

Quadcopter v8: Kaji Pengolahan Citra untuk Misi Terbang Pendeteksian Keberadaan Manusia

Rikemaulani Affandi^{1,*}, Budi Hartono²

^{1,2}Jurusan Teknik Mesin, Politeknik Negeri Bandung, Bandung 40012
E-mail : ^{1,*}rikemaulani.affandi.aer20@polban.ac.id; ²buhar@polban.ac.id

ABSTRAK

Quadcopter memiliki bentuk yang kecil sehingga leluasa untuk bergerak di tempat yang sulit dan quadcopter dapat terbang secara vertikal, yang berarti tidak perlu landasan pacu untuk terbang. Kemajuan teknologi terus berkembang dengan cepat. Teknologi dalam mendeteksi suatu objek pada saat ini sangat populer mulai dari kebutuhan dalam mendeteksi objek seperti warna, wajah, sidik jari, dan sejenisnya menjadi awal pada pengembangan aplikasi citra digital yang lebih modern. Pada penelitian ini, telah dibuat quadcopter dengan sistem pengolahan citra menggunakan metode algoritma YOLO (*You Only Look Once*) yang dapat mengenali objek manusia. Quadcopter ini dapat terbang secara stabil dengan menggunakan *flight controller* Pixhawk Px4 yang telah diimplementasikan dengan sistem pendeteksi keberadaan manusia menggunakan Raspberry Pi, yang dapat terbang stabil pada ketinggian 2-6 meter.

Kata Kunci

quadcopter, pengolahan citra, *You Only Look Once* (YOLO)

1. PENDAHULUAN

Quadcopter merupakan salah satu jenis drone atau pesawat tanpa awak yang dapat terbang baik secara kendali manual maupun otomatis. Pada awalnya drone hanya digunakan oleh militer untuk digunakan sebagai mata-mata untuk mengawasi keadaan pada saat yang tidak memungkinkan untuk mengirim orang ke suatu wilayah darurat.

Namun semakin berkembangnya zaman, drone semakin berkembang dan dapat digunakan oleh masyarakat umum. Sebagian drone saat ini telah dipasang kamera sehingga drone dapat digunakan untuk memantau dan merekam dari ketinggian. Drone ini digerakkan melalui tenaga baterai yang terpasang di dalamnya.

Citra adalah representasi objek dua dimensi dari dunia visual, menyangkut berbagai macam disiplin ilmu yang mencakup seni, *human vision*, astronomi, teknik, dan sebagainya. Citra merupakan suatu kumpulan piksel-piksel atau titik-titik yang berwarna yang berbentuk dua dimensi.

Pengolahan citra sangat penting dalam perkembangan teknologi. Teknik ini bertujuan untuk memperbaiki kualitas citra agar mudah diinterpretasi oleh manusia atau mesin komputer yang dapat berupa foto maupun gambar bergerak.

Pengolahan citra dapat digunakan dalam beberapa penerapan seperti *image recognition* dan *object detection*.

Image recognition adalah proses pengenalan gambar dan video untuk mengidentifikasi dan mendeteksi suatu objek. Tujuan utamanya adalah untuk memungkinkan mesin atau komputer untuk “melihat” dan mengenali objek, pola, wajah, teks, atau fitur lain dalam gambar dengan cara yang mirip dengan manusia. Salah satu algoritma yang digunakan untuk mendeteksi objek yaitu YOLO (*You Only Look Once*).

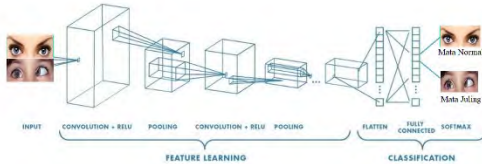
YOLO adalah algoritma yang dapat mendeteksi dan mengenali objek dalam gambar (secara real-time). Algoritma YOLO menggunakan *convolutional neural network* (CNN) untuk mendeteksi objek secara real-time [1].

Pada penelitian ini, dilakukan implementasi pengolahan citra pada quadcopter untuk mendeteksi keberadaan manusia menggunakan algoritma YOLO. Pengimplementasian ini diharapkan dapat membantu beberapa pekerjaan manusia seperti pencarian korban pada saat terjadi bencana alam.

2. DASAR TEORI

2.1 Convolutional Neural Network (CNN)

Convolutional neural network adalah algoritma *Deep Learning* tiruan yang paling sering digunakan dalam pengenalan gambar, penglihatan komputer, dan pemrosesan citra [2].



Gambar 1. Contoh proses *Convolutional Neural Network*

Arsitektur CNN terdiri dari beberapa lapisan, termasuk lapisan konvolusi, lapisan pooling, dan lapisan terhubung penuh (*fully connected layers*). Lapisan konvolusi adalah inti dari CNN, di mana filter (kernel) konvolusi diterapkan pada gambar untuk menghasilkan fitur-fitur berdasarkan pola lokal dalam gambar. Setiap filter akan menghasilkan peta fitur yang berbeda, kemudian akan digabungkan untuk membentuk representasi fitur yang lebih abstrak (perhatikan Gambar 1).

Lapisan *pooling* digunakan untuk mengurangi dimensi spasial dari fitur yang dihasilkan oleh lapisan konvolusi. Operasi *pooling* seperti *max pooling* atau *average pooling* akan mengambil nilai maksimum atau rata-rata dari area yang ditentukan dalam peta fitur, sehingga mengurangi ukuran data tetapi mempertahankan informasi yang penting. [3]

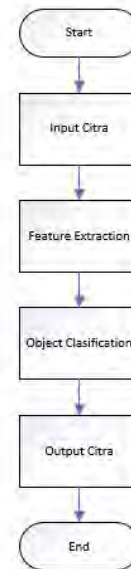
Lapisan terhubung penuh (*fully connected layers*) terletak di akhir arsitektur CNN dan bertanggung jawab untuk melakukan klasifikasi berdasarkan fitur-fitur yang ditemukan oleh lapisan konvolusi dan *pooling* sebelumnya. *Fully connected layers* menghubungkan semua neuron di lapisan sebelumnya dengan semua neuron lapisan berikutnya dan menghasilkan output klasifikasi.

2.2 You Only Look Once (YOLO)

Algoritma YOLO (*You Only Look Once*) adalah metode yang digunakan dalam pendekatan deteksi objek dalam pengenalan gambar dan penglihatan komputer [4], seperti ditunjukkan pada Gambar 2. Algoritma YOLO membagi gambar menjadi grid dan setiap grid bertanggung jawab untuk memprediksi kotak pembatas (*bounding box*) serta kelas objek yang ada di dalamnya. Proses deteksi

objek dilakukan dalam satu tahap, dimana jaringan saraf memproses seluruh gambar.

Algoritma YOLO menjadi populer karena kemampuannya dalam mendeteksi objek secara *real-time* dengan kecepatan tinggi dan kinerja yang baik dalam mengenali objek-objek dalam berbagai konteks visual.



Gambar 2. Diagram alir pendeteksian objek menggunakan YOLO

2.3 Quadcopter

Quadcopter adalah sebuah helikopter multirotor tanpa awak yang diangkat dan didorong oleh empat rotor. Quadcopter diklasifikasikan sebagai pesawat rotor, karena *lift*-nya dihasilkan oleh satu set rotor *propeller* [5].

3. Metodologi

Pada tahap ini dijelaskan bagaimana pelaksanaan pendeteksian manusia beserta pengujian dari hasil pendeteksian objek keberadaan manusia yang telah dilakukan.

3.1 Dataset Gambar

Tahap ini dilakukan dengan mengumpulkan gambar yang diperoleh dari dokumentasi pribadi. Gambar ini dibagi menjadi *train set* dan *validation set*. Setelah dilakukan pembagian folder dataset selanjutnya dilakukan *labeling* untuk dapat dilakukan *training* pada Google Colaboratory.

3.2 Labeling Dataset

Labeling dataset YOLO (*You Only Look Once*) anotasi pada setiap objek dalam sebuah dataset menggunakan format YOLO. Dalam konteks labeling dataset YOLO, setiap objek pada gambar akan diberikan label kelas yang sesuai, serta kotak pembatas (*bounding box*) yang mengelilingi objek tersebut.

3.3 Training Dataset

Setelah dataset diberikan label kemudian di-*training* untuk dapat digunakan untuk mendeteksi objek. Training ini dilakukan menggunakan algoritma YOLO yang metodenya berbasis CNN. Pada prosesnya, data *training* diambil menggunakan Google Colaboratory. Setelah training selesai, maka perlu evaluasi kinerja model menggunakan data validasi atau data uji yang terpisah. Dapat juga melihat nilai metrik seperti presisi, *recall*, atau *mean Average Precision* (mAP) untuk mengukur sejauh mana model mampu mendeteksi objek dengan akurasi yang baik. Pada saat *training* perlu juga terhubung pada GPU agar waktu proses pada saat *training* lebih cepat.

3.3 Pengujian Kinerja Pendeteksian Objek

Pengujian ini dilakukan untuk mengetahui keakuratan dalam mendeteksi pada YOLOv5. Salah satunya menggunakan metode *intersection over union* (IoU) dengan memanfaatkan *bounding box* yang didapatkan dari hasil pendeteksian objek.

Class confidence score memberi nilai kelas spesifik untuk setiap kotak, yang mengkodekan kemungkinan kelas yang muncul di kotak dan seberapa sesuai kotak yang diprediksi dengan objek. Jika tidak ada objek yang terdeteksi maka nilai *confidence* adalah nol. Untuk mendapatkan nilai *confidence score* dapat dituliskan dengan persamaan berikut:

$$\text{Confidance Score} = pr(\text{Class}_i) \times IoU_{pred}^{truth} \quad (1)$$

Intersection over Union (IoU) dapat dihitung dengan membandingkan *ground-truth bounding box* dan *predicted bounding box* yang dapat ditulis dengan persamaan berikut:

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}} \quad (2)$$

Setelah didapatkan nilai *IuO*, maka diperoleh pula nilai rata-rata *IuO* atau dapat disebut dengan *Mean average presision* (Map). Pada penelitian ini *Map@IoU* membutuhkan nilai *threshold* yang melebihi 0.5 untuk diakui keberhasilannya. Jika nilai tersebut kurang dari 0.5 maka hasil dianggap kurang akurat. Untuk ketetapan *threshold* 0.5 maka diketahui jika:

1. Jika $IoU \geq 5$ maka klasifikasikan objek sebagai True Positive (TP)
2. Jika $IoU \leq 5$ maka klasifikasikan objek sebagai False Positive (FP)
3. Jika ground truth menampilkan gambar dan model yang gagal mendeteksi objek maka klasifikasikan sebagai False Negative (FN)
4. Setiap bagian citra yang tidak dideteksi maka klasifikasikan sebagai True Negative (TN).

Nilai di atas dapat digunakan untuk menghitung *precision* dan *recall* menggunakan rumus:

$$\text{Precision} = \frac{TP}{TP+FP} \quad (3)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad \dots(4)$$

Dari hasil penghitungan *IoU* didapatkan nilai *True Positive*, *False Positive*, dan *False Negative* yang digunakan untuk mencari nilai *precision* dan *recall* dari hasil pendeteksian manusia. Nilai *precision* dan *recall* kemudian digambarkan ke dalam sebuah kurva yang dinamakan *precision-recall curve*. Nilai *Average Precision* (AP) sendiri didapatkan dari hasil penghitungan area di bawah kurva dari tiap kelas yang terdeteksi oleh sistem. Sedangkan untuk nilai *mean Average Precision* (mAP) didapatkan dengan cara merata-ratakan nilai AP dari seluruh kelas yang terdeteksi.

3.5 Pendeteksian Objek

Pendeteksian objek dilakukan dengan menggunakan model yang telah dilatih menggunakan dataset yang sesuai. Proses ini dimulai dengan memproses input berupa citra dengan ukuran yang merupakan kelipatan dari 32. Semakin besar ukuran gambar maka semakin akurat pula hasil pendeteksian, namun waktu komputasi yang dibutuhkan juga akan semakin lama. Sebaliknya, ukuran gambar yang lebih kecil dapat menghasilkan deteksi yang lebih cepat

namun dengan tingkat akurasi yang lebih rendah. Selanjutnya, dilakukan ekstraksi fitur dimana citra diubah menjadi sekumpulan angka yang dapat diproses oleh komputer. Setelah itu, objek diklasifikasikan berdasarkan angka-angka tersebut. Hasil klasifikasi objek ini akan menghasilkan prediksi yang mencakup koordinat *bounding box*, *confidence score*, dan nama kelas objek (misalnya “person”).

4. HASIL DAN PENGUJIAN

Pada bagian ini dijelaskan hasil perakitan komponen quadcopter dan pengujian sistem pengolahan citra yang sudah didesain.

4.1 Perakitatan Komponen Quadcopter

Pada tahapan ini, dilakukan perakitan komponen quadcopter. Komponen penyusun quadcopter disesuaikan dengan yang sudah ditentukan pada tahap perancangan perangkat dan sistem. Gambar 3 menunjukkan koneksi antar komponen quadcopter.



Gambar 3 Perakitatan Komponen Quadcopter

4.2 Pembuatan Sistem Pendeteksian Manusia

Sistem pendeteksian manusia dibuat dengan menggunakan algoritma YOLO. Untuk membangun sistem ini, langkah awalnya adalah mengumpulkan dataset gambar yang kemudian dibagi menjadi set pelatihan (*train set*) dan set validasi (*validation set*). Setelah dataset gambar terkumpul, langkah berikutnya adalah melakukan proses pelabelan pada satu kelas objek, yaitu ‘person’. Setelah pelabelan selesai, dilakukan pelatihan model menggunakan dataset tersebut agar model dapat membuat prediksi untuk mendeteksi objek manusia.

4.1.1. Pembuatan Dataset

Pembuatan dataset dilakukan dengan menyiapkan data atau gambar yang akan di-*training*. Data yang digunakan adalah gambar manusia yang didokumentasikan secara pribadi. Total gambar yang disiapkan adalah sebanyak 500 foto yang didalamnya terdapat objek manusia yang dibagi menjadi *train set* 300 foto dan *validation set* sebanyak 200 gambar. Gambar 4 menunjukkan beberapa contoh gambar yang akan di-*train*.



Gambar 4 Beberapa gambar yang akan di-*train*

4.1.2. Training Data

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) 50. Setelah tiga kali proses *training*, didapatkan nilai mAP terbaik sebesar 91%.

4.1.3. Hasil Pengujian

Tujuan dilakukan pengujian kinerja adalah untuk mengetahui tingkat akurasi model pada YOLOv5. Pada prosesnya, dibutuhkan dataset 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 96%.

Tabel 1 Pengujian kinerja YOLOv5

Load Model	YOLOv5
Gambar yang di-train	300
Waktu proses train	144.0s
Precision	0.909
Recall	0.992
F1-score	0.847
mAP	96%

4.1.4. Hasil perakitan quadcopter

Penelitian ini menghasilkan produk berupa quadcopter dilengkapi dengan sistem pendeteksian manusia serta kendali jarak jauh menggunakan remote control FlySky i6. Gambar 5 menunjukkan quadcopter yang selesai dirakit.



Gambar 5 Hasil Perakitan Quadcopter

4.3 Hasil Pendeteksian Manusia

Pendeteksian manusia dilakukan menggunakan kamera webcam laptop yang di arahkan ke objek. Gambar 6 menunjukkan bahwa objek yang terdeteksi ditandai dengan *bounding box* yang lengkap dengan nama kelas (*person*), dengan nilai *confidence score* 0.93, dimana hal tersebut menunjukkan bahwa sistem yakin sebesar 93% bahwa objek yang tertangkap kamera merupakan manusia. Serta FPS-nya menunjukkan nilai 15.13.



Gambar 6 Hasil pendeteksian manusia menggunakan kamera webcam

4.4 Pengujian Berdasarkan Jarak Kamera

Tabel 2 menunjukkan hasil pengujian berdasarkan jarak kamera.

Tabel 2 Pengujian berdasarkan jarak kamera

Jarak (m)	Keterangan	Confidance Score
1	Terdeteksi	78%
1,5	Terdeteksi	81%
2	Terdeteksi	78%
2.5	Terdeteksi	57%
3	Terdeteksi	91%

3.5	Terdeteksi	90%
4	Terdeteksi	94%
4.5	Terdeteksi	99%
5	Terdeteksi	95%
5.5	Terdeteksi	96%
6	Terdeteksi	94%

4.5 Pengujian Berdasarkan Jumlah Objek

Gambar 7 menunjukkan hasil pengujian berdasarkan jumlah objek yang terdeteksi.



Gambar 7 Pengujian berdasarkan jumlah objek

Pada Gambar 7 terdapat enam orang namun sistem hanya dapat mendeteksi tiga orang. Hal ini disebabkan karena sistem tidak dapat memprediksi citra yang tidak terdeteksi. Seperti yang terlihat pada Gambar 7, orang yang tidak terdeteksi dikarenakan orang yang ada pada gambar sedang berkerumun dan berdekatan.



Gambar 8 Quadcopter saat melakukan uji terbang

5. KESIMPULAN

Berdasarkan uji terbang, quadcopter dapat terbang secara stabil pada ketinggian 860 meter di atas permukaan laut, atau 5 meter di atas permukaan tanah.

Sistem pendeteksian keberadaan manusia yang telah dibuat berhasil mendeteksi objek berupa manusia. Hal ini dikarenakan nilai mAP yang diperoleh dari hasil pengujian kinerja YOLOv5 adalah sebesar 91% yang berarti akurasi sistem dalam mendeteksi objek manusia sudah baik. Waktu pada saat di-train yaitu 144.1 detik untuk 300 gambar. Selanjutnya, hasil pengujian sistem berdasarkan jarak dari kamera ke objek dengan rentang jarak 1 m sampai 6 m memiliki rata-rata *confidence score* sebesar 91.2%. Dapat disimpulkan bahwa akurasi pendeteksian manusia sudah baik sampai jarak 6 m. Sistem juga dapat mendeteksi manusia lebih dari satu dalam satu *frame*.

DAFTAR PUSTAKA

- [1] Y. d. Z. Khairunnas, "Pembuatan Modul Deteksi Objek Manusia Menggunakan Metode YOLO untuk Mobile Robot," no. 10, 2021.
- [2] Arfienda, "Materi Pendamping Memahami Convnet dengan Tensorflow," 18 April 2019. [Online]. Available: <https://algoritma.blog/convolutional-neural-networks-tensorflow/>. [Diakses 30 Juni 2023].
- [3] Trivusi, "Jenis-jenis Operasi Pooling pada Algoritma CNN," 17 September 2022. [Online]. Available: <https://www.trivusi.web.id/2022/07/jenis-jenis-pooling.html>. [Diakses 2023 July 15].
- [4] R. Mulyana, "Implementasi Deteksi Real Time Klasifikasi Jenis Knedaraan Di Indonesia Menggunakan Metode YOLOV5," 2022.
- [5] S. d. S. Mulyawan, "Identifikasi dan Tracking Objek Berbasis Image Processing Secara Real Time".