

# Implementasi Pengolahan Citra pada Quadcopter untuk Deteksi Manusia Menggunakan Algoritma YOLO

Alya Nur Sugandi<sup>1</sup>, Budi Hartono<sup>2</sup>

<sup>1,2</sup>Jurusan Teknik Mesin, Politeknik Negeri Bandung, Bandung 40012  
E-mail : alya.nur.aer19@polban.ac.id; buhar@polban.ac.id

## ABSTRAK

Sistem pengolahan citra merupakan sistem yang telah dikembangkan hingga saat ini. Dengan menggunakan sistem pengolahan citra, sebuah gambar dapat diproses menjadi sebuah informasi yang dibutuhkan. Sistem pengolahan citra sudah banyak diterapkan pada sistem lain, salah satunya quadcopter. Beberapa manfaat implementasi sistem pengolahan citra pada quadcopter di antaranya, untuk pemetaan lahan, pemantauan udara, bahkan evakuasi korban bencana. Pada penelitian ini, dibuat sebuah sistem pengolahan citra menggunakan algoritma *You Only Look Once* (YOLO) yang merupakan salah satu bagian dari *Convolutional Neural Network* (CNN). Sistem ini dapat diterapkan pada quadcopter, khususnya untuk pendeteksian objek manusia. Hasilnya, sistem pengolahan citra memiliki akurasi model sebesar 86.8% dan dapat mendeteksi objek manusia dengan baik.

## Kata Kunci

*pengolahan citra, pendeteksian, You Only Look Once (YOLO), Convolutional Neural Network (CNN) quadcopter.*

## 1. PENDAHULUAN

Teknologi telah berkembang sangat pesat di Indonesia khususnya pada era teknologi industri 4.0 yang telah memberikan banyak manfaat dalam kemajuan di berbagai aspek, salah satunya adalah penggunaan mesin untuk membantu manusia agar semakin mudah dalam melaksanakan pekerjaan dan kegiatannya sehari-hari.

Penggunaan mesin dapat meningkatkan kinerja manusia dan mengurangi resiko terjadinya *human error*, karena mesin dapat menghasilkan pekerjaan yang lebih konsisten. Salah satu penggunaan mesin yang sampai saat ini masih dikembangkan yaitu pengimplementasian pengolahan citra pada suatu mesin untuk membantu manusia mengumpulkan informasi dari beberapa gambar.

Pengolahan citra adalah teknik mengolah gambar, baik berupa gambar bergerak atau diam menjadi suatu informasi yang bermanfaat [1]. Selain mencari informasi, pengolahan citra dapat digunakan dalam beberapa penerapan seperti *image recognition* dan *object detection*.

*Image recognition* merupakan suatu konsep untuk membuat mesin dapat mengidentifikasi objek dari sebuah gambar, sedangkan *object detection* adalah suatu konsep untuk membuat mesin dapat mengenali suatu objek dan menentukan posisi dari objek tersebut di dalam sebuah gambar. *Image recognition* dan *object detection* dapat diimplementasikan pada sebuah mesin dengan menggunakan algoritma yang dapat dipahami oleh mesin. Salah satu algoritma yang ada pada pendeteksian objek ini adalah *You Only Look Once* (YOLO).

YOLO adalah sebuah algoritma yang dikembangkan untuk mendeteksi sebuah objek secara *real-time* [2]. Algoritma YOLO membutuhkan beberapa data gambar untuk dipelajari agar saat diimplementasikan pada suatu mesin, mesin dapat mendeteksi beberapa objek dan mengklasifikasikannya berdasarkan gambar yang dipelajari sebelumnya oleh algoritma YOLO.

Pendeteksian objek menggunakan mesin ini telah membantu beberapa pekerjaan manusia saat ini, contohnya adalah CCTV pendeteksi penggunaan masker pada manusia, CCTV

pendeteksi penggunaan helm di lalu lintas, dan lainnya. Berdasarkan contoh tersebut, algoritma pendeteksian objek dapat diimplementasikan pada beberapa mesin seperti CCTV, quadcopter, dan lain sebagainya.

Pada penelitian ini, dilakukan implementasi pengolahan citra pada quadcopter untuk mendeteksi objek menggunakan algoritma YOLO. Pengimplementasian ini diharapkan dapat membantu beberapa pekerjaan manusia, seperti membantu proses evakuasi bencana, pemetaan lahan, pemantauan dari udara, dan lain-lain.

## 2. DASAR TEORI

### 2.1. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) merupakan salah satu *deep learning* untuk *computer vision* [2], dimana *computer vision* adalah kemampuan komputer untuk melihat seperti manusia. Banyak hal yang dapat dilakukan oleh CNN, di antaranya untuk analisis citra. Selain itu, CNN juga dapat digunakan untuk pendeteksian dan pengenalan objek pada sebuah citra [3]. Pada prinsipnya, CNN bekerja dengan memanfaatkan proses konvolusi dengan cara menggerakkan filter berukuran tertentu ke sebuah gambar, lalu komputer mendapatkan informasi representatif baru dari hasil perkalian bagian gambar tersebut dengan filter yang digunakan [4]. Dengan kata lain, CNN bekerja bagaikan saraf manusia yang harus terhubung satu sama lainnya agar dapat bekerja dengan baik.

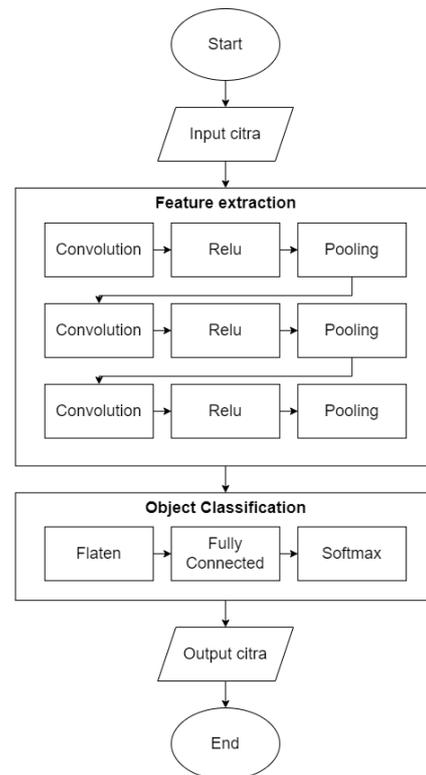
Terdapat dua bagian yang menyusun arsitektur CNN, yaitu *Feature Extraction Layer* dan *Fully Connected Layer* (MLP).

- 1) *Feature Extraction Layer*. Pada bagian ini, terjadi proses “*encoding*” yang dilakukan untuk mengubah gambar menjadi fitur berupa angka-angka yang merepresentasikan gambar tersebut (*feature extraction*) [4]. Pada *feature extraction* terdapat dua jenis layer, yaitu *Convolutional Layer* dan *Pooling Layer*.
- 2) *Fully Connected Layer* (MLP). Pada bagian ini, semua *neuron* aktivitas dari lapisan sebelumnya terhubung semua dengan *neuron* di lapisan aktivitas selanjutnya seperti jaringan saraf tiruan biasa [4]. MLP bertujuan untuk mengolah data sehingga bisa diklasifikasikan.

### 2.2. You Only Look Once (YOLO)

YOLO merupakan algoritma yang digunakan untuk mendeteksi objek secara *real-time*. Arsitektur YOLO mirip dengan CNN. Proses YOLO dalam mendeteksi objek, ditunjukkan pada Gambar 1.

Prinsip kerja YOLO adalah YOLO membagi gambar input menjadi beberapa kotak dan memprediksi setiap *bounding box* serta probabilitas untuk setiap kotak. Jika sebuah objek pusatnya jatuh di salah satu sel kotak, maka kotak tersebut harus mendeteksi objek tersebut. Setiap sel kotak memprediksi *bounding box* dan *confidence score* dari tiap *bounding box* tersebut. *Confidence score* merupakan besarnya persentase keyakinan dan akurasi model bahwa terdapat objek pada kotak tersebut. Terdapat lima prediksi pada *bounding box*, yaitu  $x, y, w, h$ , dan *confidence*. Koordinat  $(x,y)$  mewakili pusat dari kotak relatif ke batas sel grid. Koordinat  $(w,h)$  atau lebar dan tinggi mewakili pusat dari kotak relatif ke gambar. Serta *confidence* mewakili *Intersection over Union* antara kotak prediksi dan kotak *ground truth* [2].



Gambar 1 Diagram alir pendeteksian objek menggunakan YOLO.

### 2.3. Quadcopter

Quadcopter merupakan kendaraan udara tanpa awak (*unmanned aerial vehicle*) yang memiliki empat rotor yang diletakkan dalam formasi persegi dengan jarak yang sama dari pusat ke massa quadcopter [5].

## 3. METODOLOGI

Pada bagian ini dijelaskan tahap-tahap pelaksanaan pendeteksian objek beserta pengujian dari hasil pendeteksian objek yang telah dilakukan.

### 3.1. Dataset Gambar

Tahap ini dilakukan dengan mengumpulkan gambar yang diperoleh dari dokumentasi pribadi dan dataset dari internet. Dataset gambar dibagi menjadi *train set* dan *validation set*. Setelah dataset dibuat, selanjutnya dilakukan *labeling* untuk mendapatkan hasil *training*.

### 3.2. Labeling Dataset

*Labeling* merupakan proses pemberian informasi pada dataset gambar dengan memberikan *bounding box* atau kotak pembatas. *Labeling* dilakukan agar diperoleh koordinat *ground-truth bounding box* yang akan dibandingkan dengan *predicted bounding box* sehingga diperoleh nilai *Intersection over Union* (IoU). Pada prosesnya, *labeling* dilakukan melalui *website Make Sense*, kemudian hasilnya di ekspor ke dalam YOLO dengan format .txt pada setiap *file-nya*.

### 3.3. Training Dataset

Dataset yang telah melalui proses *labeling* kemudian di-*training* atau dilatih untuk membuat suatu prediksi sehingga dapat digunakan untuk mendeteksi objek pada suatu citra. *Training* yang dilakukan menggunakan algoritma YOLO yang metodenya berbasis CNN. Pada prosesnya, data *training* diambil menggunakan perangkat Google Colaboratory.

### 3.4. Pengujian Kinerja Pendeteksian Objek

Pengujian ini dilakukan untuk mengetahui kelayakan model hasil *training* dataset yang telah dibuat serta tingkat akurasi pada YOLOv5. Salah satu cara untuk menguji suatu model deteksi objek adalah dengan menggunakan metode *Intersection over Union*

(IoU) dengan memanfaatkan *bounding box* yang didapatkan dari hasil pendeteksian objek.

Proses ini diawali dengan mencari nilai *bounding box ground truth* dan nilai *bounding box* hasil prediksi. Kedua *bounding box* ini kemudian dibandingkan untuk mengetahui nilai IoU. Nilai IoU diperoleh melalui Persamaan 1.

$$IoU = \frac{area(BB_{prediksi} \cap BB_{groundTruth})}{area(BB_{prediksi} \cup BB_{groundTruth})} \quad (1)$$

Setelah nilai IoU diperoleh, maka dapat dihitung nilai *mean Average Precision* (mAP) yang merupakan rata-rata nilai IoU. Pada penelitian ini, dibutuhkan nilai *threshold* lebih dari 0.5 untuk mengakui keberhasilan dari nilai mAP. Nilai *threshold* ini juga digunakan untuk mencari nilai *precision* dan *recall* dari model pendeteksian objek. *Precision* merupakan hasil perbandingan antara kebenaran prediksi dan keseluruhan hasil prediksi yang positif, sedangkan *recall* merupakan perbandingan hasil kebenaran prediksi dengan keseluruhan prediksi. Berikut merupakan ketentuan dari nilai *threshold* 0.5:

1. Jika  $IoU \geq 0.5$  maka objek diklasifikasikan sebagai *True Positive* (TP).
2. Jika  $IoU < 0.5$  maka objek diklasifikasikan sebagai *False Positive* (FP).
3. Jika *ground-truth* menampilkan objek dan model gagal mendeteksi objek maka diklasifikasikan sebagai *False Negative* (FN).
4. Jika *ground-truth* tidak menampilkan objek dan model tidak mendeteksi objek maka diklasifikasikan sebagai *True Negative* (TN).

Berdasarkan ketentuan tersebut, dapat dihitung nilai *precision* dan *recall* menggunakan rumus pada Persamaan 2 dan 3.

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

Nilai *precision* dan *recall* kemudian divisualisasikan menjadi sebuah kurva, dimana jika area dibawah kurva dikalkulasi, maka diperoleh nilai *Average Precision* (AP). Untuk nilai mAP sendiri didapatkan dari rata-rata nilai AP seluruh kelas yang terdeteksi.

Selain itu, nilai *precision* dan *recall* dapat digunakan untuk mencari nilai *F1-score* menggunakan rumus pada Persamaan 4.

$$F1\text{-score} = 2 \times \left( \frac{R \times P}{R + P} \right) \quad (4)$$

Nilai *F1-score* ini digunakan untuk mengindikasikan bahwa model pendeteksian objek memiliki nilai *precision* dan *recall* yang baik.

### 3.5. Pendeteksian Objek

Pendeteksian objek dilakukan berdasarkan model hasil *training* dataset yang telah dibuat. Proses ini diawali dengan mengolah input berupa citra yang memiliki ukuran berkelipatan 32. Semakin besar ukuran gambar maka semakin akurat pula hasil pendeteksian, akan tetapi proses komputasi akan membutuhkan waktu yang lebih lama, begitu pula sebaliknya. Selanjutnya, dilakukan *feature extraction* dimana citra direpresentasikan menjadi kumpulan angka-angka agar bisa diproses oleh komputer. Kemudian, dilakukan klasifikasi objek berdasarkan angka-angka tersebut. Klasifikasi objek ini menghasilkan *output* berupa prediksi yang terdiri dari koordinat *bounding box*, *confidence score*, dan *class name* (nama objek yang diprediksi).

## 4. HASIL DAN PENGUJIAN

Pada bagian ini dijelaskan hasil pengujian dari sistem pengolahan citra yang sudah didesain.

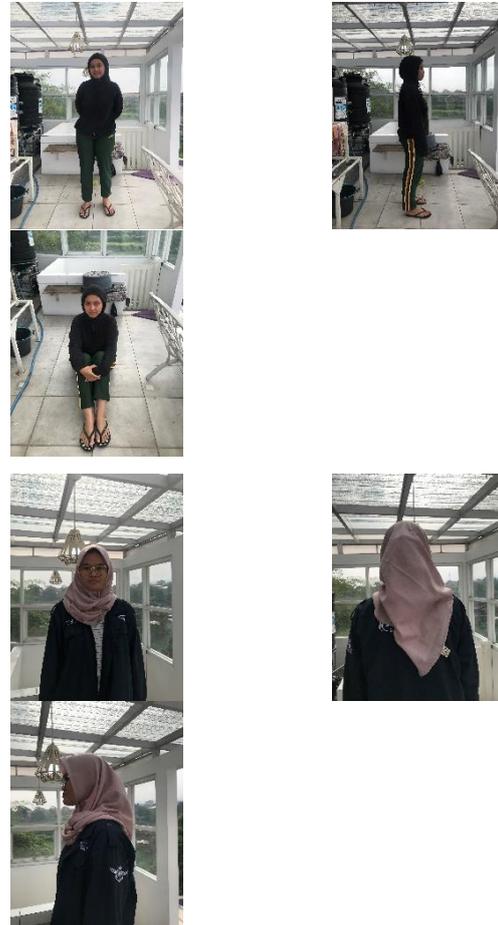
### 4.1. Pembuatan Sistem Pendeteksian Manusia

Sistem pendeteksian manusia dibuat menggunakan algoritma YOLO. Untuk membangun sistem ini, langkah pertama yang dilakukan adalah mengumpulkan dataset gambar yang kemudian dibagi menjadi *train set* dan *validation set*. Lalu setelah dataset gambar terkumpul, dilakukan proses *labeling* dalam satu kelas yaitu objek manusia. Setelah dilakukan *labeling*, selanjutnya dilakukan *training* dataset untuk melatih data agar model dapat membuat prediksi untuk mendeteksi objek.

#### 4.1.1. Pembuatan Dataset

Tujuan pembuatan dataset adalah untuk menyiapkan data-data yang akan ditraining. Data yang digunakan adalah gambar manusia yang didokumentasikan secara pribadi dan dataset yang diunduh dari internet. Total

gambar yang disiapkan adalah sebanyak 1000 gambar yang dibagi menjadi *train set* sebanyak 700 gambar dan *validation set* sebanyak 300 gambar. Gambar 2 menunjukkan beberapa gambar yang akan di-*train*



Gambar 2 Beberapa gambar yang akan di-*train*.

#### 4.1.2. Training Data

*Training* data bertujuan untuk menghasilkan sebuah bobot yang akan digunakan dalam proses pendeteksian manusia. Pada prosesnya, *training* data dilakukan menggunakan algoritma YOLOv5 dan perangkat *Google Colaboratory*. Proses *training* dilakukan sebanyak tiga kali dengan menggunakan *img size* (resolusi gambar yang di-*train*) 480, *batch* (jumlah *mini batch size*) 25, dan *epoch* (banyaknya iterasi atau pelatihan yang dilakukan) 100. Setelah tiga kali proses *training*, didapatkan nilai *mAP* terbaik sebesar 86.8%.

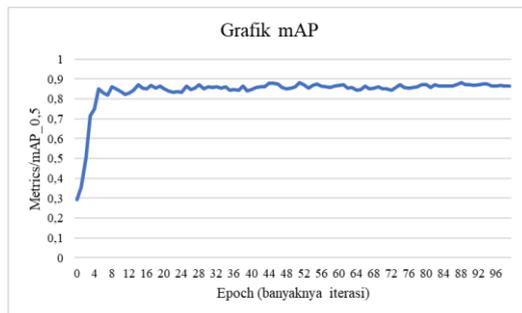
#### 4.1.3. Hasil Pengujian Kinerja YOLOv5

Tujuan dilakukan pengujian kinerja adalah untuk mengetahui tingkat akurasi model pada YOLOv5. Pada prosesnya, dibutuhkan dataset *test* yang belum pernah disertakan pada proses *training*. Selain itu, dibutuhkan juga *validation set* yang sudah diberi label. Selanjutnya, *validation set* diolah menjadi *ground-truth box* yang dibandingkan dengan *predicted box* kemudian menghasilkan *confusion matrix*, lalu dikalkulasikan untuk mendapatkan nilai *Precision*, *Recall*, *Average Precision (AP)*, *F1-score*, *Intersection over Union*, dan mean *Average Precision (mAP)*.

Pengujian kinerja dilakukan menggunakan *Google Colaboratory*. Hasil dari pengujian dapat dilihat pada Tabel 1. Pada pengujian ini didapatkan nilai mAP sebesar 86.8%. Grafik mAP terhadap *epoch* ditunjukkan oleh Gambar 3.

Tabel 1 Pengujian kinerja YOLOv5

Load Model	YOLOv5
Gambar yang di-train	700
Waktu proses train	831.6 s
Precision	0.862
Recall	0.815
F1-score	0.838
mAP	86.8%



Gambar 3 Grafik nilai mAP terhadap *epoch*.

Gambar 3 menunjukkan bahwa semakin banyak iterasi yang dilakukan atau semakin besar *epoch* yang di-*input* saat *training* data, maka nilai mAP akan semakin mendekati 1. Hal tersebut mengindikasikan bahwa banyaknya iterasi berpengaruh pada nilai mAP yang dihasilkan dan tingkat akurasi model.

#### 4.2. Hasil Pendeteksian Manusia

Pendeteksian manusia dilakukan menggunakan webcam yang di arahkan ke

objek. Gambar 4 menunjukkan bahwa objek yang terdeteksi ditandai dengan *bounding box* yang lengkap dengan nama kelas, dengan nilai *confidence* 0.96, dimana hal tersebut menunjukkan bahwa sistem yakin sebesar 96% bahwa objek yang tertangkap kamera merupakan manusia. Serta FPS-nya menunjukkan nilai 7.



Gambar 4 Hasil pendeteksian manusia menggunakan kamera *webcam*.

#### 4.3. Pengujian Berdasarkan Jarak Kamera

Tabel 2 menunjukkan hasil pengujian model pendeteksian objek berdasarkan jarak kamera.

Tabel 2 Hasil pengujian pendeteksian manusia berdasarkan jarak kamera

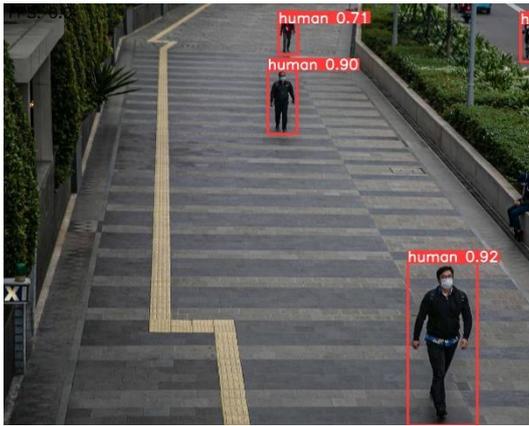
Jarak (m)	Keterangan	Confidence Score
0.5	Terdeteksi	94%
1.0	Terdeteksi	92%
1.5	Terdeteksi	88%
2.0	Terdeteksi	94%
2.5	Terdeteksi	93%
3.0	Terdeteksi	94%
3.5	Terdeteksi	93%
4.0	Terdeteksi	91%
4.5	Terdeteksi	91%

Berdasarkan hasil pengujian yang terlihat pada Tabel 2, rata-rata *confidence score* yang didapatkan dari hasil pengujian berbagai macam jarak adalah 92.2%. Dapat dilihat pada Tabel 2, bahwa semakin jauh jarak objek dari kamera, nilai *confidence score* menjadi berkurang. Namun, hal tersebut tidak berlaku pada saat objek yang dideteksi berada pada jarak 1.5 meter dari kamera. Hal tersebut bisa saja dipengaruhi oleh beberapa variabel seperti kecerahan dan ketajaman citra yang tertangkap.

#### 4.4. Pengujian Berdasarkan Jumlah Objek

Gambar 5 menunjukkan hasil pengujian berdasarkan jumlah objek yang terdeteksi.

Pada Gambar 5 terdapat lima orang yang sebenarnya, namun sistem hanya dapat mendeteksi empat orang. Hal ini dapat disebabkan karena sistem tidak dapat memprediksi citra manusia yang tidak terdeteksi. Seperti yang terlihat pada gambar, citra manusia yang tidak terdeteksi memang kurang jelas, sehingga hal tersebut dapat menjadi faktor citra tidak terdeteksi. Namun, secara keseluruhan sistem pendeteksi manusia sudah mampu untuk mendeteksi objek lebih dari satu.



Gambar 5 Hasil pendeteksian manusia berdasarkan jumlah objek.

## 5. SIMPULAN

Berdasarkan pengujian sistem yang telah dilakukan, penulis berhasil mengimplementasikan sistem pendeteksian manusia. Simpulan dari hasil pengujian yang telah dilakukan menunjukkan bahwa sistem pendeteksian manusia yang telah dibuat berhasil mendeteksi objek berupa manusia. Hal ini ditunjukkan oleh nilai mAP yang diperoleh dari hasil pengujian kinerja YOLOv5 adalah sebesar 86.8% yang berarti akurasi sistem untuk mendeteksi objek sudah baik. Waktu pemrosesan yang dibutuhkan untuk *training* adalah selama 831.6 s untuk 700 gambar. Selanjutnya, hasil pengujian sistem berdasarkan jarak dari kamera ke objek dengan rentang jarak 0.5 m sampai 4.5 m memiliki rata-rata *confidence score* sebesar 92.2%. Dapat disimpulkan bahwa akurasi pendeteksian sudah baik dan optimal sampai jarak 4.5 m. Selain itu, sistem juga dapat mendeteksi objek lebih dari satu dalam satu *frame*.

## DAFTAR PUSTAKA

[1] N. Kumar, "Digital Image Processing: Geeks for Geeks," [Online]. Available:

<https://www.geeksforgeeks.org/digital-image-processing-basics/>. [Accessed 10 Maret 2022].

- [2] Khairunnas, E. M. Yuniarno and A. Zaini, "Pembuatan Modul Deteksi Objek Manusia Menggunakan Metode YOLO untuk Mobile Robot," Jurnal Teknik ITS, vol. 10, no. 1, pp. 50-55, 2021.
- [3] C. Geraldly and C. Lubis, "Pendeteksian dan Pengenalan Jenis Mobil Menggunakan Algoritma You Only Look Once dan Convolutional Neural Network," Jurnal Ilmu Komputer dan Sistem Informasi, pp. 197-199.
- [4] Q. Lina, "Apa itu Convolutional Neural Network?: Medium," 2 Januari 2019. [Online]. Available: <https://medium.com/@16611110/apa-itu-convolutional-neural-network-836f70b193a4>. [Accessed 28 Mei 2022].
- [5] X. Zhang, Z. Zhao, Z. Wang and X. Wang, "Fault Detection and Identification Method for Quadcopter," MDPI, vol. 21, no. 2, pp. 1-16, 2021.