolutional Neural Network (CNN).pdf," *Jurnal JTIK*, 2023.
- [19] D. Armiady and I. Muslem R, "Klasifikasi Kualitas Buah Pisang Berdasarkan Citra Buah Menggunakan Stochastic Gradient Descent," *KLIK Kaji. Ilm. Inform. dan Komput.*, vol. 4, no. 2, pp. 1207–1215, 2023, doi: 10.30865/klik.v4i2.1243.
- [20] I. N. T. A. Putra, K. S. Kartini, Y. K. Suyitno, I. M. Sugiarta, and N. K. E. Puspita, "Penerapan Library Tensorflow, Cvzone, dan Numpy pada Sistem Deteksi Bahasa Isyarat Secara Real Time," *J. Krisnadana*, vol. 2, no. 3, pp. 412–423, 2023, doi: 10.58982/krisnadana.v2i3.335.
- [21] C. Haryanto, N. Rahaningsih, and F. Muhammad Basysyar, "Komparasi Algoritma Machine Learning Dalam Memprediksi Harga Rumah," *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 7, no. 1, pp. 533–539, 2023, doi: 10.36040/jati.v7i1.6343.