

# Pengenalan Objek Alfabet *American Sign Language* (ASL) Menggunakan Algoritma YOLOv5

1<sup>st</sup> Rishma MaudynaUniversitas Buana Perjuangan Karawang  
Karawang, Indonesia  
if18.rishmamaudyna@mhs.ubpkarawang.ac.id2<sup>nd</sup> Hanny Hikmayanti H.Universitas Buana Perjuangan Karawang  
Karawang, Indonesia  
hanny.hikmayanti@ubpkarawang.ac.id3<sup>rd</sup> Santi Arum Puspita LestariUniversitas Buana Perjuangan Karawang  
Karawang, Indonesia  
santi.arum@ubpkarawang.ac.id

**Abstract—** Kurangnya penggunaan dalam berkomunikasi menggunakan bahasa isyarat mengakibatkan minimnya pengetahuan terkait metode komunikasi menggunakan bahasa isyarat. Penggunaan model algoritma YOLOv5 banyak digunakan untuk deteksi objek dan menghasilkan akurasi yang hampir tepat, maka dari itu penelitian ini dibuat untuk mendeteksi pengenalan objek alfabet dari *American Sign Langauge* (ASL) dengan menggunakan YOLOv5. Tujuan penelitian ini adalah untuk melihat hasil deteksi dan analisa performa model YOLOv5 serta hasil evaluasi dari *Confusion Matrix*. Tahapan penelitian yang dilakukan adalah pengumpulan dataset dan anotasi objek menggunakan *Roboflow*. Untuk melihat performa model YOLOv5 dibuat perbandingan antara *epoch* 200 dan 300. Hasil model YOLOv5 dari kedua *epoch* tersebut menunjukkan dari 26 kelas huruf alfabet nilai *Mean Avarege Precision* (mAP) *confidence threshold* rata-rata mencapai 0.950 mAP@0.5. Hasil *precision* pada 200 dan 300 *epoch* didapatkan yaitu sebesar 1 dan 0.7428. Hasil *recall* pada 200 dan 300 *epoch* didapatkan sebesar 0.7878 dan 0.7027. Pada *epoch* 200 dan 300 didapatkan nilai akurasi terbesar mencapai 0.96, sedangkan nilai kecil terdapat pada *epoch* 200 dengan nilai 0.85.

**Kata kunci —** Bahasa Isyarat, Confusion Matrix, Deteksi Objek, YOLOv5

## I. PENDAHULUAN

Bahasa Isyarat atau *sign language* berguna dalam kehidupan sehari-hari karena sistem dapat digunakan sebagai alat komunikasi untuk memahami maksud dari lawan bicara dengan mengenali pola huruf yang dibentuk jari sebagai penanda pembentukan kata [1]. *American Sign Language* (ASL) adalah salah satu bahasa isyarat yang berasal dari Amerika. Ada batasan kemampuan orang yang mampu berkomunikasi menggunakan bahasa isyarat [2].

Deteksi objek termasuk salah satu dari *deep learning* yang merupakan pembelajaran *computer vision*. Implementasi model *deep learning* untuk deteksi objek telah banyak digunakan untuk kepentingan umum sampai edukasi pengetahuan contoh seperti deteksi pengenalan pola bentuk wajah dan deteksi gerak lainnya. YOLOv5 merupakan model pendekripsi objek generasi kelima yang rilis pada April 2020 oleh Glenn Jocher pada *platform Github* [3]. Ide inti dari YOLO adalah menggunakan seluruh gambar sebagai input jaringan, dan langsung mengembalikan posisi kotak pembatas dan kelas kotak pembatas pada *output* [3].

Pada penelitian sebelumnya yang berjudul Implementasi Model *Deep Learning* Untuk Deteksi Objek Candi Prambanan, Candi Borobudur, Dan Candi Ratu Boko Menggunakan YOLOv5 yang dilakukan oleh Levina Anora [3]. Penelitian ini menggunakan YOLOv5 dengan *AI Project Life Cycle* sebagai metode kerja untuk membuat projek *AI*. Hasil *Precision*, *Recall*, dan Akurasi pada hasil uji data *testing* masing-masing adalah sebesar 95.9%, 81.2%, 78.5%. Keakuratan hasil pengujian dipengaruhi oleh jumlah FN yang dihasilkan dalam pendekripsi, hasil akurasi pendekripsi target paling kecil adalah Candi Borobudur yaitu 69,8%, sedangkan tingkat akurasi Candi Prambanan sebesar 91,7%, akurasi terbesar adalah Candi Ratu Boko yaitu sebesar 96,1%. Penelitian selanjutnya berjudul Deteksi dan Pengenalan *American Sign Language* Menggunakan Algoritma YOLOv5 oleh Tasnim Ferdous Dima dan MD [4]. Penelitian ini menggunakan *Roboflow* untuk pengumpulan dan *labeling* dataset. Hasil penelitian mendapatkan 0.987 untuk mAP pada model YOLOv5.

Berdasarkan latar belakang model algoritma YOLOv5 banyak digunakan untuk deteksi objek dan menghasilkan akurasi yang hampir tepat. Model YOLOv5 akan lebih optimal jika menggunakan *Roboflow*, maka dari itu penelitian ini dibuat dengan *American Sign Languge* (ASL) sebagai objek deteksi. Tujuan penelitian ini adalah untuk mengevaluasi model dari algoritma Yolov5 pada objek simbol tangan terhadap bahasa isyarat *American Sign Language* atau ASL. Diharapkan penelitian ini dapat membantu memahami arti dari bahasa isyarat dan menghasilkan nilai akurasi yang tepat.

## II. TINJAUAN PUSTAKA

### A. American Sign Language (ASL)

*American Sign Language* (ASL) adalah bahasa isyarat yang memiliki sifat linguistik yang sama dengan bahasa lisan yang diungkapkan melalui gerak tangan dan tubuh [5]. ASL banyak digunakan oleh tuna rungu di Amerika utara sebagai bahasa komunikasi utama. ASL juga digunakan di Kanada, Meksiko, Afrika Barat, dan Asia. Lebih dari 20 negara lain di mana bahasa Inggris adalah bahasa komunikasi utama, termasuk Jamaika, Panama, Thailand, dan Malaysia, menggunakan ASL untuk komunikasi utama untuk komunitas tunarungu [6].

### B. Intersection over Union (IoU)

Untuk deteksi objek, *Intersection over Union* (IoU) mengukur area tumpang tindih antara kotak pembatas yang diprediksi Bp dan kotak pembatas *ground-truth* Bgt dibagi dengan luas unit diantara keduanya [7]. Dari berbagai kumpulan data beranotasi yang

digunakan dalam deteksi objek *Mean Average Precision* (mAP) adalah nilai rata-rata dari *Average Precision* (AP) yang merupakan metrik yang paling umum digunakan untuk pengukuran akurasi deteksi [8].

Nilai mAP diformulasikan sebagai berikut:

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i$$

Dimana  $AP_i$  adalah AP untuk kelas ke- $i$  dan  $N$  adalah jumlah total kelas yang dievaluasi [7]. Konsep umum untuk menghitung nilai *precision* maupun *recall* biasanya dipetakan dalam bentuk *confusion matrix* [3].

### C. Confusion Matrix

*Confusion Matrix* merupakan salah satu metode yang digunakan untuk mengevaluasi metode klasifikasi [9]. Gambaran pada Tabel 2.1 untuk mempermudah pemahaman tentang istilah *confusion matrix* dalam keluaran klasifikasi.

Tabel 1 Istilah *Confusion Matrix*

		Kelas Aktual ( <i>Expectation</i> )	
		<i>Positive</i>	<i>Negative</i>
Kelas Prediksi ( <i>Observation</i> )	<i>Positive</i>	TP	FP
	<i>Negative</i>	FN	TN

Keterangan:

TP = *True Positive*

TN = *True Negatif*

FP = *False Positive*

FN = *False Negative*

Berikut rumus persamaan *Precision*, *Recall*, dan *Akurasi*:

$$\textbf{Precision} = \frac{\sum_i^n \frac{TP_i}{TP_i + FP_i}}{n} = \frac{TP}{TP + FP} \quad (1)$$

$$\textbf{Recall} = \frac{\sum_i^n \frac{TP_i}{TP_i + FN_i}}{n} = \frac{TP}{TP + FN} \quad (2)$$

$$\textbf{Akurasi} = \frac{\sum_i^n \frac{TP_i + TN_i}{TP_i + FP_i + TN_i + FN_i}}{n} = \frac{TP}{TP + FP + FN} \quad (3)$$

### D. Algoritma YOLO

YOLOv5 ditulis dalam bahasa pemrograman *python* dan *library PyTorch*. YOLO (*You Only Look Once*) adalah mengambil seluruh gambar sebagai input jaringan dan langsung mengembalikan posisi kotak pembatas dan kelas kotak pembatas pada *output* [3]. Kerangka kerja YOLO adalah *wide* dan *height* untuk memprediksi lebar dan tinggi seluruh gambar (masing-masing relatif terhadap seluruh gambar) [10]. YOLO terdiri dari tiga bagian utama [3] yaitu:

1. *Backbone* : Jaringan saraf convolutional yang menggabungkan detail berbagai gambar untuk membentuk fitur gambar.
2. *Neck* : Serangkaian lapisan jaringan mencampur dan menggabungkan fitur gambar dan masuk ke lapisan prediksi.
3. *Head* : Komponen ini dapat memprediksi fitur gambar, menghasilkan kotak pembatas, dan memprediksi kategori. Keyakinan mewakili keakuratan klarifikasi dalam kondisi tertentu

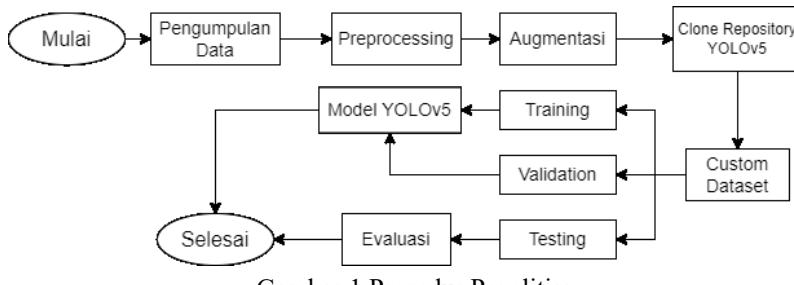
Dengan mempelajari secara otomatis *anchor box* terbaik pada kumpulan data apapun, model YOLOv5 tidak perlu mempertimbangkan parameter ini sebagai masukan. Karena itu, model dapat belajar langsung dari kumpulan data dan berlatih secara otomatis [11].

### E. Roboflow

*Roboflow* adalah sebuah *platform* yang dirancang khusus untuk membantu AI *engineer* dengan kumpulan data aplikasi untuk proyek *computer vision* [3]. Pada *roboflow* terdapat 20 library model untuk *computer vision* terdiri dari YOLOX, *Vision Transformer*, YOLOv5, YOLOR, *OpenAI Clip*, *Scaled-YOLOv4*, Resnet32, *EfficientNet*, *Faster R-CNN*, YOLO v3 *PyTorch*, YOLO v3 *Keras*, *MobileNetSSDv2*, YOLOv4, *PyTorch*, *MobileNetV2 Classification*.

### F. Metode Penelitian

Tahapan penelitian yang dilakukan pada penelitian ini adalah pengumpulan dataset, Anotasi objek menggunakan *Roboflow*, *clone repository* YOLOv5, custom dataset COCO, *training* dan *validation* pada proses model YOLOv5, terakhir *testing* untuk mengevaluasi hasil dengan menggunakan bahasa pemrograman *Python* pada *Google Colaboratory* yang digambarkan seperti berikut:



Gambar 1 Prosedur Penelitian

## III. HASIL DAN PEMBAHASAN

## A. Pengumpulan Data

Data gambar diambil dari berbagai kondisi pencahayaan dan teknik pengambilan posisi gambar untuk menghasilkan postur gambar. Dataset yang terkumpul sebanyak 1728 gambar *close-up* berwarna dan hitam putih terdiri dari 26 kelas yaitu huruf abjad A sampai Z. Data penelitian akan dibagi menjadi 3 bagian terdiri dari *train set* 1512 gambar dengan berbagai kondisi gambar berwarna, *close-up*, dan hitam putih, *validation set* 144 gambar dengan kondisi gambar *close-up* dan berwarna, dan *test set* dengan 72 gambar. Berikut merupakan contoh gambar ASL berwarna dan hitam putih.

## B. Anotasi Objek

Anotasi objek atau pelabelan objek adalah proses menandakan dengan label pada gambar dengan cara memberikan kotak pembatas (*bounding box*) beserta nama kelas pada objek. Anotasi objek dilakukan dalam dua proses yaitu *preprocessing* data dan *augmentasi* data. Pada proses *pre-processing* untuk melatih model menggunakan *platform Roboflow* <https://roboflow.com>. Berikut rincian tahap *preprocessing* pada data ASL



Gambar 2 Contoh ASL Berwarna dan Hitam Putih

1. *Auto-Orient* sumber data teraplikasikan dengan otomatis pada *roboflow*
2. *Resize* sumber data yang berukuran *stretch to*  $416 \times 416$
3. *Grayscale* sumber data mengaplikasikan sebesar 10% dari gambar. Mengoversikan gambar dengan saluran RGB menjadi gambar dengan saluran skala abu-abu tunggal, yang dapat menghemat memori. Nilai setiap piksel skala abu-abu dihitung sebagai jumlah bobot piksel *red*, *green*, dan *blue* yang sesuai dengan referensi dari fosfor CRT yaitu  $Y = 0.2125 R + 0.7154 G + 0.0721 B$

Kualitas kebijakan augmentasi teratas pada dataset COCO yang kompetitif pada backbone yang berbeda arsitektur dan algoritma deteksi. Augmentasi untuk menerjemahkan gambar dengan beberapa piksel sehingga menghasilkan pengklasifikasi gambar. Arsitektur dasar YOLOv5 menggunakan CSPDarkNet dan augmentasi data standar teknik yang sebagian besar disesuaikan dengan pelatihan klasifikasi gambar. Terdiri dari flipping horizontal dengan mengaplikasi grayscale 10% dimana gambar resize sebesar  $460 \times 460$ .

Gambar 3 Anotasi Objek Pada *Roboflow*

### C. Clone Repository YOLOv5

Untuk menerapkan model YOLOv5 maka diperlukan untuk clone repository. *Clone Repository* YOLOv5 dapat dilakukan pada github <https://github.com/ultralytics/yolov5> dibuat dalam *platform Google Colab*. Cloning YOLOv5 menghasilkan *library* dan data COCO dimana data tersebut akan diubah menjadi data klasifikasi huruf ASL. Dengan memanggil perintah *library* dari *pytorch* untuk pemakaian GPU dan CPU. Berikut proses *clone repository* YOLOv5 pada *platform google colab*.

```
[2] !git clone https://github.com/ultralytics/yolov5 # clone
%cd yolov5
!pip install -qr requirements.txt # install

import torch
import utils
display = utils.notebook_init() # checks

YOLOv5 v6.1-344-g0e165c5 Python-3.7.13 torch-1.12.0+cu113 CUDA:0 (Tesla T4, 15110MiB)
Setup complete (2 CPUs, 12.7 GB RAM, 38.6/78.2 GB disk)
```

Gambar 4 Clone Repository pada *Google Colab*

### D. Model YOLOv5

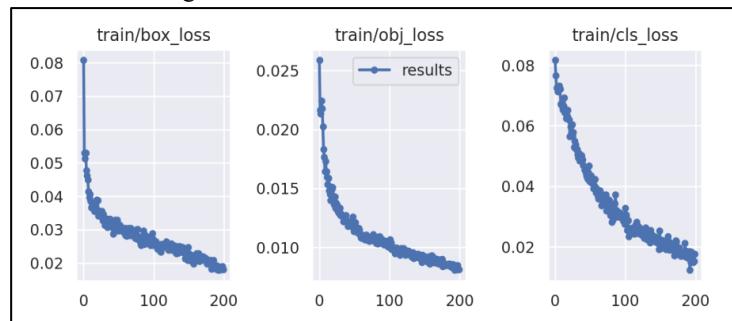
Tahap model YOLOv5 menggunakan model terlatih untuk pelatihan model. Selama pelatihan model, konfigurasi dengan tabel berikut telah ditetapkan. [3].

Table 2 Konfigurasi Pelatihan Model

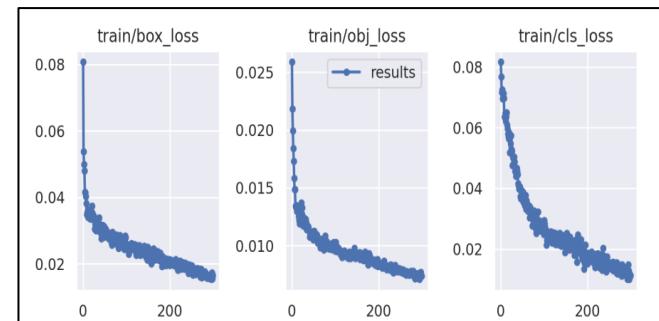
<i>Image Size</i>	<i>Batch Size</i>	<i>Epoch</i>	<i>Data Source</i>
620	32	200, 300	Data.yaml

Untuk menemukan *epoch* yang optimal dikembangkannya model training dalam menggunakan YOLOv5 dengan dataset alfabet ASL 26 kelas dijalankan dengan epoch yang berbeda. Proses validasi disediakan precision dan recall dari setiap kelas individu [12]. Proses *training* dan *validation* dimana proses pengimplementasi model YOLOv5 terjadi.

#### 1. Training



Gambar 5 (a) Hasil *Training epoch 200 batch 32*

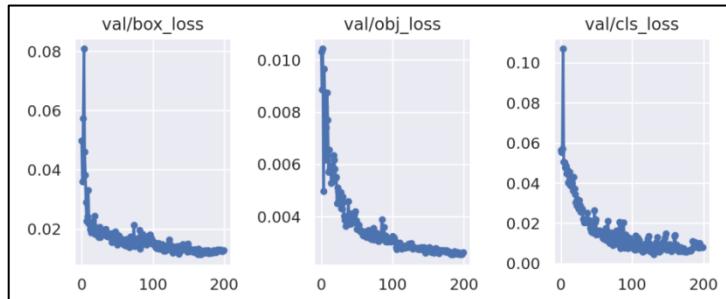


Gambar 5 (b) Hasil *Training epoch 300 batch 32*

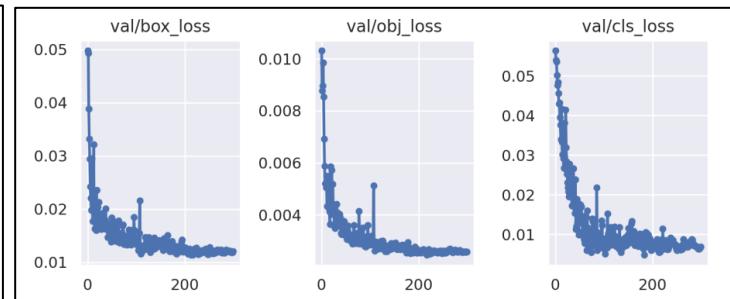
*Box\_loss* atau hilangnya bingkai deteksi menunjukkan apakah suatu algoritma dapat menemukan pusatnya titik objek dengan baik dan apakah target deteksi tercakup oleh batas yang diprediksi kotak. Semakin kecil nilai fungsi kerugian, semakin akurat bingkai prediksi. Pada dasarnya *Classification loss* adalah ukuran probabilitas bahwa target deteksi ada di wilayah yang diminati. Semakin kecil nilai fungsi kerugian, semakin tinggi akurasinya. *Classification loss* mewakili kemampuan algoritma untuk memprediksi dengan benar suatu yang diberikan kategori objek. Semakin kecil nilai kerugian, semakin akurat klasifikasinya.

Pada Gambar 5 (a) menujukkan hasil nilai fungsi *loss* menurun selama pelatihan. Selama pelatihan, sebelum mencapai *epoch* ke-80 menujukkan nilai fungsi *loss* menurun dengan cepat. Jaringan model terus jalan selama proses pelatihan, hasil keseluruhan menujukkan menurun dengan lambat hingga mencapai *epoch* akhir. Hasil Gambar 5 (b) juga sama menunjukkan nilai fungsi menurun dengan cepat pada awal pelatihan hingga mencapai *epoch* ke-100 mulai menurun dengan lambat. Model jaringan pada dasarnya mencapai keadaan konvergensi, dan bobot jaringan yang optimal diperoleh pada akhir pelatihan.

#### 2. Validation



Gambar 6 (a) Hasil *Validation epoch 200 batch 32*



Gambar 6 (a) Hasil *Validation epoch 300 batch 32*

*Box\_loss* atau hilangnya bingkai deteksi menunjukkan apakah suatu algoritma dapat menemukan pusatnya titik objek dengan baik dan apakah target deteksi tercakup oleh batas yang diprediksi kotak. Semakin kecil nilai fungsi kerugian, semakin akurat bingkai prediksi. Pada dasarnya *Classification loss* adalah ukuran probabilitas bahwa target deteksi ada di wilayah yang diminati. Semakin kecil nilai fungsi kerugian, semakin tinggi akurasinya. *Classification loss* mewakili kemampuan algoritma untuk memprediksi dengan benar suatu yang diberikan kategori objek. Semakin kecil nilai kerugian, semakin akurat klasifikasinya.

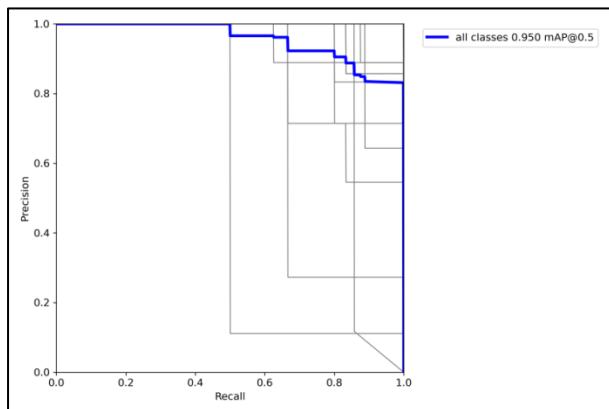
Pada Gambar 6 (a) menunjukkan hasil nilai fungsi *loss* menurun selama validasi berlangsung. Selama proses berjalan nilai fungsi *loss* menurun stabil sebelum *epoch* ke-10. Tren mulai menurun lambat pada *epoch* ke-50 dan menunjukkan hasil yang tidak stabil. Jaringan model terus berjalan selama proses validasi model. Hingga *epoch* akhir hasil tetap menunjukkan ketidakstabilan. Pada Gambar 6 (b) menunjukkan nilai fungsi *loss* yang menurun lambat dengan tidak stabil. Nilai fungsi *loss* yang tidak stabil disebabkan karena data gambar pada *validation set* lebih sedikit dari *train set*. Inti dari hasil tren menunjukkan nilai fungsi *loss* tetap menurun dan hasil klasifikasi terdeteksi.

### 3. Testing

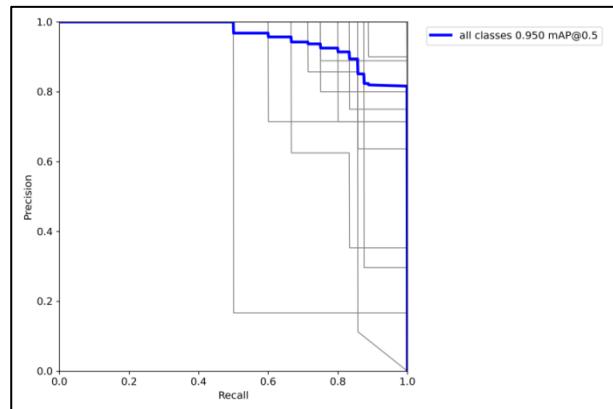
Proses *testing* dilakukan evaluasi terhadap hasil mAP, *Precision*, *recall*, dan akurasi pada *epoch* yang berbeda. Hasil *testing* berupa perbandingan *batch* dan *epoch* pada proses *training* dan *validation* model YOLOv5.

#### E. Evaluasi

##### a. Mean Average Precision (mAP)



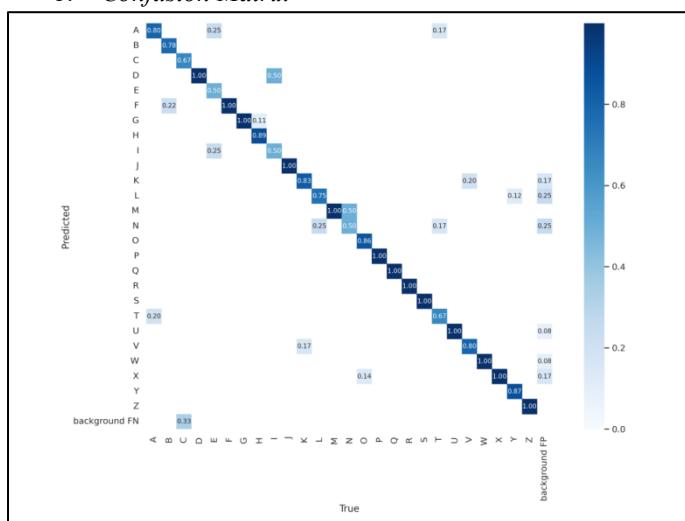
Gambar 7 (a) Hasil mAP Evaluasi Model *epoch* 200  
*batch* 32



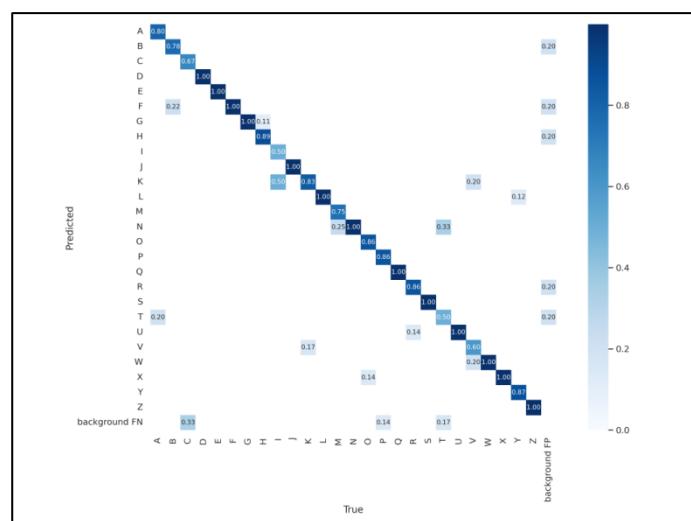
Gambar 7 (b) Hasil mAP Evaluasi Model *epoch* 300  
*batch* 32

Pada perbandingan Gambar 4.7 (a) dan (b) dari 26 kelas huruf alfabet nilai mAP yang terdiri dari *epoch* 200 dan 300 menunjukkan hasil dengan *confidence threshold* rata-rata mencapai 0.950 mAP@0.5.

##### b. Confusion Matrix



Gambar 8 (a) *Confusion Matrix* *epoch* 200 *batch* 32



Gambar 8 (b) *Confusion Matrix* *epoch* 300 *batch* 32

Gambar 3 (a) Hasil Perhitungan *Confusion Matrix epoch 200 batch 32*

Alfabet	Confusion Matriks		
	TP	FP	FN
A	1	2	0
B	1	0	0
C	1	0	0
D	1	1	0
E	1	0	0
F	1	0	1
G	1	1	0
H	1	0	0
I	1	0	1
J	1	0	0
K	1	2	0
L	1	2	0
M	1	1	0
N	1	0	1
O	1	0	0
P	1	0	0
Q	1	0	0
R	1	0	0
S	1	0	0
T	1	0	1
U	1	1	0
V	1	0	1
W	1	1	0
X	1	1	1
Y	1	0	0
Z	1	0	0
Jumlah	26	14	7

Gambar 3 (b) Hasil Perhitungan *Confusion Matrix epoch 300 batch 32*

Alfabet	Confusion Matriks		
	TP	FP	FN
A	1	0	1
B	1	1	0
C	1	0	1
D	1	0	0
E	1	0	0
F	1	1	1
G	1	1	0
H	1	1	0
I	1	0	0
J	1	0	0
K	1	1	1
L	1	1	0
M	1	0	0
N	1	1	1
O	1	0	0
P	1	0	1
Q	1	0	0
R	1	1	0
S	1	0	0
T	1	1	2
U	1	0	1
V	1	0	1
W	1	0	1
X	1	0	1
Y	1	0	0
Z	1	0	0
Jumlah	26	9	11

Berikut hasil *confusion matrix* yang didapatkan dari perbandingan kedua *epoch*

#### c. Precision dan Recall

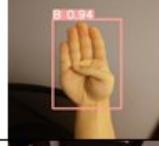
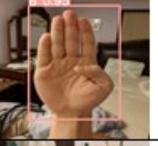
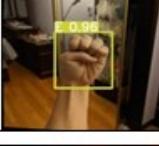
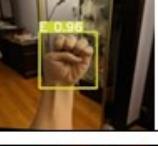
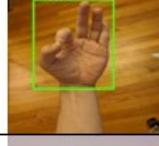
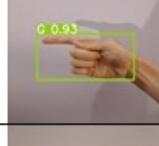
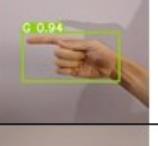
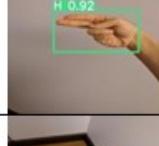
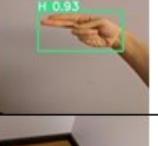
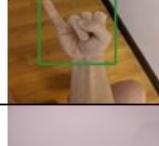
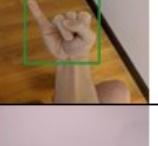
Didapatkan hasil dari tabel perhitungan *confusion matrix* dijumlahkan dengan persamaan *precision* dan *recall* untuk mengetahui hasil evaluasi objek.

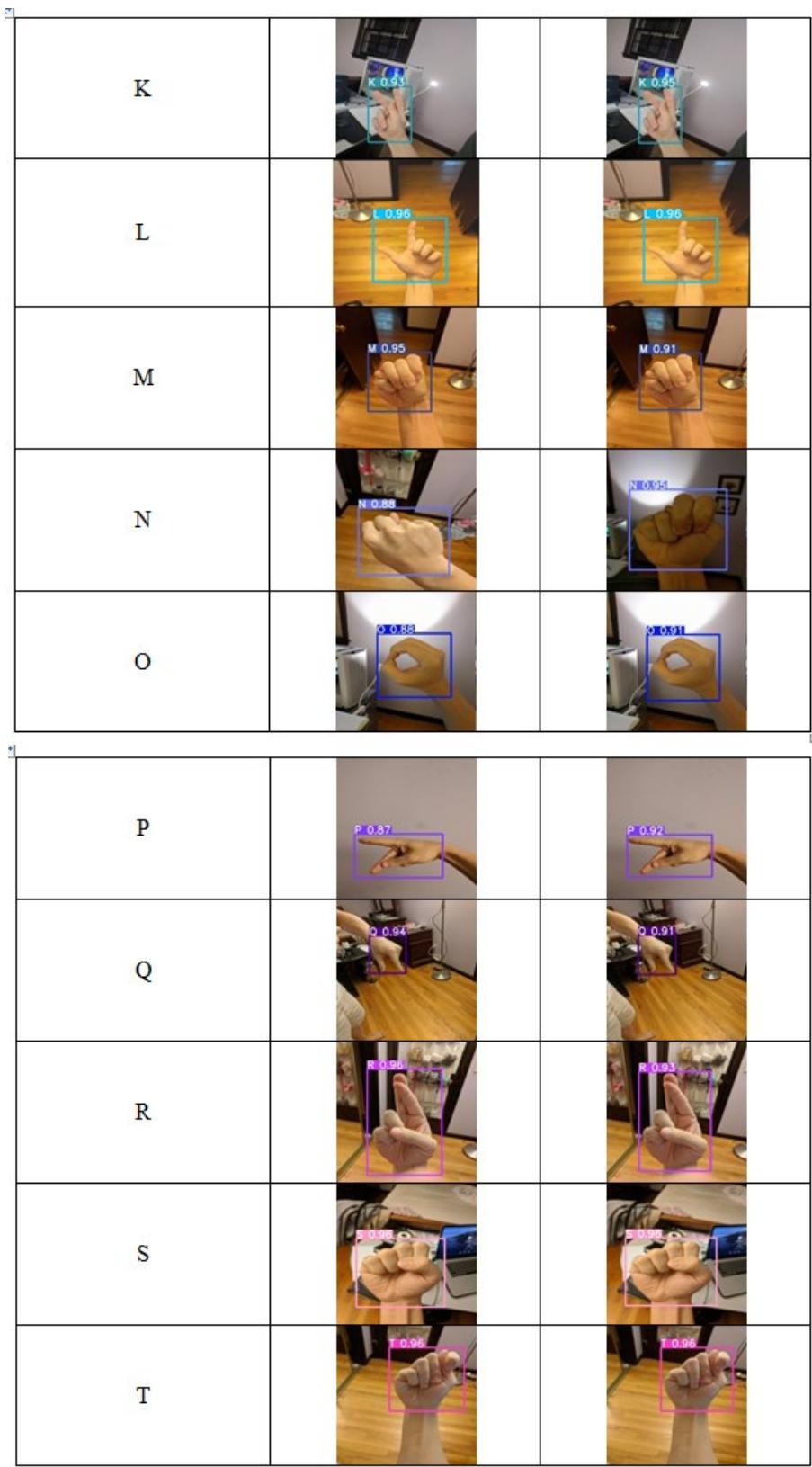
Tabel 3 Hasil Perhitungan Perbandingan *Confusion Matrix*

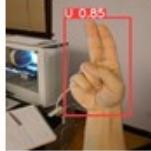
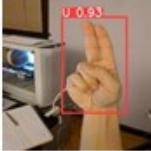
	Precision	Recall
Epoch 200 batch 32	1	0.7878
Epoch 300 batch 32	0.7428	0.7027

#### d. Akurasi

Hasil pengujian pada akurasi berupa deteksi gambar tangan simbol dari *alfabet ASL (american sign language)*. Evaluasi akurasi menggunakan data dari *testing set* yang berisi gambar sebanyak 72 gambar. Hasil perbandingan evaluasi akurasi dapat dilihat pada tabel berikut.

<i>Alfabet American Sign Language (ASL)</i>	<i>Epoch 200 Batch 32</i>	<i>Epoch 300 Batch 32</i>
A		
B		
C		
D		
E		
F		
G		
H		
I		
J		



U				
V				
W				
X				
Y				
Z				

Gambar 8 Hasil Perbandingan Akurasi

Hasil akurasi setiap kelas huruf pada *epoch* yang berbeda memiliki nilai akurasi yang sedikit berbeda. Pada 200 *epoch* nilai akurasi yang terkecil ada pada kelas huruf "U" yaitu dengan hasil akurasi 0.85, sedangkan nilai akurasi yang terbesar mencapai 0.96 terdapat pada banyak kelas. Pada 300 *epoch* nilai akurasi yang terkecil ada pada kelas huruf "J" yaitu dengan hasil akurasi 0.88, sedangkan nilai akurasi yang terbesar mencapai 0.96 terdapat pada banyak kelas. Adapun yang memiliki nilai akurasi yang sama terdapat pada kelas huruf "E", "F", "L", "T".

#### IV. KESIMPULAN DAN SARAN

##### A. Kesimpulan

Selama *training* model dilakukan perbandingan antara 200 dan 300 *epoch*. Selama proses *training* dan validasi berjalan hasil menunjukkan nilai fungsi *loss* menurun yang menandakan semakin kecil nilai fungsi *loss* nya dengan pergerakan yang tidak stabil, yang artinya semakin kecil nilai fungsi *loss*, semakin tinggi akurasinya. Dari hasil pelatihan model, semakin besar nilai *epoch* maka semakin baik nilai *loss* dan akurasinya. Dari 26 kelas huruf nilai *Mean Average Precision* (*mAP*) yang terdiri dari 200 dan 300 *epoch* menunjukkan hasil dengan *confidence threshold* rata-rata mencapai 0.950 *mAP@0.5*. Hasil *mAP* dipengaruhi oleh nilai *precision*, artinya semakin baik nilai presisi maka semakin baik nilai *mAP*. Sedangkan hasil perhitungan *precision* pada 200 dan 300 *epoch* didapatkan yaitu sebesar 1 dan 0.7428. Hasil perhitungan *recall* pada 200 dan 300 *epoch* didapatkan sebesar 0.7878 dan 0.7027. Hasil akurasi setiap kelas huruf pada *epoch* yang berbeda memiliki nilai akurasi yang sedikit berbeda. Pada *epoch* 200 dan 300 didapatkan nilai akurasi terbesar mencapai 0.96, sedangkan nilai kecil terdapat pada *epoch* 200 dengan nilai 0.85.

##### B. Saran

Berdasarkan penelitian yang telah dilakukan, saya memberikan saran untuk penelitian di masa depan, antara lain:

1. Karena sample perbandingan *epoch* dan *batch* yang terbatas, disarankan membandingkan dengan sampel yang cukup hingga mendapat hasil yang lebih baik.
2. Menggunakan model algoritma tambahan agar mendapatkan hasil yang lebih optimal.

## PENGAKUAN

Naskah ilmiah ini adalah sebagian dari penelitian Tugas Akhir milik Rishma Maudyna, dengan judul Pengenalan Objek Alfabet American Sign Language (ASL) menggunakan Algoritma YOLOv5 yang dibimbing oleh Hanny Hikmayanti Handayani dan Santi Arum Puspita Lestari.

## DAFTAR PUSTAKA

- [1] M. E. Al Rivan, H. Irsyad, K. Kevin, and A. T. Narta, “Pengenalan Alfabet American Sign Language Menggunakan K-Nearest Neighbors Dengan Ekstraksi Fitur Histogram Of Oriented Gradients,” *J. Tek. Inform. dan Sist. Inf.*, vol. 5, no. 3, pp. 328–339, 2020, doi: 10.28932/jutisi.v5i3.1936.
- [2] R. Ridwang, “Pengenalan bahasa isyarat indonesia (SIBI) menggunakan leap motion controller dan algoritma data mining naïve bayes,” *J. INSYPRO (Information Syst. Process.)*, vol. 2, no. 2, 2017.
- [3] A. Levina, “Implementasi Model Deep Learning Untuk Deteksi Objek Candi Prambanan, Candi Borobudur, Dan Candi Ratu Boko Menggunakan YOLO V5.” Institut Teknologi Telkom Purwokerto, 2021.
- [4] T. F. Dima and M. E. Ahmed, “Using YOLOv5 Algorithm to Detect and Recognize American Sign Language,” in *2021 International Conference on Information Technology (ICIT)*, 2021, pp. 603–607.
- [5] P. Studi and T. Informatika, “Klasifikasi American Sign Language Menggunakan Fitur Scale Invariant Feature Transform dan Jaringan Saraf Tiruan,” vol. 1, no. 1, 2020.
- [6] M. F. Naufal *et al.*, “Analisis Perbandingan Algoritma Klasifikasi MLP dan CNN pada Dataset American Sign Language,” *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 3, pp. 489–495, 2021, doi: 10.29207/resti.v5i3.3009.
- [7] R. Padilla, S. L. Netto, and E. A. B. Da Silva, “A survey on performance metrics for object-detection algorithms,” in *2020 international conference on systems, signals and image processing (IWSSIP)*, 2020, pp. 237–242.
- [8] A. Amwin, “Deteksi Dan Klasifikasi Kendaraan Berbasis Algoritma You Only Look Once (YOLO),” 2021.
- [9] A. Rohim, Y. A. Sari, and T. Tibyani, “Convolution neural network (cnn) untuk pengklasifikasian citra makanan tradisional,” *JPTIIK (Jurnal Pengemb. Teknol. Inf. dan Ilmu Komputer)*, vol. 3, no. 7, pp. 7037–7042, 2019.
- [10] J. Yao, J. Qi, J. Zhang, H. Shao, J. Yang, and X. Li, “A real-time detection algorithm for Kiwifruit defects based on YOLOv5,” *Electronics*, vol. 10, no. 14, p. 1711, 2021.
- [11] D. Thuan, “Evolution of Yolo algorithm and Yolov5: The State-of-the-Art object detection algorithm,” 2021.
- [12] J. Ieamsaard, S. N. Charoensook, and S. Yammen, “Deep learning-based face mask detection using yolov5,” in *2021 9th International Electrical Engineering Congress (iEECON)*, 2021, pp. 428–431.