PENGENALAN BAHASA ISYARAT MENGGUNAKAN DETEKSI OBJEK DEEP LEARNING

David Arian Virgiawan[1], Fachrim Irhamna Rahman[2], Rizki Yusliana Bakti[3]
[1,2,3]*Informatika, Universitas Muhammadiyah Makassar, Makassar, 90221, Indonesia*
105841107920@student.unismuh.ac.id,

Abstract

Based on technological developments, especially in the field of computing, it is increasingly possible to develop systems that are able to detect sign language more efficiently. One of the main problems faced is how to detect and classify sign language movements accurately using the YOLOv8 algorithm. This research aims to implement YOLOv8 in detecting and classifying alphabets in Indonesian Sign Language (SIBI). This research was conducted at [University Name] University, using a dataset collected by taking photos of A-Z alphabet hand symbols which were then processed for labeling and model training. The model training process is carried out using data divided into three parts: training (60%), validation (20%), and testing (20%). Model testing resulted in a very high accuracy rate of 99.5%, with 99.1% precision, and 99.4% recall. These results show that the system developed is very reliable in detecting sign language in real-time. This research suggests that future research add variations in gesture data from various users to enrich the dataset, and consider using new algorithms or combining several algorithms to improve detection performance.

Keywords: *Sign Language Recognition, YOLOv8, Deep Learning, Object Detection, SIBI*

Abstrak

Berdasarkan perkembangan teknologi, khususnya di bidang komputasi, semakin memungkinkan pengembangan sistem yang mampu mendeteksi bahasa isyarat dengan lebih efisien. Salah satu masalah utama yang dihadapi adalah bagaimana cara mendeteksi dan mengklasifikasi gerakan bahasa isyarat secara akurat menggunakan algoritma YOLOv8. Penelitian ini bertujuan untuk mengimplementasikan YOLOv8 dalam mendeteksi dan mengklasifikasi abjad pada bahasa isyarat Indonesia (SIBI). Penelitian ini dilakukan di Universitas [Nama Universitas], dengan menggunakan dataset yang dikumpulkan melalui pengambilan foto simbol tangan abjad A-Z yang kemudian diproses untuk pelabelan dan pelatihan model. Proses pelatihan model dilakukan menggunakan data yang dibagi menjadi tiga bagian: pelatihan (60%), validasi (20%), dan pengujian (20%). Pengujian model menghasilkan tingkat akurasi yang sangat tinggi sebesar 99,5%, dengan presisi 99,1%, dan recall 99,4%. Hasil ini menunjukkan bahwa sistem yang dikembangkan sangat andal dalam mendeteksi bahasa isyarat secara real-time. Penelitian ini menyarankan agar penelitian selanjutnya menambahkan variasi data isyarat dari berbagai pengguna untuk memperkaya dataset, serta mempertimbangkan penggunaan algoritma terbaru atau penggabungan beberapa algoritma untuk meningkatkan kinerja deteksi .

Kata Kunci *: Pengenalan Bahasa Isyarat, YOLOv8, Deep Learning, Deteksi Objek, SIBI*

1. Pendahuluan

Perkembangan teknologi yang pesat, khususnya dalam bidang informasi dan komunikasi, berdampak signifikan pada kehidupan manusia. Sebagai makhluk sosial, manusia sangat bergantung pada komunikasi yang efektif. Namun, bagi komunitas tunarungu yang menggunakan bahasa isyarat sebagai sarana utama komunikasi, terdapat tantangan dalam berinteraksi dengan masyarakat luas yang tidak memahami bahasa isyarat. Bahasa isyarat adalah bahasa yang menggunakan gerak bibir, tubuh, dan tangan untuk mengekspresikan maksud dalam komunikasi.[1] Di Indonesia, terdapat dua variasi bahasa isyarat utama, yaitu SIBI (Sistem Isyarat Bahasa Indonesia) dan BISINDO (Bahasa Isyarat Indonesia)[2]

Untuk menjembatani kesenjangan komunikasi ini, sistem pengenalan bahasa isyarat berbasis teknologi menjadi solusi yang potensial. Penelitian sebelumnya, seperti yang dilakukan oleh [3], mengembangkan sistem deteksi bahasa isyarat secara real-time menggunakan teknik pembelajaran mesin dan pengolahan citra. Hasil penelitian menunjukkan akurasi sebesar 47,5% dalam mendeteksi gerakan tangan alfabet BISINDO melalui 20 kali uji coba. Tantangan yang dihadapi termasuk kondisi pencahayaan selama pengujian dan kesamaan bentuk tangan untuk beberapa gerakan.

Pendekatan Machine Learning, khususnya Deep Learning, yang merupakan salah satu teknik paling andal saat ini, dapat digunakan untuk membangun sistem pengenalan bahasa isyarat yang lebih canggih. Deep Learning menggunakan jaringan saraf tiruan untuk mempelajari pola-pola yang kompleks [4]. Algoritma You Only Look Once (YOLO) adalah salah satu algoritma Deep Learning berbasis CNN yang digunakan untuk deteksi objek. YOLO unggul dalam memproses gambar secara real-time dengan kecepatan 45 FPS dan memiliki akurasi yang cukup baik[5].

2. Metode Penelitian

2.1 *Machine Learning*

Machine learning adalah cabang kecerdasan buatan yang mengembangkan algoritma agar komputer dapat belajar dari data empiris, seperti data dari sensor atau basis data, sehingga mampu menghasilkan perilaku yang sesuai[6]. Teknik ini memungkinkan komputer mengubah data menjadi pola-pola tertentu secara otomatis, tanpa campur tangan manusia, dan menggunakan pola tersebut untuk mengidentifikasi masalah secara mandiri. Machine learning mempelajari algoritma dan model statistik yang memungkinkan komputer menyelesaikan tugas tanpa instruksi eksplisit, dengan membentuk model matematika berdasarkan "data latih" atau "training data" [7]

2.2 Bahasa Isyarat

Bahasa isyarat adalah bentuk komunikasi menggunakan gerakan bibir, tubuh, dan tangan untuk menyampaikan maksud. Sayangnya, mayoritas penduduk Indonesia tidak memahami atau tertarik mempelajari bahasa isyarat, sehingga komunikasi dengan individu tunarungu menjadi terbatas [1]

Pentingnya aksesibilitas komunikasi bagi individu dengan disabilitas pendengaran menekankan kebutuhan pengembangan sistem bahasa isyarat. Karena tidak semua orang mampu menggunakan bahasa isyarat, dan aksesibilitas seringkali tidak memadai, pengembangan ini menjadi sangat penting [8]

2.3 Deteksi Objek

Deteksi objek adalah metode dalam visi komputer yang bertujuan mengidentifikasi dan menemukan objek dalam gambar atau video, biasanya menggunakan pembelajaran mesin atau deep learning. Proses ini mereplikasi kemampuan manusia dalam mengenali objek dengan komputer, melibatkan klasifikasi objek dan pembuatan kotak pembatas di sekitarnya. Deteksi objek memproses gambar sebagai input dan mengidentifikasi elemen seperti manusia, gedung, atau kendaraan, dengan berbagai aplikasi dalam otomatisasi dan bidang visi komputer lainnya [9]

2.4 Klasifikasi

Klasifikasi adalah metode yang digunakan untuk mengidentifikasi pola dalam data, bertujuan memprediksi kelas dari objek yang belum dikenal. Teknik ini mengaitkan kelas tertentu ke data, sehingga memungkinkan prediksi dan analisis yang lebih akurat[10]. Dalam proses pengenalan pola, klasifikasi melibatkan pengelompokan objek ke dalam kelompok yang memiliki kesamaan. Ada dua jenis klasifikasi: klasifikasi diawasi (supervised), di mana model dilatih dengan data berlabel, dan klasifikasi tidak diawasi (unsupervised), di mana model mencoba menemukan pola tanpa data berlabel[11]

2.5 You Only Look Once (YOLO)

YOLO adalah algoritma deteksi objek yang membagi gambar input menjadi grid SxS, di mana setiap grid memprediksi kotak pembatas dan kelas objek. Algoritma ini kemudian menggabungkan kotak pembatas dari semua grid dan menyaring yang memiliki skor keyakinan rendah[12]. Metode YOLO membagi gambar atau frame video menjadi grid kecil, memperkirakan kotak pembatas (Bounding Box) dan kelas objek di setiap grid. Setiap kotak diberi skor yang menunjukkan tingkat keyakinan adanya objek. Algoritma kemudian menggabungkan dan menyaring kotak dengan skor rendah [13]

2.6 YoloV8

Algoritma YOLO mampu beroperasi pada kecepatan 45 fps menggunakan kartu grafis Titan X, menjadikannya ideal untuk deteksi objek real-time. YOLO terus berkembang, dengan versi terbaru YOLOv8 yang menawarkan peningkatan akurasi dan kecepatan. Karena YOLO memerlukan komputasi intensif, perangkat keras yang kuat diperlukan. Dalam penelitian ini, YOLOv8 diterapkan dengan dukungan perangkat keras yang memadai[14]

2.7 OpenCv

OpenCV adalah perpustakaan open source untuk visi komputer dan pembelajaran mesin yang menyediakan lebih dari 2500 algoritma. Dengan lisensi BSD, OpenCV mendukung penggunaan dan modifikasi oleh perusahaan, menawarkan berbagai algoritma visi komputer klasik serta teknik terbaru dalam pembelajaran mesin. OpenCV dikenal sebagai salah satu metode tercepat dan paling lengkap di bidang visi komputer [15]

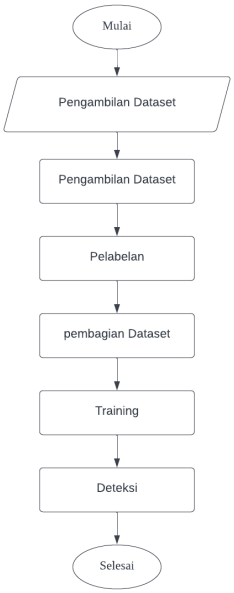
2.8 Deep Learning

Deep learning adalah cabang dari machine learning yang mengatasi batasan performa algoritma machine learning tradisional. Meskipun machine learning dapat meningkatkan performa dengan lebih banyak data, ia memiliki batasan signifikan ketika mencapai titik tertentu. Deep learning menawarkan solusi dengan kemampuan untuk mengatasi batasan tersebut dan meningkatkan performa secara signifikan

Dalam pengolahan citra digital, deep learning diterapkan untuk membantu dalam pengenalan dan klasifikasi objek dengan cepat dan akurat, sambil menangani data dalam jumlah besar. Salah satu algoritma deep learning yang digunakan dalam image processing adalah Convolutional Neural Network (CNN) [16]

2.9 Perancangan Sistem

Flowchart adalah diagram yang menggambarkan opsi dan langkah-langkah yang diperlukan untuk menyelesaikan suatu proses dalam program, yang dikenal sebagai diagram alur. Setiap langkah dijelaskan dalam diagram tersebut, dengan garis atau panah yang menghubungkan setiap tahap.



Gambar 1 Perancangan Sistem

2.10 Teknik pengujian Sistem

Pada penelitian ini, teknik pengujian sistem menggunakan Pengujian Empiris. Sistem pengujian empiris dalam penelitian pengenalan bahasa isyarat menggunakan deteksi objek dengan metode

YOLOv8 adalah proses pengujian yang mengandalkan observasi, pengukuran, dan analisis data empiris untuk mengevaluasi kinerja sistem tersebut dalam mengenali bahasa isyarat.

2.11 Teknik Analisis Data

Analisis data adalah proses yang melibatkan pengumpulan, pengelompokan, dan seleksi data secara menyeluruh dari berbagai sumber seperti hasil wawancara, catatan lapangan, dan dokumentasi. Proses ini dilakukan secara sistematis dengan mengorganisir data ke dalam kategori, membagi data menjadi unit-unit yang relevan, melakukan sintesis, mengidentifikasi pola, memprioritaskan informasi yang penting untuk dipelajari, dan menyimpulkan temuan dengan cara yang jelas dan mudah dipahami oleh peneliti serta pembaca lainnya.

3. Hasil dan Pembahasan

3.1 Pengambilan Dataset

Proses pengambilan data pada dataset dilakukan dengan menggunakan kamera ponsel untuk mengambil gambar simbol abjad SIBI mulai dari huruf A hingga Z. Keseluruhan dataset terdiri dari 1924 data gambar yang berhasil dikumpulkan. Gambar-gambar di bawah ini merupakan contoh dari dataset yang berisi simbol-simbol abjad SIBI.

Gambar 2 Dataset

Tabel 1 Jumlah Data

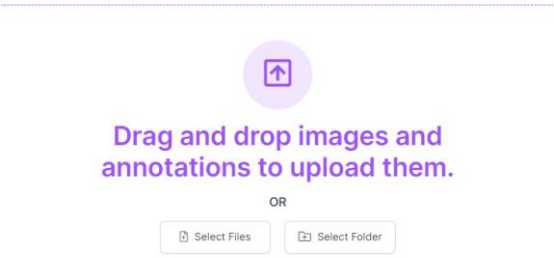
Nama Kelas	Jumlah Data
A	74
B	74
C	74
D	74
E	74
F	74
G	74
H	74
I	74
J	74
K	74
L	74
M	74
N	74
O	74
P	74
Q	74
R	74
S	74
T	74
U	74
V	74
W	74
X	74
Y	74
Z	74
Total	1924

3.2 Pelabelan Gambar

Pada tahap ini, gambar – gambar diberi label untuk memungkinkan sistem mengenali nama dari simbol yang akan di deteksi. Proses pelabelan dilakukan dengan menggunakan aplikasi Roboflow, yang bertujuan untuk menambahkan label pada gambar – gambar simbol abjad SIBI sesuai dengan pembagian dan kategori yang telah di tetapkan sebelumnnya.

a. Upload Dataset

Data gambar yang telah dikumpulkan kemudian dimasukkan ke dalam Roboflow sebelum proses pelabelan dilakukan.



Gambar 3 Upload Dataset

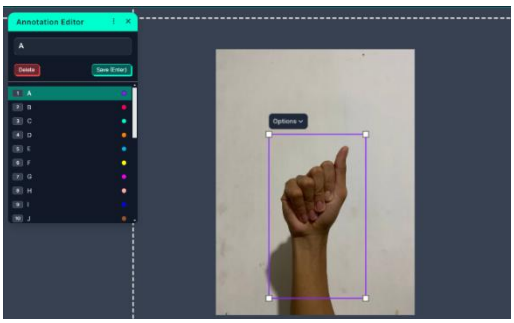
b. Pembuatan Kelas Dataset

Tujuan dari pembuatan kelas ini adalah untuk menyederhanakan proses penentuan dataset gambar yang telah diberi bounding box. Kelas ini secara otomatis akan menghasilkan 26 kelas yang telah ditentukan sebelumnya, di mana dataset akan disusun sesuai dengan kategori masing-masing.

Gambar 4 Pembuatan Kelas Dataset

c. Proses Pelabelan

Proses pelabelan gambar menggunaka Roboflow dengan menambahkan bounding box atau bingai di sekitar objek dalam gambar. Setelah pembuatan bounding box, secara otomatis muncul kelas dengan kategori yang telah di tetapkan sebelumnnya.



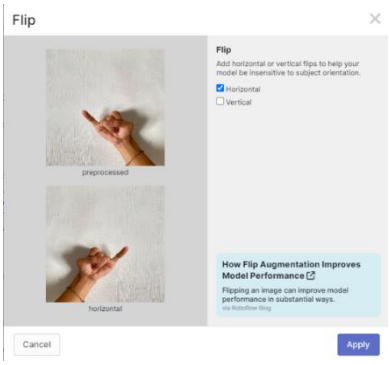
Gambar 5 Proses Pelabelan

3.3 Pembagian Dataset

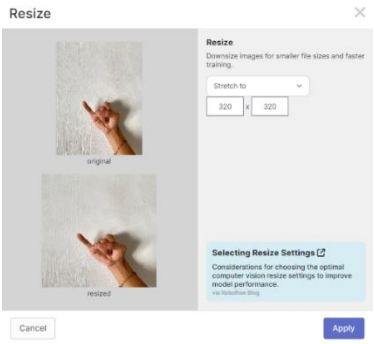
a. Pengolahan Gambar

Dalam pengolahan gambar menggunakan Roboflow, modifikasi dilakukan seperti perubahan warna, bentuk, ukuran, dan penerapan teknik Preprocessing serta Augmentation. Salah satu teknik

yang digunakan adalah resizing gambar menjadi 320 x 320 untuk mengurangi beban kerja GPU dan mempercepat pelatihan model. Augmentasi bertujuan untuk memperkaya variasi data pelatihan dengan memodifikasi pola, posisi, dan atribut citra asli. Dalam augmentasi bounding box, pembalikan citra dilakukan secara vertikal dan horizontal untuk meningkatkan kemampuan model mengenali beragam pola.



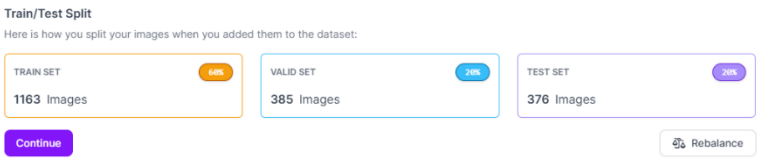
Gambar 7 Flip



Gambar 6 Resize

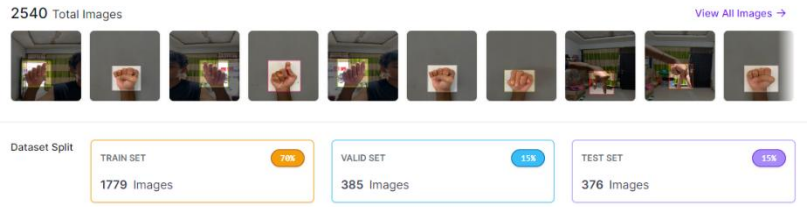
b. Pembagian data

pengelompokan dataset dilakukan menggunakan alat Roboflow untuk membaginya menjadi training, validasi, dan pengujian. Sebelum dilakukan augmentasi, total dataset yang terdiri dari 1924 gambar akan dibagi menjadi 60% untuk data pelatihan (1163 gambar), 20% untuk data validasi (385 gambar), dan 20% untuk data pengujian (376 gambar).



Gambar 8 Pembagian Dataset

Setelah melakukan augmentasi dengan menggunakan bounding box dan menerapkkan flip secara horizontal da vertikal, data awal akan di manipulasi untuk memperkaya dataset pelatihan. Setelah augmentasi, dataset yang terdiri dari 2514 gambar akan dibagi menjadi 70% untuk dataset pelatihan (1779 gambar), 15% untuk dataset validasi (385 gambar), dan 15% untuk data set pengujian (376 gambar).

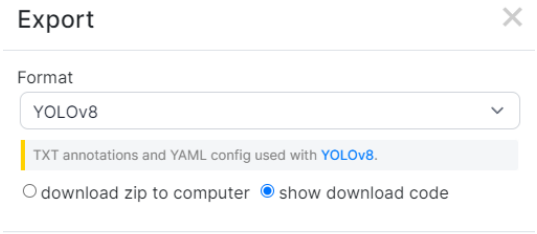


Gambar 9 Dataset Split

3.4 Training

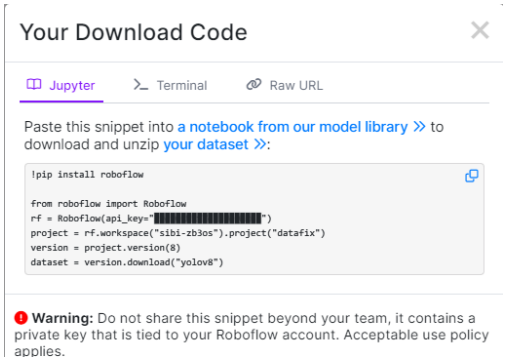
a. Export Dataset Roboflow

Setelah dataset dianotasi, dilakukan ekspor dataset menggunakan format yang kompatibel dengan YOLOv8.



Gambar 10 Export Dataset

Setelah proses ekspor dataset, Roboflow akan menyediakan API yang berisi dataset yang telah dianotasi. API ini kemudian akan digunakan untuk melatih model di Google Colab.



Gambar 11 API Dataset

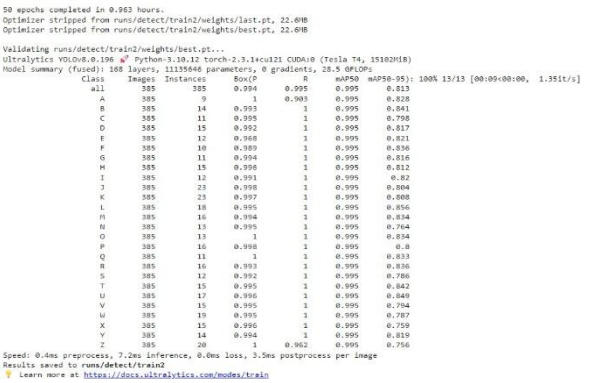
b. Training GoogleColab

Pelatihan model dilakukan dengan memanfaatkan API yang diperoleh dari Roboflow. API ini akan diakses dari Roboflow untuk melatih model di Google Colab

```
!mkdir {HOME}/datasets
%cd {HOME}/datasets

!pip install roboflow

from roboflow import Roboflow
rf = Roboflow(api_key="29d9QRWktYquIUdQMx6i")
project = rf.workspace("sibi-zb3os").project("datafix")
version = project.version(8)
dataset = version.download("yolov8")
```

Setelah API dari Roboflow berhasil diakses, API tersebut akan digunakan untuk melatih model.

```
%cd {HOME}

!yolo task=detect mode=train model=yolov8s.pt
data=/content/datasets/datafix-8/data.yaml epochs=50 imgsz=800 plots=True
```



Gambar 12 Hasil Training

Setelah melakukan percobaan pelatihan di Google Colab dengan menggunakan 60% data pelatihan, 20% data validasi, dan 20% data pengujian, diperoleh akurasi sebesar 99,5%.

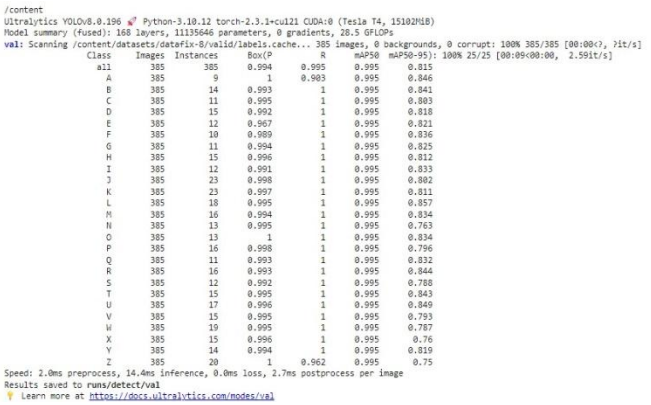
Setelah proses pelatihan selesai, akan ada berkas yang dihasilkan, yaitu best.pt. berkas tersebut akan dipakai utnuk menguji model pada video yang belum dilihat oleh model sebelumnya.

Gambar 13 Model Dataset

c. Validating

Setelah dilatih, model divalidasi dengan data baru yang belum pernah dilihat sebelumnya untuk menguji kemampuannya mengenali objek dengan akurasi tinggi. Proses ini penting untuk mengevaluasi kinerja model dalam situasi nyata dan menentukan apakah model siap digunakan atau memerlukan penyesuaian.

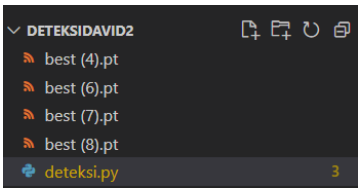
```
%cd {HOME}
!yolo task=detect mode=val model={HOME}/runs/detect/train/weights/best.pt
data=/content/datasets/datasets/datafix-8/data.yaml
```



Gambar 14 Hasil Validasi

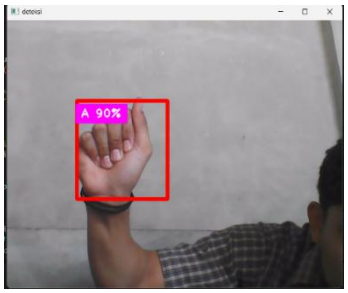
3.5 Pengujian Sistem

Langkah penting dalam pengujian adalah menggunakan model terbaik (best.pt) untuk deteksi objek real-time dengan YOLOv8. Model ini dimuat ke script Python, memproses video langsung dari kamera, dan melakukan inferensi pada setiap frame. Hasil deteksi ditampilkan dengan kotak pembatas dan label, memastikan sistem bekerja secara akurat dan andal pada data langsung.



Gambar 15 Testing Model

Tabel 2 Hasil Pengujian

Data pengujian	Kelas dataset	Hasil Deteksi	Tingkat Kepastian
	A		90%
	B		92%

Dalam proses testing, 26 kelas gambar simbol abjad SIBI diuji, masing-masing dengan 74 gambar. Hasil deteksi YOLOv8 menunjukkan tingkat kepastian tertinggi 94% dan terendah 74%. Kepastian rendah terjadi karena model kesulitan mendeteksi atau mengklasifikasikan huruf yang mirip, seperti R dan U.

4. Kesimpulan

Hasil penelitian menunjukkan bahwa implementasi metode YOLO untuk deteksi objek pada simbol abjad SIBI berhasil dengan baik. Sistem ini mendeteksi 1924 gambar yang terbagi dalam 26 kelas, dengan akurasi mencapai 99.5%, presisi 99.1%, dan recall 99.4%. Tingkat kepastian deteksi bervariasi, dengan nilai tertinggi 94% dan terendah 74%, mengindikasikan bahwa sistem deteksi ini akurat dan andal dalam kondisi pengujian. Untuk penelitian selanjutnya, disarankan menambah variasi data isyarat dan pengguna, menggunakan algoritma terbaru atau kombinasi beberapa algoritma untuk meningkatkan kinerja, serta memperluas penelitian ke bahasa isyarat dari berbagai negara dan budaya untuk mengembangkan model yang lebih universal.

Referensi

[1] R. Haris Alfikri *et al.*, “PEMBANGUNAN APLIKASI PENERJEMAH BAHASA ISYARAT DENGAN METODE CNN BERBASIS ANDROID,” 2022. [Online]. Available: https://ejurnal.teknokrat.ac.id/index.php/teknoinfo/index

[2] A. S. Nugraheni, A. P. Husain, and H. Unayah, “Optimalisasi Penggunaan Bahasa Isyarat Dengan Sibi Dan Bisindo Pada Mahasiswa Difabel Tunarungu Di Prodi Pgmi Uin Sunan Kalijaga,” *J. Holistika*, vol. 5, no. 1, p. 28, 2023, doi: 10.24853/holistika.5.1.28-33.

[3] 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.

[4] A. Prima, "Rancang Bangun Sistem Pendeteksi Aneka Ragam Buah Menggunakan MobileNetv2," *J. Sistim Inf. dan Teknol.*, vol. 5, no. 2, pp. 208–215, 2023, doi: 10.60083/jsisfotek.v5i2.217.

[5] N. D. G. Drantantiyas *et al.*, "Performasi Deteksi Jumlah Manusia Menggunakan YOLOv8," *JASIEK (Jurnal Apl. Sains, Informasi, Elektron. dan Komputer)*, vol. 5, no. 2, pp. 63–68, 2023, doi: 10.26905/jasiek.v5i2.11605.

[6] Rahmadini, E. Lubis Lorencis Erika, A. Priansyah, Y. R.W.M, and T. Meutia, "PENERAPAN DATA MINING UNTUK MEMPREDIKSI HARGA BAHAN PANGAN DI INDONESIA MENGGUNAKAN ALGORITMA K-NEAREST NEIGHBOR," 2023.

[7] Jérémy, H. Frezza-Buet, M. Geist, and F. Pennerath, "Machine Learning.pdf," 2020.

[8] I. Sari, Fivrenodi, E. Altiarika, and Sarwindah, "Sistem Pengembangan Bahasa Isyarat Untuk Berkomunikasi dengan Penyandang Disabilitas (Tunarungu)," *J. Inf. Technol. Soc.*, vol. 1, no. 1, pp. 20–25, 2023, doi: 10.35438/jits.v1i1.21.

[9] R. I. Tiyar and D. H. Fudholi, "Kajian Pengaruh Dataset dan Bias Dataset terhadap Performa Akurasi Deteksi Objek," *Petir*, vol. 14, no. 2, pp. 258–268, 2021, doi: 10.33322/petir.v14i2.1350.

[10] A. Syukron, S. Sardiarinto, E. Saputro, and P. Widodo, "Penerapan Metode Smote Untuk Mengatasi Ketidakseimbangan Kelas Pada Prediksi Gagal Jantung," *J. Teknol. Inf. dan Terap.*, vol. 10, no. 1, pp. 47–50, 2023, doi: 10.25047/jtit.v10i1.313.

[11] N. Wakhidah, "Clustering Menggunakan K-Means Algorithm," *J. Transform.*, vol. 8, no. 1, p. 33, 2010, doi: 10.26623/transformatika.v8i1.45.

[12] A. Tenriawaru and F. Qamaria, "Implementasi Algoritma Convolutional Neural Network untuk Menghitung Kepadatan Ayam Menggunakan You Only Look Once," vol. 9, pp. 1–5, 2024.

[13] I. Andi, M. Muchtar, and J. Y. Sari, "Mask Detection Using the YOLO (You Only Look Once) Method," *J. Media Inf. Teknol.*, vol. 1, no. 1, pp. 1–12, 2024, doi: 10.69616/mit.v1i1.165.

[14] A. Setiyadi, E. Utami, and D. Ariatmanto, "Analisa Kemampuan Algoritma YOLOv8 Dalam Deteksi Objek Manusia Dengan Metode Modifikasi Arsitektur," *J. Sains Komput. Inform. (J-SAKTI*, vol. 7, no. 2, pp. 891–901, 2023.

[15] H. Muchtar and R. Apriadi, "Implementasi Pengenalan Wajah Pada Sistem Penguncian Rumah Dengan Metode Template Matching Menggunakan Open Source Computer Vision Library (Opencv)," *Resist. (elektRonika kEndali Telekomun. tenaga List. kOmputeR)*, vol. 2, no. 1, p. 39, 2019, doi: 10.24853/resistor.2.1.39-42.

[16] F. F. Maulana and N. Rochmawati, "Klasifikasi Citra Buah Menggunakan Convolutional Neural Network," *J. Informatics Comput. Sci.*, vol. 1, no. 02, pp. 104–108, 2020, doi: 10.26740/jinacs.v1n02.p104-108.