



IMPLEMENTASI ALGORITMA CONVOLUTIONAL NEURAL NETWORK UNTUK MENERJEMAHKAN BAHASA ISYARAT

Izza Amri
Universitas Pamulang
izzaamri969@gmail.com

ABSTRACT

Sign language interpretation is an important part of facilitating effective communication with individuals with hearing disabilities. This research describes the implementation of the Convolutional Neural Network (CNN) algorithm using YOLOV5 to translate sign language into text. Experimental methodology is used by utilizing a dataset containing hand images depicting signs in Indonesian Sign Language (BISINDO). The YOLOV5 training process includes object detection and classification steps to recognize complex signatures. Experimental results show YOLOV5's ability to translate sign language with a satisfactory level of accuracy. This research contributes to the development of a sign language translation system that can be used in a variety of contexts, supporting inclusivity and facilitating communication between the deaf community and individuals who do not use sign language.

Keywords: Convolutional Neural Network, YOLOV5, object detection, Indonesian Sign Language, BISINDO.

ABSTRAK

Penerjemahan bahasa isyarat merupakan bagian penting dalam memfasilitasi komunikasi efektif dengan individu penyandang disabilitas pendengaran. Penelitian ini menggambarkan implementasi algoritma *Convolutional Neural Network* (CNN) menggunakan YOLOV5 untuk menerjemahkan bahasa isyarat menjadi teks. Metodologi eksperimen digunakan dengan memanfaatkan dataset berisi citra tangan yang menggambarkan isyarat dalam Bahasa Isyarat Indonesia (BISINDO). Proses pelatihan YOLOV5 melibatkan langkah-langkah deteksi objek dan klasifikasi untuk mengenali isyarat tangan yang kompleks. Hasil eksperimen menunjukkan kemampuan YOLOV5 dalam menerjemahkan bahasa isyarat dengan tingkat akurasi yang memuaskan. Penelitian ini memberikan kontribusi pada pengembangan sistem penerjemah bahasa isyarat yang dapat digunakan dalam berbagai konteks, mendukung inklusivitas dan memudahkan komunikasi antara komunitas tuli dengan individu yang tidak menggunakan bahasa isyarat.

Kata Kunci: Convolutional Neural Network, YOLOV5, Deteksi Objek, Bahasa Isyarat Indonesia.

PENDAHULUAN

1.1 Latar Belakang

Komunikasi merupakan peranan penting bagi interaksi manusia. Melalui komunikasi, seseorang dapat memberitahu orang lain apa yang mereka pikirkan dan memahami arti dari pesan yang sama. Makna pesan tersampaikan dengan benar dan



tujuan penyampaian pesan tercapai. Komunikasi dapat dilakukan oleh siapa saja dalam bentuk komunikasi verbal maupun non verbal. Komunikasi verbal adalah komunikasi dalam bentuk kata-kata, sedangkan komunikasi nonverbal adalah komunikasi menggunakan bahasa tubuh. (Gustiari, Sitorus, & Midyanti, 2020).

Bahasa isyarat merupakan bahasa yang berfokus pada komunikasi melalui bahasa tubuh. Bahasa isyarat adalah bahasa yang digunakan oleh komunitas tuli untuk berkomunikasi. Selain itu, bahasa isyarat juga menjadi alat bagi penggunanya untuk mengidentifikasi diri dan mengumpulkan informasi (Algebar & Suharjito, 2020). Dua jenis bahasa isyarat yang umum digunakan di Indonesia yaitu Sistem Bahasa Isyarat Indonesia (SIBI) dan Bahasa Isyarat Indonesia (BISINDO). SIBI adalah bahasa isyarat yang berkembang dari perolehan Bahasa Isyarat Amerika (ASL) dan merupakan sarana representasi tata bahasa Indonesia lisan dalam bentuk isyarat. BISINDO merupakan bahasa isyarat yang berkembang secara alami pada masyarakat tuli dan memiliki struktur tata bahasa yang berbeda dengan bahasa Indonesia lisan (Arisandi, 2022).

Namun, sering terjadi kesalahan dalam penerjemahan bahasa isyarat karena masyarakat awam kurang memiliki pengetahuan dan keterampilan untuk memahami bahasa isyarat. (Nurhayati, Eridani, & Tsalavin, 2022). Keterbatasan jumlah penerjemah bahasa isyarat merupakan salah satu kesulitan dalam berkomunikasi dengan komunitas Tuli. Selain itu, interaksi dengan komunitas Tuli menjadi sulit karena banyak orang yang tidak menguasai bahasa isyarat. Oleh karena itu, diperlukan sebuah sistem

1.2 Rumusan Masalah

Dengan mengacu pada latar belakang masalah dan telah diuraikan pada identifikasi masalah di atas maka pokok permasalahan dapat dirumuskan masalah sebagai berikut:

- a. Bagaimana cara mengimplementasikan metode CNN menggunakan algoritma YOLOV5 untuk menerjemahkan bahasa isyarat BISINDO secara *real time* dengan menggunakan *webcam*?
- b. Bagaimana kinerja algoritma YOLOV5 dalam menerjemahkan bahasa isyarat BISINDO ke dalam bahasa Indonesia?

TINJAUAN PUSTAKA

2.1. Penelitian Terkait

Pada penelitian ini melakukan analisis pada jurnal yang berkaitan dengan implementasi algoritma *Convolutional Neural Network* sebagai acuan terhadap penelitian dengan mencantumkan kajian-kajian yang dilakukan oleh penelitian sebelumnya sehingga dapat membantu untuk membangun aplikasi dan membantu membangun sebuah kerangka pemikiran. Penelitian ini menggunakan beberapa sumber pustaka. Pustaka yang relevan pada penelitian ini ditinjau dari sisi kasus penelitian dan metode yang digunakan dalam penelitian. Kasus penelitian yang dilakukan adalah mengenai menerjemahkan bahasa isyarat BISINDO. Metode yang digunakan dalam penelitian ini adalah metode *Convolutional Neural Network*.



2.2. Landasan Teori

2.2.1. Bahasa Isyarat Bisindo

BISINDO merupakan singkatan dari Bahasa Isyarat Indonesia yang merupakan bahasa murni sahabat tuli di Indonesia (Dewi, Wahyuningrum, & Prasetyo, 2021). Bahasa murni berarti bahasa yang berkembang secara alami dari komunitas penggunanya, tanpa campur tangan dari pihak luar. BISINDO berbeda dengan SIBI (Sistem Isyarat Bahasa Indonesia) yang merupakan bahasa isyarat yang disesuaikan dengan struktur bahasa Indonesia (Arisandi, 2022). SIBI dibuat oleh pemerintah pada tahun 2004 sebagai upaya untuk meningkatkan akses pendidikan bagi penyandang tunarungu.

2.2.2 Python

Bahasa *Python* adalah sebuah bahasa pemrograman yang dirancang untuk mudah dibaca dan ditulis oleh manusia, serta mudah diproses dan dijalankan oleh mesin. Bahasa Python dibuat oleh Guido van Rossum dan dirilis pada tahun 1991. Bahasa Python memiliki sintaks yang sederhana, fleksibel, dan konsisten, serta mendukung berbagai paradigma pemrograman, seperti prosedural, berorientasi objek, fungsional, dan skrip (Alfarizi, Al-farish, Taufiqurrahman, Ardiansah, & Elgar, 2023).

2.2.3. Artificial Intelligence (AI)

Artificial Intelligence (AI) atau dalam Bahasa Indonesia biasa disebut kecerdasan buatan merupakan salah satu disiplin ilmu dalam bidang komputer yang terus berkembang. Bidang AI berusaha tidak hanya untuk memahami tetapi juga untuk membangun entitas cerdas. AI meliputi banyak sub-bidang, mulai dari bidang umum sampai untuk tugas-tugas spesifik (Tim Teaching Kecerdasan Buatan, 2012).

2.2.4. Machine Learning

Istilah machine learning pertama kali didefinisikan oleh Arthur Samuel ditahun 1959. Menurut Arthur Samuel, machine learning adalah salah satu bidang ilmu komputer yang memberikan kemampuan pembelajaran kepada komputer untuk mengetahui sesuatu tanpa pemrogram yang jelas. *Machine Learning* merupakan sebuah model komputasi statistik, yang berfokus pada prediksi menggunakan komputer. Algoritma machine learning membangun model matematika dari data sampel, yang dikenal sebagai "data pelatihan atau *data training*", untuk membuat prediksi atau keputusan tanpa diprogram secara eksplisit untuk melakukan tugas (Santoso, Megasari, & Hambali, 2020). *Machine Learning* adalah salah satu cabang ilmu Kecerdasan Buatan (*Artificial Intelligence*) yang berkembang sangat cepat. *Machine learning* menggunakan metode populer untuk menggantikan atau menirukan perilaku manusia dalam menyelesaikan masalah dan melakukan otomatisasi (Wahyono, 2020).

2.2.5 Deep Learning

Deep learning adalah salah satu cabang dari *artificial intelligence* (AI) yang menggunakan jaringan saraf tiruan (*neural network*) yang memiliki banyak lapisan untuk belajar dari data secara otomatis. *Deep learning* dapat mengolah data yang



kompleks, seperti gambar, suara, teks, dan *video*, dan menghasilkan I yang sesuai dengan tujuan pembelajaran (Santoso & Ariyanto, 2018).

2.2.6 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) merupakan pengembangan dari *multilayer perceptron* (MLP) yang didesain untuk mengolah data dua dimensi dalam bentuk citra. CNN ini termasuk kedalam jenis *Deep Neural Network* karena kedalaman jaringan yang tinggi dan banyak diaplikasikan pada data citra. Pada dasarnya klasifikasi citra dapat digunakan dengan MLP, akan tetapi dengan metode MLP kurang sesuai untuk digunakan karena tidak menyimpan informasi spasial dari data cita dan menganggap setiap piksel adalah fitur yang independen sehingga menghasilkan hasil yang kurang baik. Penelitian awal yang mendasari penemuan CNN ini pertama kali dilakukan oleh Hubel dan Wiesel (Hubel & Wiesel, T, 1968). Secara teknis, CNN merupakan arsitektur yang dapat dilatih dan berdiri dari beberapa tahap. *Input* dan *output* dari setiap tahap terdiri dari beberapa *array* yang biasa disebut *feature map*. Setiap tahap terdiri dari 3 *layer* yaitu konvolusi, fungsi aktivasi *layer*, dan *pooling layer* (Nada, 2019).

2.2.7 Object Detection

Deteksi objek merupakan suatu prosedur yang penting dalam bidang *computer vision* yang bertujuan untuk mengenali dan mengklasifikasikan objek dalam citra, serta mengestimasi lokasi relatif objek dengan pembatasan (*bounding box*) yang mengelilingi setiap objek tersebut. Jika deteksi hanya mengidentifikasi satu kelas objek dalam citra, hal ini dikenal sebagai *single-class object detection*. Namun, jika deteksi mencakup identifikasi beberapa kelas objek yang mungkin ada dalam citra, maka ini disebut sebagai *multi-class object detection* (Mir, Sharma, Rout, & Umer, 2023).

2.2.8 Yolov5

YOLOv5 merupakan penambahan terbaru dalam seri YOLO. YOLOv5 telah mengalami peningkatan berdasarkan fondasi YOLOv4, dengan peningkatan signifikan dalam kecepatan eksekusi hingga mencapai 10 fps. Selain itu, YOLOv5 memiliki ukuran yang lebih kecil dibandingkan YOLOv4, dengan ukuran file hampir 90% lebih ringan. Hal ini memungkinkan implementasi YOLOv5 untuk perangkat terintegrasi.

2.2.9 Intersection Over Union

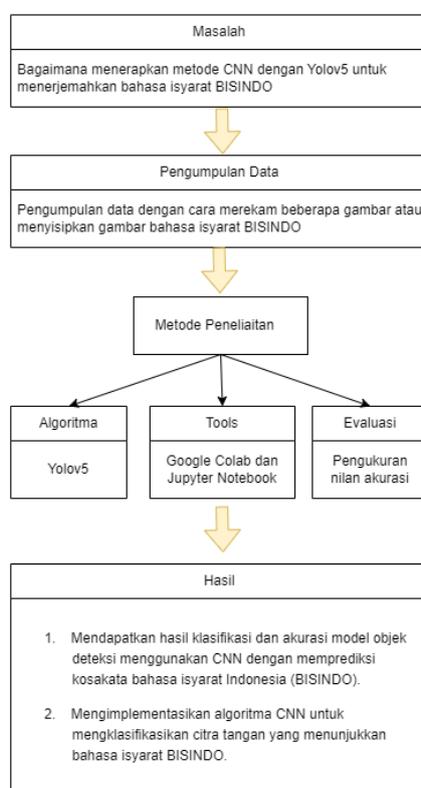
Intersection over Union (IoU) adalah metode evaluasi yang penting untuk mengukur akurasi deteksi objek dalam konteks dataset. IoU melibatkan perbandingan antara dua area yang akan dihitung, yaitu area dari *ground-truth bounding box* (*bounding box* aktual yang mengelilingi objek) dan area yang dideteksi oleh model yang dikembangkan. Metode ini digunakan untuk menentukan sejauh mana dua *bounding box* tersebut tumpang tindih atau bersinggungan. Hasil dari perhitungan IoU ini adalah angka yang mengindikasikan sejauh mana *bounding box* yang dihasilkan oleh model cocok atau akurat dibandingkan dengan *ground-truth bounding box* yang seharusnya mengelilingi objek tersebut (Shianto, Gunadi, & Setyati, 2019).

2.2.10 Confusion Matriks

Confusion matrix adalah sebuah tabel yang digunakan untuk mengukur kinerja model klasifikasi berdasarkan data *supervise*, yang memiliki jawaban benar. *Confusion matrix* digunakan untuk menyajikan hasil klasifikasi dalam bentuk tabel, dan dari tabel ini, berbagai metrik evaluasi kinerja model dapat dihitung, seperti akurasi, presisi, *F-Score*, dan beragam variabel lainnya tergantung pada kondisi data yang diprediksi atau diklasifikasikan. Tabel *confusion matrix* biasanya berisi empat sel yang merepresentasikan berbagai kombinasi dari hasil prediksi model dan kelas aktual data, dan strukturnya (Shianto, Gunadi, & Setyati, 2019).

2.3. Kerangka Pemikiran

Kerangka pemikiran dari penelitian ini digambarkan pada gambar 2.7 sebagai berikut:



Gambar 2.12 kerangka pemikiran
METODOLOGI

3.1.2. Kebutuhan Fungsional

Kebutuhan fungsional merupakan deskripsi proses fungsional yang berupa penjelasan rinci dari setiap fungsi yang digunakan untuk memecahkan masalah. Fungsi-fungsi yang dimiliki alat ini adalah membaca koordinat pola tangan yang terdiri dari posisi telapak tangan, posisi jari dan putaran tangan lalu mengurutkannya ke dalam berbagai kelas menggunakan metode *Convolutional Neural Network* (CNN). Hasil terjemahan bahasa isyarat akan muncul di monitor laptop sebagai *outputnya*.

3.1.3. Metode Penelitian

3.1.3.1. Jenis dan Pendekatan Penelitian

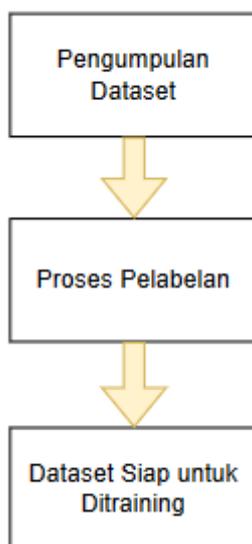
Jenis penelitian ini merupakan penelitian eksperimental yang bersifat aplikatif sehingga dari ruang lingkup masalah dapat dilakukan dengan metode studi pustaka (*library research*), metode pengumpulan data lapangan (*field research*) dan implementasi.

3.1.3.2. Sumber Data

Pengumpulan data dengan cara merekam beberapa gambar atau menyisipkan gambar bahasa isyarat BISINDO.

3.2. Perancangan Penelitian

3.2.1. Perancangan Dataset



Gambar 3.1 Perancangan dataset

Dataset adalah sekumpulan data yang digunakan untuk melatih, menguji, atau mengevaluasi model pembelajaran mesin. Dataset yang digunakan dalam penelitian ini adalah dataset yang berisi gerakan tangan yang merepresentasikan kata-kata dalam Bahasa Isyarat Indonesia (BISINDO). Dataset ini dirancang dengan beberapa langkah, yaitu:

1. Pengumpulan dataset.

Dataset ini diperoleh dari hasil pengambilan gambar atau *video* sendiri dengan menggunakan kamera *smartphone* atau *webcam*.

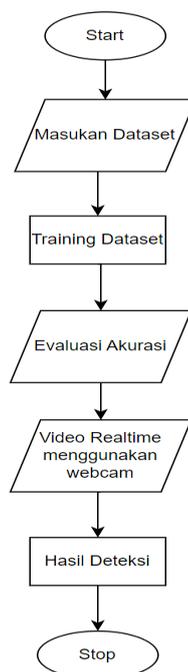
2. Proses pelabelan

Pelabelan adalah proses memberikan label atau kelas pada data, yang digunakan untuk mengidentifikasi atau mengklasifikasikan data. Pelabelan dilakukan dengan menggunakan aplikasi LabelImg, yang merupakan sebuah alat anotasi gambar yang dapat membuat *bounding box* dan label pada gambar. Pelabelan dilakukan dengan cara menandai lokasi dan ukuran tangan pada gambar dengan *bounding box*, dan memberikan label yang sesuai dengan huruf yang direpresentasikan oleh tangan. Label yang digunakan adalah kata halo, kapan, maaf, terimakasih, sama-sama dalam BISINDO. Pelabelan dilakukan secara manual oleh peneliti.

3. Dataset siap untuk ditraining



Dataset yang telah dilabeli kemudian disimpan dalam format yang sesuai untuk di *training* oleh model pembelajaran mesin. Format yang digunakan adalah format XML, yang merupakan sebuah format yang mudah dibaca dan ditulis oleh manusia dan mesin. Format XML berisi informasi tentang nama file, ukuran gambar, koordinat bounding box, dan label. Selain itu, dataset juga dibagi menjadi dua bagian, yaitu train dan test, yang digunakan untuk melatih dan menguji model. Pembagian dataset dilakukan dengan menggunakan rasio 80:20, yaitu 80% data untuk train dan 20% data untuk test. Pembagian dataset dilakukan secara acak dengan menggunakan aplikasi Python, yang merupakan sebuah bahasa pemrograman yang populer digunakan dalam pembelajaran mesin . Dataset yang telah dibagi dan disimpan dalam format XML kemudian disimpan dalam sebuah file konfigurasi yang berisi informasi penting tentang dataset, seperti nama, deskripsi, format, sumber, lisensi, splits, dan labels. File konfigurasi ini disebut sebagai `dataset.yaml`, yang menggunakan format YAML (YAML Ain't Markup Language), yaitu sebuah format yang mudah dibaca dan ditulis oleh manusia dan mesin⁵. File `dataset.yaml` ini dapat digunakan untuk memuat, memproses, atau menyimpan dataset dengan menggunakan fungsi-fungsi yang tersedia dalam library pembelajaran mesin, seperti *PyTorch*, TensorFlow, atau Scikit-learn. File `dataset.yaml` ini juga dapat digunakan untuk mendokumentasikan atau berbagi dataset dengan pihak lain yang tertarik dengan dataset tersebut



Gambar 3.2 kerangka uji

ANALISA DAN PEMBAHASAN

4.1 Implementasi Yolov5

4.1.1 Menyiapkan Dataset

Menyiapkan Dataset adalah tahapan krusial dalam penelitian ini untuk mengumpulkan dataset yang diperlukan untuk pelatihan model deteksi bahasa isyarat dengan menggunakan algoritma YOLOv5. Data gambar diambil melalui perangkat *webcam* sebanyak 490 gambar menggunakan kode program seperti berikut:

```
import uuid
import os
import time
IMAGES_PATH = os.path.join('dataset', 'gambar') #/dataset/gambar
labels = ['halo', 'maaf', 'kapan', 'terimakasih', 'sama-sama']
number_imgs = 20

cap = cv2.VideoCapture(0)
for label in labels:
    print("Mengumpulkan gambar untuk kategori {}".format(label))
    time.sleep(5)
    # Loop through image range
    for img_num in range(number_imgs):
        print("Mengumpulkan gambar untuk kategori {}, gambar ke-{}".format(label, img_num))
        # Mendapatkan gambar dari webcam
        ret, frame = cap.read()
        # Menentukan path untuk gambar yang diambil
        imgname = os.path.join(IMAGES_PATH, label+'-'+str(uuid.uuid1())+'.jpg')
        # Menyimpan gambar ke dalam file
        cv2.imwrite(imgname, frame)
        # Menampilkan gambar di layar
        cv2.imshow("Pengumpulan Gambar", frame)
        # Waktu delay selama 2 detik antara pengambilan gambar
        time.sleep(2)
        if cv2.waitKey(10) & 0xFF == ord('q'):
            break
    cap.release()
    cv2.destroyAllWindows()
```

Gambar 4.1 pengumpulan data



Gambar 4.2 hasil dari pengambilan gambar

Tahapan ini memastikan bahwa dataset gambar yang diperlukan untuk pelatihan model deteksi bahasa isyarat dikumpulkan dengan baik dan siap untuk tahapan selanjutnya dalam penelitian ini. Data yang dihasilkan akan digunakan untuk melatih model deteksi objek menggunakan algoritma YOLOv5.

4.1.2 Labelling Images

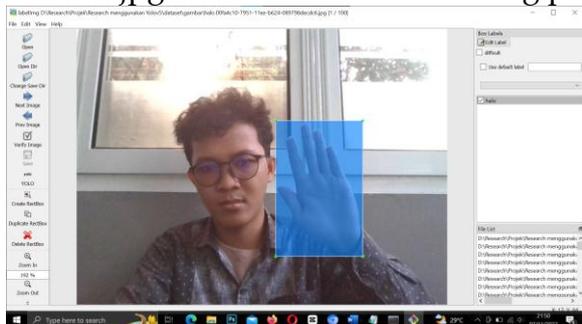
Proses pengumpulan gambar merupakan tahap awal yang krusial dalam persiapan data untuk melatih model deteksi bahasa isyarat menggunakan YOLOv5. Akan tetapi, demi memastikan keberhasilan pelatihan model, informasi lebih lanjut mengenai letak dan jenis objek dalam setiap gambar perlu diperoleh. Oleh karena itu, langkah berikutnya adalah proses penandaan (*labeling*) gambar, di mana kotak pembatas (*bounding box*) ditempatkan di sekitar objek dalam gambar, serta label yang

sesuai ditetapkan. Dalam konteks penandaan gambar, perangkat lunak yang digunakan *LabelImg*, yang merupakan *open source*.

Setelah selesai dengan proses pengumpulan dan penandaan gambar, langkah berikutnya adalah tahap anotasi. Anotasi menjadi langkah awal dalam mengidentifikasi objek dalam gambar dengan mendefinisikan fitur-fitur khusus yang merepresentasikan setiap objek. Informasi yang diperoleh dari proses anotasi ini menjadi dasar pembelajaran ketika model menjalani proses pelatihan.

Proses anotasi bertujuan untuk menggali beberapa fitur yang dianggap signifikan dalam dataset. Fitur-fitur tersebut mencakup nama folder (kelas), lokasi penyimpanan dataset, dimensi layer (tinggi, lebar, kedalaman), serta dimensi objek (tinggi dan lebar).

Dalam melaksanakan anotasi, alat *LabelImg* digunakan untuk menempatkan kotak pembatas pada area yang dianggap penting pada objek. Hasil dari proses ini disimpan dalam file dengan ekstensi *.xml* setelah awalnya berupa gambar dalam format *.jpg*. Gambaran visual tentang proses anotasi dapat dilihat pada Gambar 4.3.



Gambar 4.3 *labelling images*

4.1.3 Konfigurasi Dataset

Konfigurasi dataset (*dataset.yaml*) adalah elemen kunci dalam persiapan dataset pelatihan *YOLOv5*. File ini berisi informasi penting yang mengatur dataset gambar yang akan digunakan untuk melatih dan menguji model deteksi objek. Dalam konteks *YOLOv5*, konfigurasi dataset ini sangat penting untuk memahami struktur dataset, jumlah kelas objek yang harus diidentifikasi, dan lokasi penyimpanan gambar-gambar yang akan digunakan selama pelatihan.

Konfigurasi di bawah ini digunakan untuk program pelatihan (*train*) dan validasi (*val*) pada suatu proyek yang mengimplementasikan deteksi bahasa isyarat *BISINDO* menggunakan *YOLOv5*. Lokasi dataset gambar untuk pelatihan dan validasi diatur sesuai dengan struktur direktori berikut:

```
#path: /dataset # dataset root dir
train: /content/drive/MyDrive/Colab Notebooks/Research/dataset/gambar # lokasi direktori gambar pelatihan
val: /content/drive/MyDrive/Colab Notebooks/Research/dataset/gambar # lokasi direktori gambar validasi

# Classes (80 COCO classes)
nc: 5 # jumlah class
names: ['halo', 'kapan', 'maaf', 'sama-sama', 'terimakasih'] # nama kelas yang terdapat pada dataset
```

Gambar 4.4 Konfigurasi Dataset

Konfigurasi ini menyajikan rincian berikut:



1. train: Menunjukkan lokasi direktori dataset gambar yang akan digunakan untuk proses pelatihan model. Dalam kasus ini, direktori tersebut adalah /content/drive/MyDrive/Colab Notebook/Research/dataset/gambar.
2. val: Merupakan direktori dataset gambar yang disiapkan untuk tahap validasi model. Seperti train, lokasi direktori val juga ditetapkan ke /content/drive/MyDrive/Colab Notebook/Research/dataset/gambar.

4.1.4 Training Data

Dataset yang telah ditentukan berupa kordinat-kordinat yang dihasilkan dari *labelling images* yang kemudian di *training* agar yolo dapat mengenal objek yang ditentukan.

```
I | Import torch
from autolablib import pyplot as plt
import numpy as np
import cv2

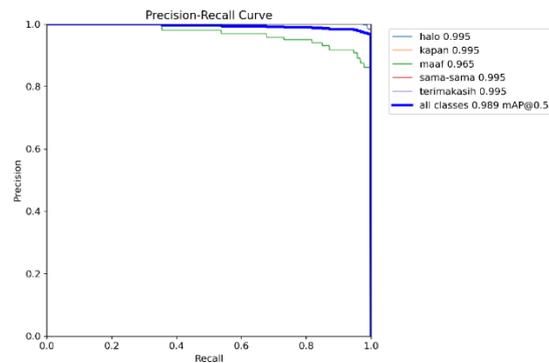
!cd yolov5
!ls
/content/drive/MyDrive/Colab Notebooks/Research/yolov5
benchmarks.py  data  hubconf.py  README.md  segment  utils
CITATION.cff  dataset.yaml  LICENSE  README.zh-CN.md  setup.cfg  val.py
classify  detect.py  models  requirements.txt  train.py  yolov5s.pt
CONTRIBUTING.md  export.py  pytorch  runs  tutorial.ipynb

!python train.py --img 320 --batch 16 --epochs 200 --data dataset.yaml --weights yolov5s.pt --cache

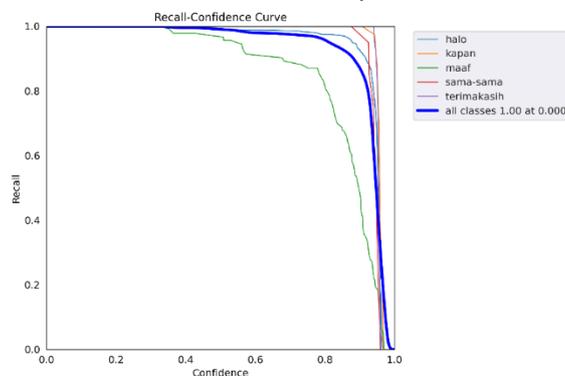
2023-11-03 14:23:08.355792: E tensorflow/compiler/xla/stream_executor/cuda/cuda_fft.cc:68] Unable to register cuFFT factory:
2023-11-03 14:23:04.355792: E tensorflow/compiler/xla/stream_executor/cuda/cuda_blas.cc:151] Unable to register cuBLAS factory:
train: weights=yolov5s.pt, cfg=, data=dataset.yaml, hyp-data=https://github.com/ultralytics/yolov5, epochs=200, batch_size=16, imgsz=320, res
github: ⚠️ yolov5 is out of date by 6 commits. Use 'git pull' or 'git clone https://github.com/ultralytics/yolov5' to update.
fatal: cannot change to '/content/drive/MyDrive/Colab': no such file or directory
YOLOv5 🚀 2023-10-6 Python 3.10.12 torch-2.1.0-cu118 CUDA-0 (Tesla T4, 15102MiB)
```

Gambar 4.5 Menjalankan training YOLO

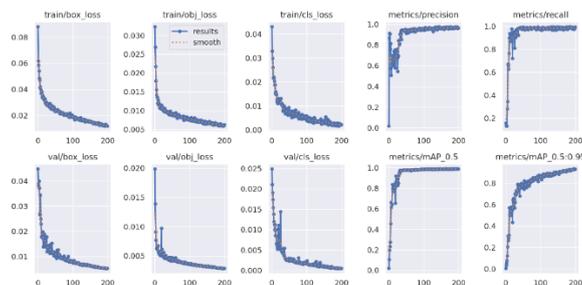
Hasil *training* pada pendeteksian gestur tangan mendapatkan nilai cukup tinggi, nilai *precision* mendapatkan nilai rata-rata 0,989 terhadap nilai *recall*. Nilai puncak *recall* rata-rata mendapatkan nilai 1,00 pada nilai *confidence* 0,00. Berikut terlampir nilai *recall*:



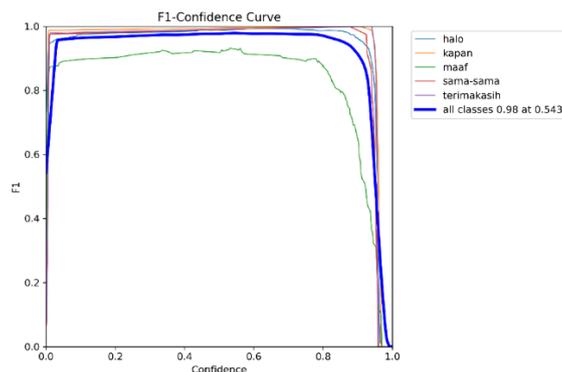
Gambar 4.6 Nilai *precision*



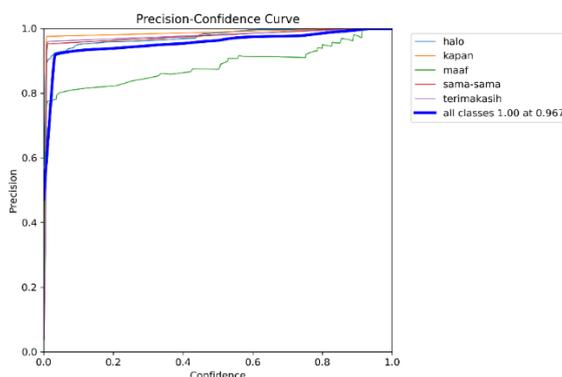
Gambar 4.7 Nilai *recall*



Gambar 4.8 Hasil evaluasi dan data *training*

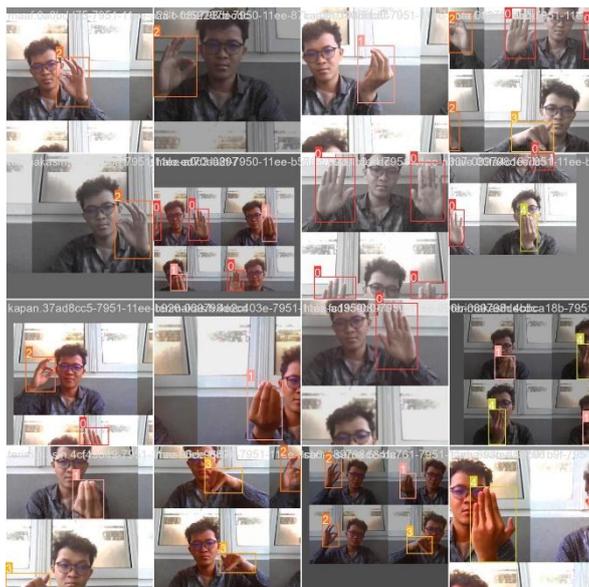


4.9 Kurva nilai F1

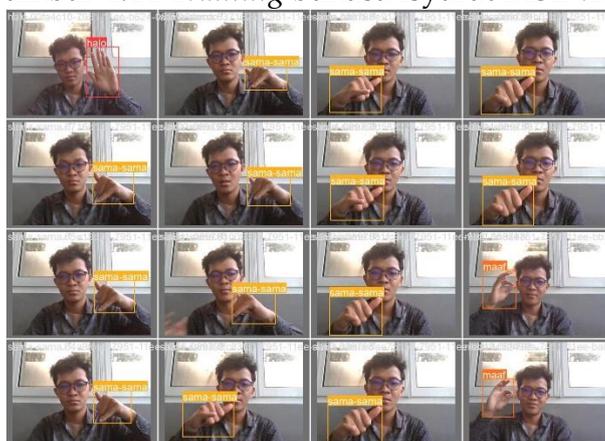


4.10 Precision terhadap nilai confidence

Pada gambar 4.9 dan 4.10 nilai F1 mendapatkan nilai puncak rata-rata 0,98 terhadap nilai *confidence* 0,543. Nilai *precision* mendapatkan nilai rata-rata 1,00 pada nilai *confidence* 0,967.

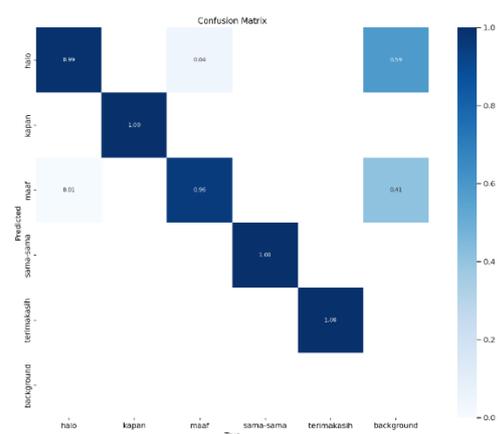


Gambar 4.11 *Training* bahasa isyarat BISINDO



Gambar 4.12 *Testing* bahasa isyarat BISINDO

4.1.5 Pembahasan



Gambar 4.13 *Confusion Matrix*

Hasil TP, FP, FN, dan TN dalam *Confusion Matrix* diatas dihitung sesuai dengan objek yang akan dideteksi, yaitu kata dalam BISINDO halo, kapan, maaf, sama-sama, terimakasih. Untuk menghitung nilai akurasi dari *Confusion Matrix* memakai persamaan dibawah ini:

$$\text{Akurasi} = \frac{TP+TN}{TP+FP+FN+TN} = \frac{(0,99 + 1,00 + 0,96 + 1,00 + 1,00) + (0,01 + 0,04)}{(0,99 + 1,00 + 0,96 + 1,00 + 1,00) + (0,59 + 0,41) + (0) + (0,01 + 0,04)} = 0,83 \text{ atau } 83 \%$$

Jadi pada penelitian menerjemahkan bahasa isyarat menggunakan gesture tangan ini mendapatkan nilai akurasi yang cukup tinggi yaitu nilai akuranya adalah 0,83 atau 83%. Pendeteksian bahasa isyarat menggunakan metode YOLOV5 berjalan dengan lancar dan nilai akurasinya cukup tinggi. Berikut merupakan hasil implementasi model yang sudah di *training*:

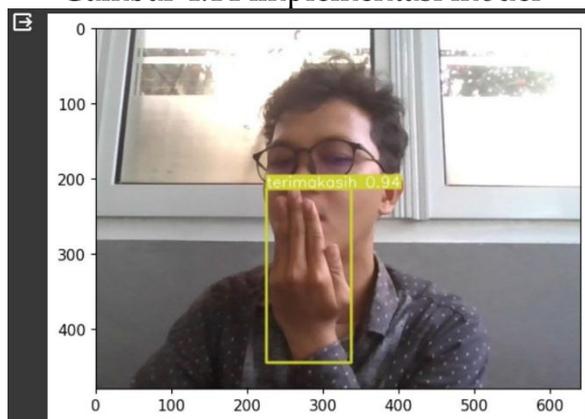
```
cap = cv2.VideoCapture(0)
while cap.isOpened():
    ret, frame = cap.read()

    # Make detections
    results = model(frame)

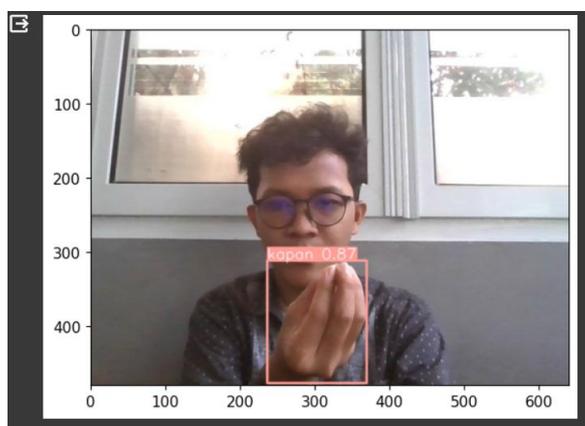
    cv2.imshow('YOLO', np.squeeze(results.render()))

    if cv2.waitKey(10) & 0xFF == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
```

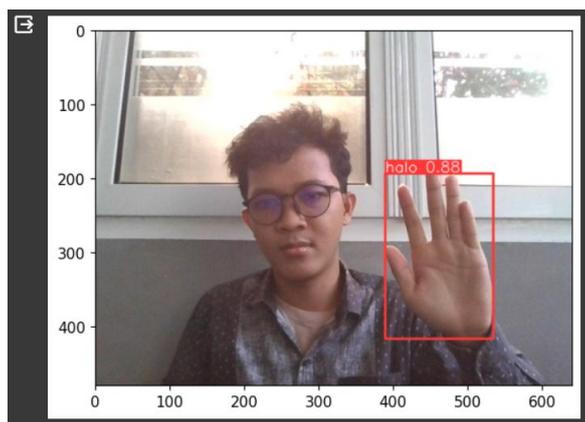
Gambar 4.14 implementasi model



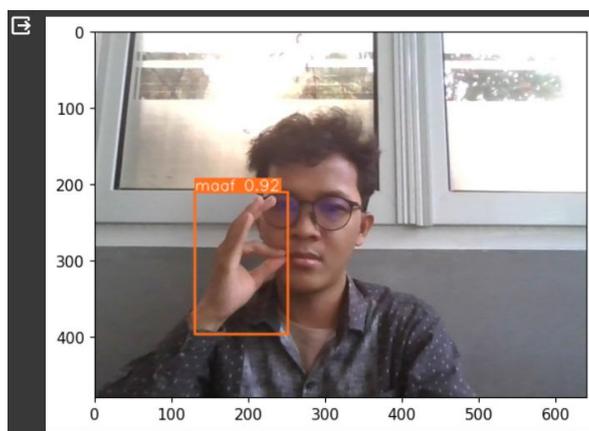
Gambar 4.15 pendeteksian kata terimakasih



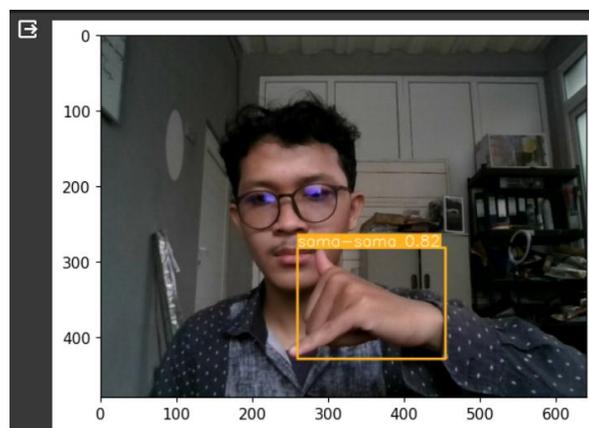
Gambar 4.16 Pendeteksian kata kapan



Gambar 4.17 Pendeteksian kata halo



Gambar 4.18 pendeteksian kata maaf



Gambar 4.19 pendeteksian kata sama-sama

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan dalam penerjemahan bahasa isyarat melalui gestur tangan menggunakan algoritma YOLOV5, maka diperoleh kesimpulan sebagai berikut:

1. Implementasi metode CNN menggunakan algoritma YOLOV5 sudah sangat baik dan berhasil memprediksi bahasa isyarat melalui gesture tangan dengan



menggunakan data training sebanyak 490 gambar dengan presentase keakuratan sebesar 83%

2. Pengujian model didapat akurasi tertinggi sebesar 94%

5.2 Saran

Dari hasil penelitian ini, disarankan untuk meningkatkan akurasi dengan mempertimbangkan penggunaan dataset yang lebih besar dan melakukan iterasi pelatihan yang lebih banyak. Selain itu, optimalisasi penggunaan kamera, penyesuaian jarak dan sudut pengambilan citra, serta proses anotasi citra yang optimal juga dianggap dapat memberikan kontribusi positif terhadap peningkatan akurasi hasil deteksi.

DAFTAR PUSTAKA

- Alfarizi, M. S., Al-farish, M. Z., Taufiqurrahman, M., Ardiansah, G., & Elgar, M. (2023). PENGGUNAAN PYTHON SEBAGAI BAHASA PEMROGRAMAN UNTUK MACHINE LEARNING DAN DEEP LEARNING. *Karimah Tauhid*, 1-6.
- Alfikri, R. H., Utomo, M. S., & Nurwahyudi, E. (2022). Pembangunan Aplikasi Penerjemah Bahasa Isyarat Dengan Metode CNN berbasis Android. *Jurnal Teknoinfo*, 183-197.
- Alfikri, R. H., Utomo, M. S., Februariyanti, H., & Nurwahyudi, E. (2022). Pembangunan Aplikasi Penerjemah Bahasa Isyarat Dengan Metode CNN Berbasis Android. *Jurnal Teknoinfo*, 183-197.
- Aljabar, A., & Suharjito. (2020). BISINDO (Bahasa Isyarat Indonesia) Sign Language Recognition Using CNN and LSTM. *ASTESJ*, 282-287.
- Amrizalm, & Aini, Q. (2013). *Kecerdasan Buatan (Q. Aini, Ed.)*. Jakarta: Halaman Moeka Publishing.
- Arisandi, L. (2022). Sistem Klarifikasi Bahasa Isyarat Indonesia (Bisindo) Dengan Menggunakan Algoritma Convolutional Neural Network. *Jurnal Sistem Cerdas*, 135-146.
- Azhar, K. M., Santoso, I., & Adi S, Y. A. (2021). IMPLEMENTASI DEEP LEARNING MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK DAN ALGORITMA YOLO DALAM SISTEM PENDETEKSI UANG KERTAS RUPIAH BAGI PENYANDANG LOW VISION. *TRANSIENT*, 502-508.
- Dewi, M., Wahyuningrum, T., & Prasetyo, N. A. (2021). Pengenalan Kata Bahasa Isyarat Indonesia (BISINDO) Menggunakan Augmented Reality (AR). *J. Of INISTA*, 53-60.
- Dicoding Intern. (2020, Agustus 19). *Machine Learning*. Diambil kembali dari dicoding: <https://www.dicoding.com/blog/machine-learning-adalah/>
- Fadillah, R. Z., Irawan, A., & Susanty, M. (2022). Model Penerjemah Bahasa Isyarat Indonesia (BISINDO). *PETIR: Jurnal Pengkajian dan Penerapan Teknik Informatika*, 1-9.
- Gustiar, D., Sitorus, S. H., & Midyanti, D. M. (2020). Penerjemah Bahasa Isyarat Menggunakan Metode Generalized Learning Vector Quantization (GLVQ). *Jurnal Komputer dan Aplikasi*, 8(3), 1-8.



- Irawan, F. S. (2020, November 25). *Convolutional Neural Network (CNN)*. Diambil kembali dari kotakode: [https://kotakode.com/blogs/2707/Convolutional-Neural-Network-\(CNN\)](https://kotakode.com/blogs/2707/Convolutional-Neural-Network-(CNN))
- Kemendikbud. (2017). *KBBI Daring*. Diambil kembali dari kbki kemdikbud: <https://kbki.kemdikbud.go.id/entri/bahasa%20isyarat>
- Kemendikbud. (2020, Desember 30). *Kamus SIBI*. Diambil kembali dari pmpk kemdikbud: <https://pmpk.kemdikbud.go.id/sibi>
- Lee, S.-H., & Chen, H.-C. (2021). U-SSD: Improved SSD Based on U-Net. *Applied Sciences*, 1-17.
- Ma'arif, A. (2023). *BUKU AJAR PEMROGRAMAN LANJUT BAHASA PEMROGRAMAN PYTHON*. YOGYAKARTA.
- Mir, R. N., Sharma, V. K., Rout, R. K., & Umer, S. (2023). *Advancement of Deep Learning and its Applications in Object Detection and Recognition*. River Publishers.
- Nada, M. (2019, January 02). *Penerapan Deep Learning Menggunakan Convolutional Neural Network (CNN)*. Diambil kembali dari medium: <https://medium.com/@mukhlisatunnada02/penerapan-deep-learning-menggunakan-convolutional-neural-network-cnn-d02dc6532f5b>
- Nelson, J., & Solawetz, J. (2020, juni 10). *YOLOv5 is Here: State-of-the-Art Object Detection at 140 FPS*. Diambil kembali dari Roboflow: <https://blog.roboflow.com/yolov5-is-here/>
- 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 (JTIK)*, 819-828.
- Nurhikmat, T., & Purwaningsih, T. (2018). ResearchGate. *Implementasi Deep Learning Untuk Image Classification Menggunakan Algoritma Convolutional Neural Network (CNN) Pada Citra Wayang Golek*, 8 - 81.
- Prasetyo, B., Sari, M. M., Redjeki, W., Hendrayanto, D., & Setyohadi, I. S. (2018). *Buku Saku Bahasa Isyarat*. Jakarta.
- Rachardi, F. (2020). ResearchGate. *Deteksi Gambar Gestur Kosakata Bahasa Isyarat Indonesia dengan Convolutional Neural Network*, 15 - 39.
- Rahmadani, D., & Kasiyati. (2023). Efektivitas Mengenal Huruf Vokal Menggunakan Media Pembelajaran Pin Activity . *Jurnal Penelitian Pendidikan Khusus*, 1-8.
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779-788.
- Rosebrock, A. (2016, November 7). *Intersection over Union (IoU) for object detection*. Diambil kembali dari Pyimagesearch: <https://pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/>
- Santoso, A., & Ariyanto, G. (2018). Implementasi Deep Learning Berbasis Keras Untuk Pengenalan Wajah. *Emitor: Jurnal Teknik Elektro*, 15-21.



- Santoso, R. R., Megasari, R., & Hambali, Y. A. (2020). Implementasi Metode Machine Learning Menggunakan Algoritma Evolving Artificial Neural Network Pada Kasus Prediksi Diagnosa Diabetes. *JATIKOM*, 85-97.
- Shianto, K. A., Gunadi, K., & Setyati, E. (2019). Deteksi Jenis Mobil Menggunakan Metode YOLO Dan. *Jurnal Infra*, 157-163. Diambil kembali dari Pyimagesearch.
- Solawetz, J. (2020, juni 29). *Yolov5 Improvements And Evaluation*. Diambil kembali dari roboflow: <https://blog.roboflow.com/yolov5-improvements-and-evaluation/>
- Suartika, W., Wijaya, A. Y., & Soelaiman, R. (2016). Klasifikasi Citra Menggunakan Convolutional. *JURNAL TEKNIK ITS*, 65-69.
- Syahrudin, A. N., & Kurniawan, T. (2018). Input dan Output Pada Bahasa Pemrograman Python. *Jurnal Dasar Pemrograman Python STMIK*, 1-7.
- Tim Teaching Kecerdasan Buatan. (2012). *Modul Bahan Ajar Kecerdasan Buatan*. Malang: Universitas Brawijaya.
- Wahyono, Y. T. (2020). Machine Learning Konsep dan Implementasi. *ResearchGate*, 1-10.
- Wasril, A. R., Ghozali, M. S., & Mustafa, M. B. (2019). Pembuatan Pendeteksi Obyek dengan Metode You Only Look Once (YOLO) untuk Teller Automated Teller Machine (ATM). *Majalah Ilmiah UNIKOM Vol 17 no 1*, 69-75.