

Pemanfaatan Google Colab Untuk Aplikasi Pendeteksian Masker Wajah Menggunakan Algoritma Deep Learning YOLOv7

Rangga Gelar Guntara

¹Bisnis Digital, Universitas Pendidikan Indonesia
email: ranggagelar@upi.edu

Submitted:06-02-2023, Reviewed: 09-02-2023, Accepted 11-02-2023
<https://doi.org/10.47233/jteksis.v5i1.750>

Abstract

From a comprehensive review of facial mask detection techniques, there are several algorithms based on deep learning, namely You Only Look Once (YOLO), Single Shot Detector (SSD), RetinaFace, and (Faster Recurrent Convolutional Neural Network) Faster R-CNN. Previous studies focused on the detection accuracy of face masks using a two-stage detection model (ie, Faster R-CNN), while single-stage detectors (ie, YOLO) achieved fast inference times but lower accuracy. The training results in this study show that the Precision value is consistent at 0.4 – 0.8. While the maximum recall value is 0.6. Future research will focus on using YOLOv7 for other object detection.

Keywords: facial mask, deep learning, YOLOv7

Abstrak

Deteksi masker wajah berdasarkan metode deteksi objek Deep Learning memainkan peran penting dalam perang melawan COVID-19, yang mencapai hasil yang baik dengan akurasi yang tinggi. Dari tinjauan komprehensif teknik deteksi masker wajah [2], ada beberapa algoritma berbasis deep learning, yakni You Only Look Once (YOLO) [3], Single Shot Detector (SSD) [4], RetinaFace [5], and (Faster Recurrent Convolutional Neural Network) Faster R-CNN [6]. Studi sebelumnya berfokus pada akurasi deteksi masker wajah menggunakan model deteksi dua tahap (yaitu, Faster R-CNN), sementara detektor satu tahap (yaitu, YOLO) tercapai waktu inferensi yang cepat tetapi akurasi lebih rendah. Hasil training pada penelitian ini menunjukkan nilai Precision konsisten berada pada angka 0,4 – 0,8. Sedangkan nilai Recall maksimum pada angka 0,6. Untuk penelitian kedepannya akan berfokus pada pemanfaatan YOLOv7 untuk deteksi objek lainnya.

Keywords: masker wajah, deep learning, YOLOv7

This work is licensed under Creative Commons Attribution License 4.0 CC-BY International license



PENDAHULUAN

Menurut Organisasi Kesehatan Dunia (WHO), wabah COVID-19 menyumbang lebih dari 278 juta infeksi dan kurang lebih 5,4 juta kematian secara global pada akhir Desember 2021. Lebih dari dua tahun sejak awal pandemi, meskipun vaksinasi telah meningkatkan sistem kekebalan tubuh kita untuk melawan COVID-19, COVID belum sepenuhnya berakhir dan masih berdampak negatif pada kehidupan kita. Untuk mencegah penyebaran virus, salah satu metode pencegahan yang paling efektif adalah menggunakan masker wajah [1]. Saat ini, memakai masker wajah merupakan hal wajib yang harus dilakukan saat berada di tempat umum seperti rumah sakit dan sekolah. Namun, seringkali sebagian besar orang mengabaikan atau keliru dalam memakainya, sehingga mengakibatkan meningkatnya kasus infeksi dan menempatkan lebih banyak beban kasus pada sistem perawatan kesehatan publik.

Deteksi masker wajah berdasarkan metode deteksi objek *Deep Learning* memainkan peran penting dalam perang melawan COVID-19, yang mencapai hasil yang baik dengan akurasi yang tinggi. Dari tinjauan komprehensif teknik deteksi masker wajah [2], ada beberapa algoritma berbasis

deep learning, yakni You Only Look Once (YOLO) [3], Single Shot Detector (SSD) [4], RetinaFace [5], and (Faster Recurrent Convolutional Neural Network) Faster R-CNN [6]. Studi sebelumnya berfokus pada akurasi deteksi masker wajah menggunakan model deteksi dua tahap (yaitu, Faster R-CNN), sementara detektor satu tahap (yaitu, YOLO) tercapai waktu inferensi yang cepat tetapi akurasi lebih rendah.

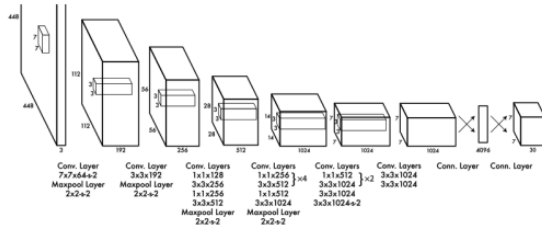
Pada penelitian ini akan menggunakan YOLOv7 yang merupakan versi paling terbaru dari Algoritma *Deep Learning* YOLO [7]. YOLOv7 diklaim memiliki tingkat akurasi yang lebih tinggi dibandingkan versi lainnya [8].

METODE PENELITIAN

Fungsi aktivasi membawa nonlinear ke model, dan sifat-sifatnya secara langsung mempengaruhi kinerja *neural*. Fungsi aktivasi asli YOLOv7 adalah SiLU, yang merupakan fungsi Sigmoid berbobot kombinasi linier, yang unggul dalam model yang dalam [9].

Penelitian ini berfokus pada algoritma deteksi masker wajah. Proses pendeteksian objek pada penelitian ini menggunakan YOLOv7. YOLO dikenalkan karena sifatnya yang open source

dikembangkan oleh University of Washington untuk memprediksi bounding boxes dan klasifikasinya dari suatu citra. Metode YOLO dikembangkan karena terinspirasi dari model GoogleNet untuk klasifikasi citra. Metode YOLO memiliki 24 layer konvolusi yang diikuti dengan 2 layer yang terkoneksi. Arsitektur dari metode YOLO dapat dilihat pada Gambar. 1.



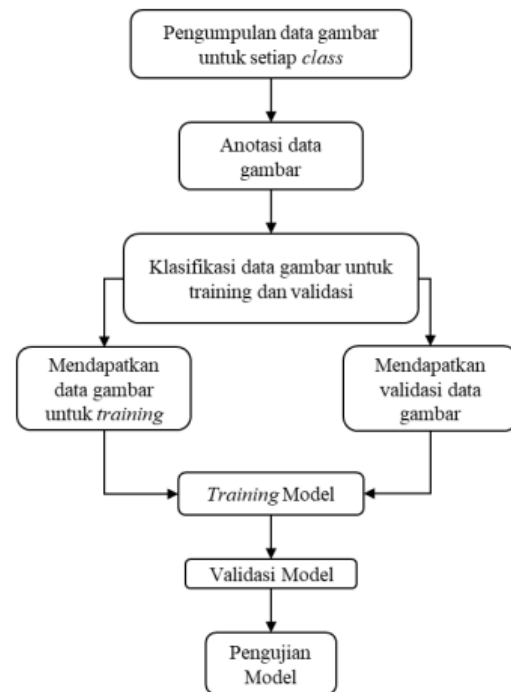
Gambar 1 Arsitektur Metode YOLO

Algoritma YOLO diklaim dapat beroperasi pada 45 fps (frame per second) pada sebuah kartu grafis Titan X. Dengan demikian, metode YOLO sangat baik untuk diterapkan pada kasus yang bersifat realtime seperti pada video. Seiring kebutuhan akan algoritma deteksi objek yang lebih baik, YOLO dikembangkan menjadi YOLO v2 dan yang terbaru adalah YOLO v3. Perbedaan dari setiap versi YOLO terletak dari peningkatan akurasi dan kecepatan pendeteksian [10]. Metode Deep Learning memerlukan komputasi yang sangat tinggi. Itu berarti membutuhkan perangkat keras dengan performa yang tinggi pula. Untuk mengatasi hal itu, pada penelitian ini akan menggunakan Google Colab.

Google Colab adalah sebuah IDE untuk pemrograman Python dimana pemrosesan akan dilakukan oleh server Google yang memiliki perangkat keras dengan performa yang tinggi [11]. Dari sisi perangkat lunak, Google Colab telah menyediakan hampir sebagian besar pustaka (library) yang dibutuhkan. Dalam penelitian ini pustaka yang dibutuhkan adalah Keras, TensorFlow, NumPy, Pandas, dan pendukung lainnya, misalnya untuk pembuatan grafik lewat Matplotlib. Bahkan seluruh versi disediakan, misalnya TensorFlow versi 1.x maupun 2.x tersedia, begitu juga versi Python yang mulai dari versi 2.x hingga 3.x. Dari sisi perangkat keras, Google Colab menyediakan layanan berupa media penyimpan yang terintegrasi dengan Google Drive, prosesor yang berupa CPU, GPU, dan TPU, serta RAM. Dengan jaminan kemampuan servernya yang stabil hampir keseluruhan pemrosesan tidak menemukan kendala dengan Google Colab selama koneksi jaringan internet lancar [11].

Penelitian ini dilakukan dalam beberapa tahapan pada proses pelaksanaannya. Tahapan

penelitian ditetapkan mengacu pada proses Deep Learning berbantuan algoritma You Only Look Once (YOLO). Berikut ini adalah tahapan penelitian meliputi pengumpulan dataset, anotasi / pelabelan, klasifikasi data untuk pelatihan dan validasi, pelatihan dan validasi model, serta pengujian model yang dapat dilihat pada Gambar. 2.

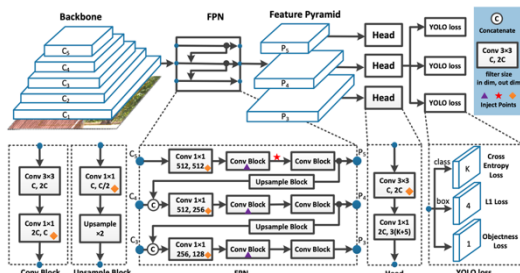


Gambar 2 Tahapan Metode Penelitian

HASIL DAN PEMBAHASAN

1. Arsitektur YOLOv7

Model YOLO (You Only Look Once) v7 merupakan model yang terbaru dalam keluarga YOLO. Model YOLO adalah detektor objek satu tahap. Dalam model YOLO, bingkai gambar ditampilkan melalui tulang punggung. Fitur-fitur ini digabungkan dan dicampur di leher, dan kemudian diteruskan ke kepala jaringan. YOLO memprediksi lokasi dan kelas objek di mana kotak pembatas harus digambar. YOLO melakukan post-processing melalui non-maximum supression (NMS) untuk sampai pada prediksi akhirnya.



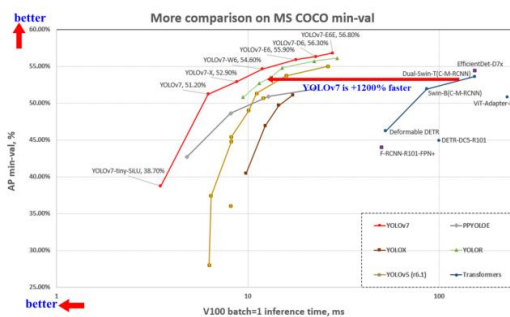
Gambar 3 Arsitektur jaringan YOLO seperti yang digambarkan dalam PP-YOLO

Model YOLO v7 ditulis oleh WongKinYiu dan Alexey Bochkovskiy (AlexeyAB).

AlexeyAB mengambil obor YOLO dari penulis aslinya, Joseph Redmon, ketika Redmon keluar dari industri Computer Vision (CV) karena masalah etika. AlexeyAB mempertahankan fork YOLOv3 untuk sementara waktu sebelum merilis YOLOv4, sebuah peningkatan dari model sebelumnya.

WongKinYiu memasuki tahap penelitian CV dengan jaringan Cross Stage Partial, yang memungkinkan YOLOv4 dan YOLOv5 membangun tulang punggung yang lebih efisien. Dari sana, WongKinYiu melaju ke depan memberikan kontribusi besar bagi keluarga penelitian YOLO dengan Scaled-YOLOv4. Model ini memperkenalkan penskalaan CSP YOLOv4 yang efisien untuk mencapai peta cangkih COCO.

Para penulis YOLOv7 berusaha untuk menetapkan kecanggihan dalam deteksi objek dengan membuat arsitektur jaringan yang akan memprediksi kotak pembatas lebih akurat daripada rekan-rekannya pada kecepatan inferensi yang sama.

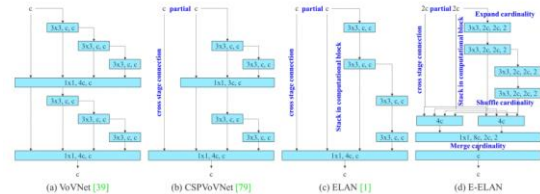


Gambar 4 YOLOv7 mengevaluasi di kiri atas - lebih cepat dan lebih akurat daripada jaringan sejenis

2. Ektensi Efisiensi Lapisan Agregasi

Efisiensi lapisan konvolusional jaringan YOLO di backbone sangat penting untuk kecepatan inferensi yang efisien. WongKinYiu memulai jalur efisiensi lapisan maksimal dengan Cross Stage Partial Networks. Di YOLOv7, penulis membangun penelitian yang telah terjadi pada topik

ini, dengan mengingat jumlah memori yang diperlukan untuk menyimpan lapisan dalam memori bersama dengan jarak yang dibutuhkan gradien untuk menyebar balik melalui lapisan. Semakin pendek gradiennya, semakin kuat jaringan mereka dapat belajar. Agregasi lapisan terakhir yang mereka pilih adalah E-ELAN, versi perluasan dari blok komputasi ELAN.

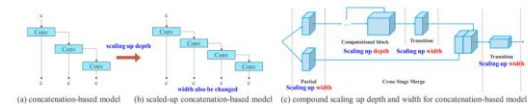


Gambar 5 Evolusi strategi agregasi layer di YOLOv7

3. Teknik Penskalaan Model

Model deteksi objek biasanya dirilis dalam serangkaian model, dengan ukuran naik dan turun, karena aplikasi yang berbeda memerlukan tingkat akurasi dan kecepatan inferensi yang berbeda.

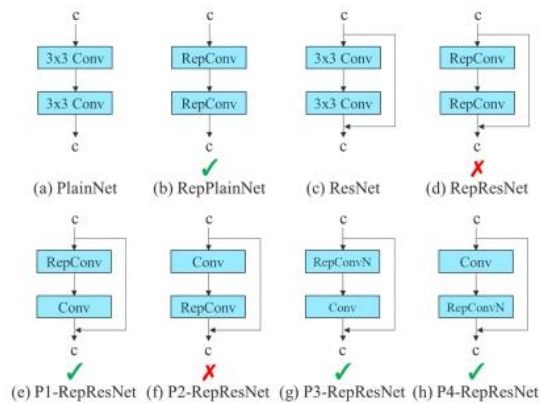
Biasanya, model deteksi objek mempertimbangkan kedalaman jaringan, lebar jaringan, dan resolusi yang digunakan untuk melatih jaringan. Di YOLOv7, penulis menskalakan kedalaman dan lebar jaringan secara bersamaan sambil menggabungkan lapisan. Studi ablasi menunjukkan bahwa teknik ini menjaga arsitektur model tetap optimal sambil menskalakan untuk ukuran yang berbeda.



Gambar 6 Penskalaan majemuk dalam ukuran model YOLOv7

4. Perencanaan parameterisasi ulang

Teknik parameterisasi ulang melibatkan rata-rata satu set bobot model untuk membuat model yang lebih kuat terhadap pola umum yang coba dimodelkan. Dalam penelitian, baru-baru ini ada fokus pada parameterisasi ulang tingkat modul di mana bagian dari jaringan memiliki strategi parameterisasi ulang mereka sendiri. Penulis YOLOv7 menggunakan jalur propagasi aliran gradien untuk melihat modul mana dalam jaringan yang harus menggunakan strategi parameterisasi ulang dan mana yang tidak.

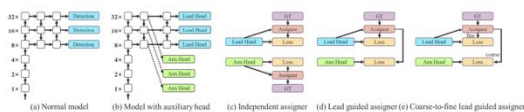


Gambar 7 Parameterisasi ulang temuan go/no-go di YOLOv7

5. Auxiliary Head dari Kasar ke Halus

Kepala jaringan YOLO membuat prediksi akhir untuk jaringan, tetapi karena berada jauh di hilir dalam jaringan, akan menguntungkan untuk menambahkan kepala tambahan ke jaringan yang terletak di suatu tempat di tengah. Saat Anda berlatih, Anda mengawasi kepala pendeteksi ini dan juga kepala yang akan membuat prediksi.

Kepala tambahan tidak berlatih seefisien kepala terakhir karena ada lebih sedikit jaringan antara itu dan prediksi - jadi penulis YOLOv7 bereksperimen dengan berbagai tingkat pengawasan untuk kepala ini, menetapkan definisi kasar ke halus di mana pengawasan diteruskan kembali dari kepala memimpin di granularities berbeda.



Gambar 8 Pengawasan kepala bantu kasar hingga halus di jaringan YOLOv7

6. Download Dataset

Di bagian ini, penulis mendownload dataset dari Kaggle.

Penulis mengubah baris kode di bawah ini menjadi komentar. Penulis menggunakan kode ini untuk mengunduh data dari Kaggle. Penulis mengunduh dataset di google drive saya. Cukup mendownloadnya sekali saja, kita tidak perlu mendownloadnya lebih dari satu kali.

```
"""
!pip install -q kaggle
from google.colab import drive
drive.mount('/content/drive')

! mkdir ~/.kaggle
!cp /content/drive/MyDrive/kaggle.json
~/.kaggle/kaggle.json
! chmod 600 ~/.kaggle/kaggle.json
```

```
%cd /content/drive/MyDrive/Dataset
! kaggle datasets download andrewmvd/face-
mask-detection
"""
!unzip
"/content/drive/MyDrive/Dataset/face-mask-
detection.zip" -d "/content/face-mask-
detection/"
Pada bagian ini akan dilakukan proses review data
di dalam dataset.
```

```
# I import libraries that I will use.
import numpy as np # linear algebra
import pandas as pd # data processing, CSV
file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt
# For preparing
import xml.etree.cElementTree as ET
import glob
import os
import json
import random
import shutil
from PIL import Image, ImageOps
Berikutnya proses pelabelan dengan format XML.
```

```
with open('/content/face-mask-
detection/annotations/maksssksksss0.xml')
as f:
    contents = f.read()
    print(contents)
Tampilkan beberapa citra latihan pada dataset.
```

```
Image.open("/content/face-mask-
detection/images/maksssksksss0.png")
```





Gambar 9 Citra latihan dalam dataset

Ubah citra latihan menjadi citra grayscale dengan kode berikut.

```
# creating a image1 object
im1 = Image.open(r"/content/face-mask-
detection/images/maksssskskss100.png")
# applying grayscale method
im2 = ImageOps.grayscale(im1)
im2
Kode berikut untuk menentukan kelas dataset.
with open('/content/classes.txt') as f:
    contents = f.read()
    print(contents)
```

7. Download YOLOv7

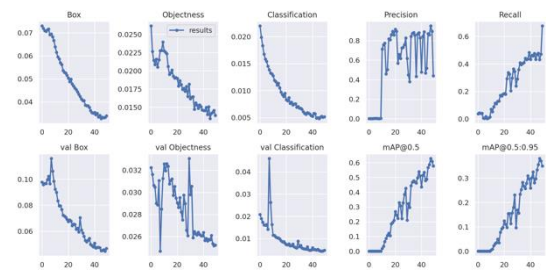
Berikutnya download model YOLOv7 ke Google Colab dengan kode berikut.

```
!# Download YOLOv7 code
%cd /content/
!git clone
https://github.com/WongKinYiu/yolo
v7
%cd yolo7
!ls
```

8. Training YOLOv7

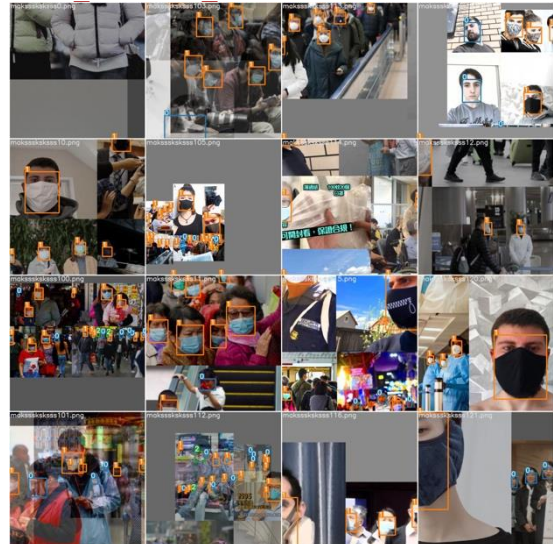
Langkah selanjutnya, tinggal melakukan proses training dataset dengan menggunakan YOLOv7.

```
# Train
!python /content/yolo7/train.py --workers
8 --device 0 --batch-size 16 --epochs 50 -
-data /content/data/data.yaml --cfg
/content/yolo7/cfg/training/yolo7.yaml -
-weights '' --name yolo7_1 --hyp
/content/yolo7/data/hyp.scratch.p5.yaml
```



Gambar 10 Visualisasi hasil training

```
Image.open("/content/runs/train/yolo
v7_1/train_batch0.jpg")
```



Gambar 11 Citra hasil pengujian

Kode berikut untuk menguji dengan citra dari luar dataset yang terdiri dari kelas with mask dan without mask.

```
Image.open("/content/runs/detect/exp3/ima
ge0.jpg")
```



Gambar 12 Citra uji

SIMPULAN

Pemanfaatan model YOLOv7 untuk aplikasi pendeteksian masker wajah manusia dapat dilakukan dengan sukses. Hasil training pada penelitian ini menunjukkan nilai Precision konsisten berada pada angka 0,4 – 0,8. Sedangkan nilai Recall maksimum pada angka 0,6. Untuk

penelitian kedepannya akan berfokus pada pemanfaatan YOLOv7 untuk deteksi objek lainnya.

DAFTAR PUSTAKA

- [1] A. Nowrin, S. Afroz, Md. S. Rahman, I. Mahmud, and Y.-Z. Cho, "Comprehensive Review on Facemask Detection Techniques in the Context of Covid-19," *IEEE Access*, vol. 9, pp. 106839–106864, 2021, doi: 10.1109/ACCESS.2021.3100070.
- [2] R. A. S. Naseri, A. Kurnaz, and H. M. Farhan, "Optimized face detector-based intelligent face mask detection model in IoT using deep learning approach," *Appl Soft Comput*, vol. 134, p. 109933, Feb. 2023, doi: 10.1016/j.asoc.2022.109933.
- [3] Althaf Adhari Rachman and Ivan Maurits, "SISTEM DETEKSI PEMAKAIAN MASKER PADA WAJAH SECARA REAL TIME MENGGUNAKAN FRAMEWORK TENSORFLOW DAN LIBRARY OPENCV," *Jurnal Ilmiah Teknik*, vol. 2, no. 1, pp. 49–59, Jan. 2023, doi: 10.56127/juit.v2i1.496.
- [4] W. Zhu, H. Zhang, J. Eastwood, X. Qi, J. Jia, and Y. Cao, "Concrete crack detection using lightweight attention feature fusion single shot multibox detector," *Knowl Based Syst*, vol. 261, p. 110216, Feb. 2023, doi: 10.1016/j.knosys.2022.110216.
- [5] X. Fan and M. Jiang, "RetinaFaceMask: A Single Stage Face Mask Detector for Assisting Control of the COVID-19 Pandemic," May 2020.
- [6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Trans Pattern Anal Mach Intell*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017, doi: 10.1109/TPAMI.2016.2577031.
- [7] N. D. T. Yung, W. K. Wong, F. H. Juwono, and Z. A. Sim, "Safety Helmet Detection Using Deep Learning: Implementation and Comparative Study Using YOLOv5, YOLOv6, and YOLOv7," in *2022 International Conference on Green Energy, Computing and Sustainable Technology (GECOST)*, Oct. 2022, pp. 164–170. doi: 10.1109/GECOST55694.2022.10010490.
- [8] S. Liu, Y. Wang, Q. Yu, H. Liu, and Z. Peng, "CEAM-YOLOv7: Improved YOLOv7 Based on Channel Expansion and Attention Mechanism for Driver Distraction Behavior Detection," *IEEE Access*, vol. 10, pp. 129116–129124, 2022, doi: 10.1109/ACCESS.2022.3228331.
- [9] S. Elfving, E. Uchibe, and K. Doya, "Sigmoid-weighted linear units for neural network function approximation in reinforcement learning," *Neural Networks*, vol. 107, pp. 3–11, Nov. 2018, doi: 10.1016/j.neunet.2017.12.012.
- [10] T. Q. Vinh and N. T. N. Anh, "Real-Time Face Mask Detector Using YOLOv3 Algorithm and Haar Cascade Classifier," in *2020 International Conference on Advanced Computing and Applications (ACOMP)*, Nov. 2020, pp. 146–149. doi: 10.1109/ACOMP50827.2020.00029.
- [11] R. T. Handayanto and H. Herlawati, "Prediksi Kelas Jamak dengan Deep Learning Berbasis Graphics Processing Units," *Jurnal Kajian Ilmiah*, vol. 20, no. 1, pp. 67–76, Jan. 2020, doi: 10.31599/jki.v20i1.71.