

Sistem Deteksi Sapi Pada Peternakan Dari Citra Dan Video Uav Menggunakan Algoritma Yolo

1st Fauzan Andaru Kurniadi

Fakultas Teknik Elektro

Universitas Telkom

Bandung, Indonesia

fauzanandaru@student.telkomuniversit
y.ac.id

2nd Casi Setianingsih

Fakultas Teknik Elektro

Universitas Telkom

Bandung, Indonesia

setiacasie@telkomuniversity.ac.id

3rd Randy Erfa Syaputra

Fakultas Teknik Elektro

Universitas Telkom

Bandung, Indonesia

resaputra@telkomuniversity.ac.id

Abstrak—Dengan semakin majunya zaman banyak aspek dalam kehidupan manusia yang berkembang, salah satunya merupakan bidang peternakan yang saat ini semakin besar, dan dengan semakin besarnya peternakan dibutuhkan juga sistem pemantauan hewan ternak yang dapat mendeteksi hewan yang sedang tidak berada didalam kandang dan itulah yang diharapkan dapat dicapai dengan penggunaan YOLOv5 pada citra yang diambil oleh drone. YOLOv5 merupakan model deteksi objek yang digunakan untuk melakukan deteksi pada sapi di peternakan melalui citra yang diambil oleh drone atau UAV, untuk mencapai kemampuan deteksi tersebut dilakukan training menggunakan dataset gambar – gambar sapi dan juga perubahan konfigurasi hyperparameter pada algoritma untuk mendapatkan hasil deteksi yang optimal. Dataset yang digunakan dalam training adalah gambar sapi berjumlah 3131 dan non – sapi berjumlah 836. Konfigurasi hyperparameter paling optimal yang didapatkan adalah batch 32, learning rate 0.01, dan epoch 350 dengan hasil nilai precision 0,943, Recall 0,925, dan mAP 0,831. Hasil pengujian yang didapatkan adalah akurasi pada ketinggian lima meter dan keadaan diam 75%, ketinggian sepuluh meter memiliki rata – rata akurasi 69,1% dan 15 meter memiliki rata – rata akurasi 47,2%. Dalam kondisi drone bergerak pada kecepatan 0,1 m/s hasil akurasi yang didapatkan adalah 75% untuk lima meter, 0% untuk sepuluh meter dan 25% untuk 15 meter.

Kata kunci—YOLO, YOLOv5, Object detection, Training, Hyperparameter, Drone, Dataset

I. PENDAHULUAN

Dengan perkembangan teknologi saat ini berbagai aspek dalam kehidupan manusia semakin berkembang, termasuk di bidang peternakan, karena kemajuan teknologi, peternakan dapat melakukan ekspansi menjadi ukuran yang jauh lebih besar dari zaman sebelumnya dan dengan bertambah besarnya suatu peternakan maka skala operasi pun semakin besar, salah satunya adalah mengembala hewan ternak dimana karena meningkatnya ukuran peternakan maka meningkat juga populasi hewan ternaknya yang menyebabkan pengawasan pada hewan ternak menjadi semakin sulit jika dilakukan secara manual sehingga meningkatkan resiko kehilangan ternak.

Untuk mengurangi resiko kehilangan hewan ternak di suatu peternakan besar maka diperlukan sistem yang dapat melakukan penghitungan ternak dari jarak jauh, untuk itu

dengan mengimplementasikan arsitektur object detection yang sudah ada yaitu, YOLO pada pada sebuah UAV maka peternak dapat mengetahui jumlah hewan ternak yang digembala dari hasil perhitungan algoritma YOLO dari jarak jauh.

YOLO merupakan salah satu algoritma object detection yang biasanya digunakan dalam memproses citra serta mengenali dan membedakan objek yang ada dalam suatu foto atau video sehingga objek dalam citra tersebut dapat diklasifikasikan dan dihitung sebagai output dari pemrosesan citra[1].

Versi algoritma yang digunakan untuk melakukan deteksi sapi dalam penelitian ini merupakan YOLOv5 yang merupakan iterasi algoritma YOLO ke lima yang dibuat dan dikembangkan oleh Glenn Jocher dan tim Ultralytics. YOLOv5 dipilih karena dibandingkan dengan versi – versi YOLO yang sebelumnya, YOLOv5 memiliki akurasi terbaik serta support dari platform dataset seperti Roboflow yang sudah terintegrasi sehingga memudahkan proses training[2].

II. KAJIAN TEORI

A. Computer Vision

Computer vision merupakan kombinasi dari pengenalan image dan pengenalan pola. Output yang diberikan oleh computer vision adalah pengertian suatu konsep dalam suatu gambar oleh mesin. Computer vision dibuat berdasarkan kemampuan manusia dalam memproses informasi yang didapatkan dari hal yang dilihat. Fungsi utama dari computer vision adalah untuk membuat model kecerdasan buatan yang dapat memperoleh informasi dari data gambar[3].

B. Deep Learning

Deep Learning merupakan salah satu bidang cabang dari ilmu kecerdasan buatan, deep learning menggunakan sistem algoritma yang mirip seperti sistem syaraf pada otak manusia yang disebut artificial neural network dimana neuron terhubung dengan satu sama lain yang membentuk jaringan neuron[4] sehingga algoritma tersebut dapat mempelajari sebuah konsep sesuai dengan informasi yang diberikan dan bahkan dapat melampaui kemampuan

manusia karena algoritma deep learning dapat mempelajari informasi dan data berjumlah banyak dalam waktu yang relatif singkat. Deep learning telah di implementasikan di berbagai bidang, contohnya object detection. Dalam computer vision, deep learning biasanya digunakan untuk memproses foto atau video dimana objek yang ada dalam foto atau video tersebut di klasifikasi oleh algoritma[5].

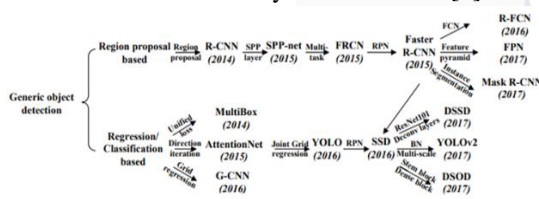
C. Object Detection

Object detection pada dasarnya merupakan sistem algoritma kecerdasan buatan dimana algoritma tersebut diajarkan untuk dapat mengenali objek dalam sebuah foto atau video[6]. Pekerjaan yang biasanya dilakukan oleh algoritma object detection umumnya dibagi ke tiga, yaitu[6]:

1. Seleksi daerah informasi: dimana algoritma melakukan scan pada seluruh image lalu menyeleksi daerah yang berkemungkinan memiliki objek
2. Ekstraksi fitur: dimana algoritma melakukan pencarian fitur objek dari hasil scan untuk menentukan objek dalam foto atau video.
3. Klasifikasi: setelah objek ditentukan berdasarkan fiturnya dari dataset, kemudian objek tersebut di klasifikasikan berdasarkan kelas yang ada di dataset.

Algoritma object detection berdasarkan deep learning saat ini dapat dibagi menjadi dua macam framework, yaitu:

4. Region Proposal Based Framework yang bekerja dengan cara melakukan scan kasar secara keseluruhan lalu memfokuskan scan ke area yang diperkirakan penting dan memiliki objek.
5. Regression/Classification Based Framework merupakan framework satu langkah yang bekerja dengan cara memetakan pixel pada gambar ke koordinat bounding box dan juga probabilitas kelas sehingga objek dalam gambar dapat di identifikasi berdasarkan klasifikasinya dalam dataset[6].



GAMBAR 1
Jenis – Jenis Object Detection

D. Convolutional Neural Network

Convolutional neural network atau CNN merupakan jaringan neural network yang di desain untuk mengolah gambar[4] dan biasanya digunakan untuk deteksi objek, CNN dapat melakukan optimasi tanpa bantuan manusia melalui training[7] CNN terdiri dari tiga lapisan yang digunakan untuk mengolah data yang di input, lapisan tersebut adalah:

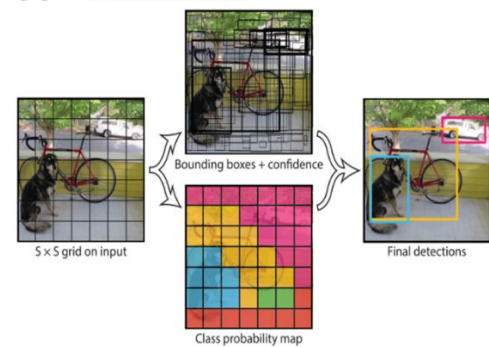
1. Convolution Layer bertugas untuk melakukan ekstraksi fitur dari data yang selanjutnya akan dimasukkan ke dalam filter untuk menentukan feature map dari data yang selanjutnya di output ke pooling layer.

2. Pooling layer bertugas untuk mengurangi dimensi dari output dari layer sebelumnya dengan menggabungkan fitur lalu dilanjutkan ke layer selanjutnya.

3. Fully connected layer merupakan lapisan atau layer yang bertugas untuk memberikan nilai kelas dari data yang didapat dari layer sebelumnya untuk nantinya digunakan untuk klasifikasi objek[7].

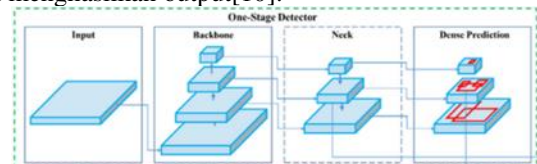
E. YOLO Algorithm

YOLO atau You Only Look Once merupakan algoritma object detection yang digunakan untuk melakukan deteksi dan klasifikasi secara real-time pada sebuah objek dalam sebuah gambar yang bekerja dengan menggunakan satu jaringan neural network[1] atau bisa disebut single-stage object detection[8]. YOLO melakukan deteksi dengan cara membagi gambar menjadi kotak – kotak yang masing – masing akan melakukan prediksi dan menentukan bounding box untuk objek, disaat yang bersamaan kotak – kotak tersebut akan melakukan prediksi kelas bagi objek dalam bounding box tersebut lalu hasil prediksi yang memiliki confidence score terbaik akan menjadi bounding box yang ditampilkan untuk menampilkan objek yang dideteksi dan memisahkan dari latar belakang dan objek lainnya[1]. Confidence Score merupakan nilai besarnya keyakinan model pada hasil deteksi, confidence score juga dapat dikonfigurasikan untuk memberi batasan nilai confidence minimal yang dapat ditampilkan pada hasil akhir deteksi[9].



GAMBAR 2
Overview Cara Kerja Algoritma Yolo

Algoritma YOLO merupakan algoritma deteksi objek single-stage yang memiliki arsitektur neural network yang terdiri dari Backbone yang berfungsi untuk ekstraksi fitur dari data, Neck yang berfungsi untuk mengumpulkan menggabungkan fitur dari data, dan Head yang berfungsi untuk melakukan deteksi berdasarkan data yang diberikan dan menghasilkan output[10].

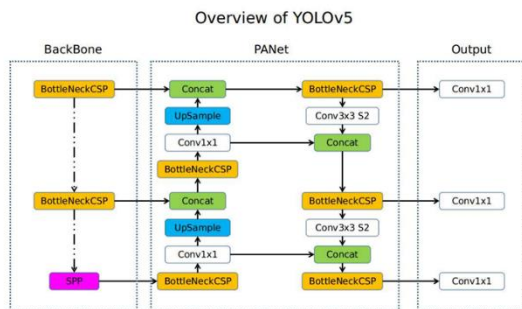


GAMBAR 3
Arsitektur Yolo

F. YOLOv5

YOLOv5 merupakan iterasi terbaru dari algoritma YOLO yang dibuat oleh Glenn Jocher dan tim Ultralytics. YOLOv5 tersedia secara open source di github dan dipilih karena dibandingkan dengan versi – versi YOLO yang

sebelumnya, YOLOv5 memiliki akurasi deteksi yang sangat baik serta support dari platform dataset seperti Roboflow yang sudah terintegrasi sehingga memudahkan proses training. YOLOv5 memiliki arsitektur Backbone: New CSP-Darknet53, Neck: PANet atau Path Aggregation Network, New CSP-PA, Head: YOLO[11].



GAMBAR 4
Arsitektur YOLOv5

G. Hyperparameter

Hyperparameter merupakan parameter pada model machine learning dan deep learning yang ditentukan sebelum memulai training untuk sebuah model, hyperparameter biasanya digunakan untuk melakukan konfigurasi pada model neural network agar bisa mendapatkan hasil yang optimal, konfigurasi ini disebut sebagai hypermeter tuning dan merupakan kunci dalam membuat model deep learning yang efektif[12]. Beberapa contoh hyperparameter pada YOLOv5 yang dapat di konfigurasi untuk dapat mengoptimalkan model yaitu, batch yang menentukan jumlah data yang diproses sebelum parameter model di update, epoch yang merupakan berapa kali model memproses seluruh data dalam sekali training dan learning rate merupakan seberapa cepat sebuah model dapat beradaptasi terhadap informasi dan melakukan koreksi saat terjadi kesalahan[12].

H. Confusion Matrix

Confusion Matrix merupakan metode untuk menentukan performa sebuah model machine learning dalam melakukan klasifikasi objek. Hasil deteksi model dimasukkan kedalam sebuah tabel prediksi benar dan salah lalu dibandingkan dengan kelas sebenarnya[13]. Tabel tersebut merupakan sebagai berikut:

TABEL 1
CONFUSION MATRIX

		Actual Class	
		Positive	Negative
Predicted Class	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True negative (TN)

Variabel yang ada dalam Confusion Matrix adalah sebagai berikut:

True Positive (TP): prediksi dari model yang menghasilkan

klasifikasi positif dan prediksi tersebut memang akurat berdasarkan kelas yang ada.

False Positive (FP): prediksi dari model yang menunjukkan klasifikasi positif namun bukan objek dari kelas yang dideteksi.

False Negative (FN): Merupakan saat dimana model tidak mendapatkan deteksi apapun dari kelas yang seharusnya dideteksi.

True Negative (TN): objek yang bukan termasuk ke dalam kelas yang seharusnya dideteksi dan juga tidak terdeteksi oleh model.

Rumus Akurasi *Confusion Matrix*:

Untuk dapat menghitung akurasi sebuah model *machine learning* atau pun *deep learning* menggunakan *confusion matrix* diperlukan rumus berikut:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

Selain akurasi, *confusion matrix* juga memiliki rumus lain untuk menghitung berbagai aspek sebuah model seperti *precision* dan *recall*. *Precision* merupakan rumus yang digunakan untuk menghitung jumlah hasil prediksi yang memang benar bernilai positif, rumusnya adalah sebagai berikut:

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

Recall merupakan rumus yang digunakan untuk menghitung jumlah positif yang memang masuk ke dalam prediksi dan yang seharusnya masuk prediksi tapi tidak masuk, rumusnya adalah sebagai berikut:

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

I. Dataset

Dataset merupakan sekumpulan data berupa gambar yang dipisahkan berdasarkan kelas yang merupakan pengelompokan berdasarkan jenis objek dalam gambar yang dijadikan fokus untuk training object detection. Dataset yang digunakan dalam training dapat terdiri dari berbagai jumlah gambar dan gambar tersebut dapat terdiri dari berbagai macam kelas, tergantung kebutuhan untuk training. MS COCO merupakan sumber dataset yang berisi 328.000 gambar dan 80 kelas objek yang sering digunakan oleh banyak developer algoritma object detection dengan YOLOv5 sebagai salah satunya. PASCAL VOC merupakan salah satu dataset seperti MS COCO yang digunakan dalam object detection, dataset ini berisi 11.000 gambar dan 20 kelas objek[14]. Dataset yang digunakan dalam training pada sebuah algoritma machine learning atau pun deep learning biasanya dibagi menjadi tiga jenis set, yaitu Training Set data yang digunakan untuk training serta pembelajaran untuk model agar dapat mengenali fitur dan pola pada data yang diberikan. Validation Set merupakan data yang digunakan untuk melakukan evaluasi pada model saat training, evaluasi biasanya dilakukan setiap akhir epoch dan akan membantu model untuk tidak overfitting dimana model dapat dengan akurat mengklasifikasikan data di training set namun kesulitan melakukan klasifikasi data yang belum pernah dilihat. Test set seperti merupakan data yang digunakan untuk menentukan performa akhir sebuah model setelah menyelesaikan training[15].

J. Unmanned Aerial Vehicle

Unmanned Aerial Vehicle (UAV) dapat disebut juga sebagai Drone merupakan pesawat tanpa alat yang dapat dikendalikan dari jarak jauh menggunakan ground control station (GCS) serta berbagai sensor dan platform yang memudahkan komunikasi antara keduanya. UAV dulunya digunakan untuk kepentingan militer dan sipil seperti pencarian dan pertolongan (SAR), monitoring cuaca, dan pengawasan, namun dengan semakin berkembangnya teknologi seperti internet, UAV sekarang digunakan di lebih banyak bidang, seperti agrikultur, inspeksi, sistem pembangunan, sistem pengiriman, transportasi, dan monitoring[16].



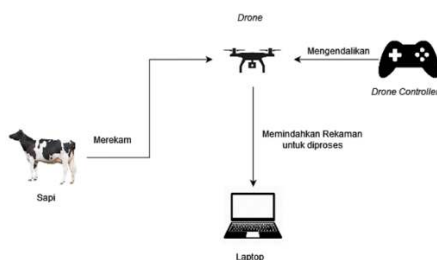
GAMBAR 5
Uav

III. METODE

A. Desain Sistem

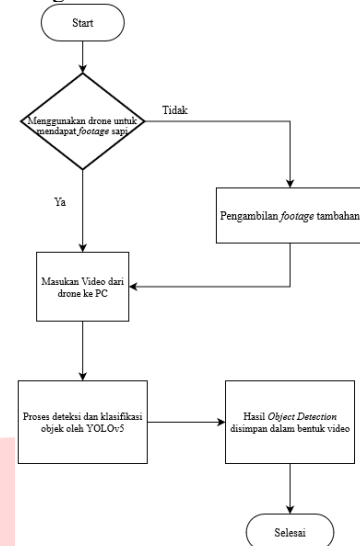
Sistem dirancang menggunakan Arsitektur YOLOv5 yang sudah tersedia namun belum diberikan training, model YOLOv5 tersebut diberikan custom training berdasarkan dataset yang dibutuhkan yaitu sapi. Untuk mendapatkan dataset yang diperlukan, ratusan hingga ribuan foto yang berisi sapi didalamnya didapatkan melalui internet, selanjutnya foto-foto tersebut di berikan label dimana label tersebut menandakan dimana subjek yang nantinya akan dideteksi.

Model YOLOv5 yang sudah trained selanjutnya digunakan pada percobaan untuk melakukan klasifikasi pada video sapi yang ditangkap menggunakan drone dari internet untuk mengetahui apakah sistem dapat berfungsi dengan benar dan telah dapat melakukan klasifikasi objek pada sapi, selanjutnya setelah didapatkan model dengan akurasi yang paling optimal dari hasil training dilakukan pengambilan data secara langsung ke peternakan dimana sapi dikeluarkan dari kandangnya agar UAV dapat merekam sapi, selanjutnya hasil rekaman dipindahkan dari UAV ke laptop untuk melakukan deteksi.



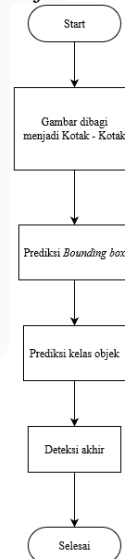
GAMBAR 6
Desain Sistem

B. Flowchart Pengambilan Data



Sistem dimulai dengan menerbangkan drone untuk mendapatkan footage sapi di peternakan, drone di terbangkan hingga mendapatkan sapi sebagai objek yang ada di dalamnya, setelah data statistik dari drone dan juga video didapatkan diterbangkan kembali untuk mendapatkan data dari kondisi yang berbeda. setelah didapatkan, footage sapi dan data dari drone dipindahkan ke laptop yang sudah di install dengan YOLOv5 Custom yang di train untuk melakukan deteksi dan klasifikasi untuk sapi sehingga dapat menampilkan outline kotak pada sapi dalam video.

C. Flowchart Deteksi Objek



Deteksi dimulai dengan gambar yang di input ke YOLOv5 dibagi menjadi kotak – kotak yang masing – masing akan melakukan prediksi dan menentukan bounding box untuk objek, disaat yang bersamaan kotak – kotak tersebut akan melakukan prediksi kelas bagi objek dalam bounding box tersebut lalu hasil prediksi yang memiliki confidence score terbaik akan menjadi bounding box yang ditampilkan untuk menampilkan objek yang dideteksi.

D. Persiapan Dataset

Dataset yang digunakan untuk Training YOLOv5 merupakan gambar sapi yang didapatkan dari berbagai sumber di internet yang lalu di beri label secara otomatis menggunakan Roboflow untuk mempercepat proses. Selain data untuk sapi digunakan juga gambar kucing dan gambar manusia yang diberi label secara manual dan diberi label null yang menandakan gambar tersebut telah diberi label namun tidak memiliki koordinat bounding box didalamnya dimana hal tersebut akan menandakan pada algoritma bahwa tidak ada objek yang dicari di dalam gambar, hal tersebut dilakukan untuk mengajarkan pada algoritma bahwa tidak semua dataset memiliki objek yang dicari, metode ini dinamakan dengan negative training.

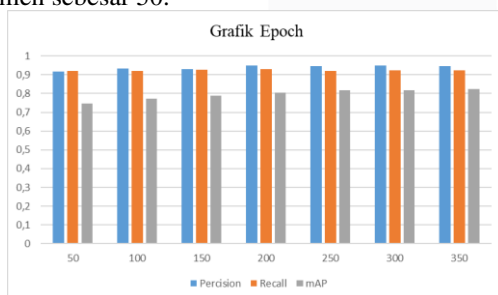
E. Metode Training YOLOv5

Training untuk YOLOv5 dilakukan di platform Google Colab dimana algoritma di-train menggunakan dataset sapi yang telah diberikan label. Algoritma YOLOv5 di train dengan menggunakan beberapa hyperparameter yang berbeda untuk mendapatkan hasil yang paling optimal, hyperparameter yang di ujikan adalah batch, learning rate, dan epoch dimana batch menentukan jumlah data yang diproses sebelum parameter YOLOv5 di update dan epoch merupakan berapa kali YOLOv5 memproses seluruh data dalam sekali training dan learning rate merupakan seberapa cepat sebuah model dapat beradaptasi terhadap informasi dan melakukann koreksi saat terjadi kesalahan.

IV. HASIL DAN PEMBAHASAN

A. Pengujian Epoch

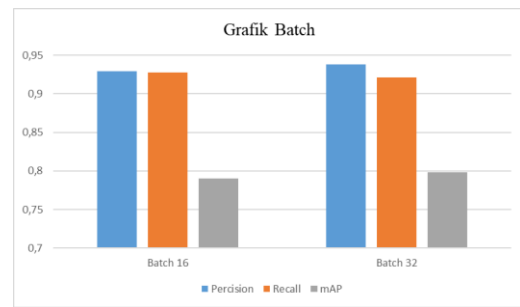
Pengujian epoch dilakukan sebanyak tujuh kali dimana epoch yang diujikan dimulai dari 50 hingga 350 dengan inkremen sebesar 50.



Berdasarkan hasil yang didapatkan dari pengujian epoch, konfigurasi epoch yang optimal dan digunakan pada deteksi sapi di lapangan adalah model dengan epoch 350 yang memiliki nilai precision 0,947, nilai recall 0,924, dan nilai mAP 0,823.

B. Pengujian Batch

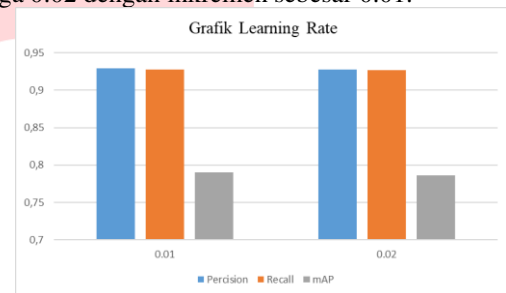
Pengujian batch dilakukan sebanyak dua kali dimana batch yang diujikan dimulai dari 16 hingga 32 dengan inkremen sebesar 16.



Berdasarkan hasil yang didapatkan dari pengujian batch, konfigurasi batch yang paling optimal dan digunakan pada deteksi sapi di lapangan adalah model dengan batch 32 dengan nilai precision 0,938, nilai recall 0,921, dan mAP 0,798. Karena model tersebut memberikan nilai – nilai hasil yang terbaik.

C. Pengujian Learning Rate

Pengujian learning rate dilakukan sebanyak dua kali dimana learning rate yang diujikan dimulai dari 0.01 hingga 0.02 dengan inkremen sebesar 0.01.



Berdasarkan hasil yang didapatkan dari pengujian learning rate, konfigurasi learning rate yang optimal dan digunakan pada deteksi sapi di lapangan adalah model dengan learning rate 0,01 dengan nilai precision 0,929, recall 0,928, dan mAP 0,79.

D. Analisis Hasil Pengujian Hyperparameter

Konfigurasi hyperparameter yang menghasilkan model paling optimal adalah batch 32, learning rate 0.01, dan epoch 350 dengan hasil nilai precision 0.943; Recall 0.925; dan mAP 0.831 nilai tersebut merupakan nilai maksimal yang bisa didapatkan melalui training dengan kondisi spesifikasi hardware dan software dalam penelitian ini.

E. Akurasi Pada Ketinggian 5 Meter Kondisi Diam

Percobaan pertama pada ketinggian 5 meter dilakukan pada tiga sudut kamera berbeda untuk pengambilan data video sapi di peternakan yaitu sudut 0°, sudut 45°, dan sudut 60° dan data dibagi menjadi *True positive*, *True negative*, *False positive*, dan *False negative*.

$$0^\circ \text{ Percobaan 1} = \frac{2 + 0}{2 + 0 + 0 + 0} = \frac{2}{2} * 100\% = 100\%$$

$$45^\circ \text{ Percobaan 1} = \frac{1 + 0}{1 + 0 + 1 + 1} = \frac{1}{3} * 100\% = 33.3\%$$

$$60^\circ \text{ Percobaan 1} = \frac{1 + 0}{1 + 0 + 0 + 1} = \frac{1}{2} * 100\% = 50\%$$

Dari hasil penghitungan akurasi pada percobaan pertama di ketinggian 5 meter menggunakan rumus akurasi dari *confusion matrix* berdasarkan data dari video sapi yang diambil di peternakan menghasilkan nilai akurasi dari

sudut 0° sebesar 100%, sudut 45° sebesar 33.3%, dan sudut 60° sebesar 50%.

Percobaan kedua pada ketinggian 5 meter dilakukan pada tiga sudut kamera berbeda untuk pengambilan data video sapi di peternakan yaitu sudut 0°, sudut 45°, dan sudut 60° dan data dibagi menjadi *True positive*, *True negative*, *False positive*, dan *False negative*.

$$0^\circ \text{ Percobaan 2} = \frac{2+0}{2+0+0+0} = \frac{2}{2} * 100\% = 100\%$$

$$45^\circ \text{ Percobaan 2} = \frac{2+0}{2+0+1+0} = \frac{2}{3} * 100\% = 66.7\%$$

$$60^\circ \text{ Percobaan 2} = \frac{2+0}{2+0+0+0} = \frac{2}{2} * 100\% = 100\%$$

Dari hasil penghitungan akurasi pada percobaan kedua di ketinggian 5 meter menggunakan rumus akurasi dari *confusion matrix* berdasarkan data dari video sapi yang diambil di peternakan menghasilkan nilai akurasi dari sudut 0° sebesar 100%, sudut 45° sebesar 66.7%, dan sudut 60° sebesar 100%.

$$\text{Rata - rata akurasi} = \frac{100+33.3+50+100+66.7+100}{6} = \frac{450}{6} = 75\%$$

Setelah hasil akurasi masing – masing dari sudut 0°, 45°, dan 60° didapatkan dari percobaan pertama dan kedua, selanjutnya hasil akurasi dari kedua percobaan tersebut di rata – ratakan untuk mendapatkan nilai rata – rata akurasi deteksi YOLOv5 di ketinggian 5 meter. Rata – rata akurasi yang didapat adalah 75%.

F. Akurasi Pada Ketinggian 10 Meter Kondisi Diam

Percobaan pertama pada ketinggian 10 meter dilakukan pada tiga sudut kamera berbeda untuk pengambilan data video sapi di peternakan yaitu sudut 0°, sudut 45°, dan sudut 60° dan data dibagi menjadi *True positive*, *True negative*, *False positive*, dan *False negative*.

$$0^\circ \text{ Percobaan 1} = \frac{1+0}{1+0+0+1} = \frac{1}{2} * 100\% = 50\%$$

$$45^\circ \text{ Percobaan 1} = \frac{2+0}{2+0+2+0} = \frac{2}{4} * 100\% = 50\%$$

$$60^\circ \text{ Percobaan 1} = \frac{2+0}{2+0+0+0} = \frac{2}{2} * 100\% = 100\%$$

Dari hasil penghitungan akurasi pada percobaan pertama di ketinggian 10 meter menggunakan rumus akurasi dari *confusion matrix* berdasarkan data dari video sapi yang diambil di peternakan menghasilkan nilai akurasi dari sudut 0° sebesar 50%, sudut 45° sebesar 50%, dan sudut 60° sebesar 100%.

Percobaan kedua pada ketinggian 10 meter dilakukan pada tiga sudut kamera berbeda untuk pengambilan data video sapi di peternakan yaitu sudut 0°, sudut 45°, dan sudut 60° dan data dibagi menjadi *True positive*, *True negative*, *False positive*, dan *False negative*.

$$0^\circ \text{ Percobaan 2} = \frac{2+0}{2+0+0+0} = \frac{2}{2} * 100\% = 100\%$$

$$45^\circ \text{ Percobaan 2} = \frac{2+0}{2+0+1+0} = \frac{2}{3} * 100\% = 66.7\%$$

$$60^\circ \text{ Percobaan 2} = \frac{1+0}{1+0+0+1} = \frac{1}{2} * 100\% = 50\%$$

Dari hasil penghitungan akurasi pada percobaan kedua di ketinggian 10 meter menggunakan rumus akurasi dari *confusion matrix* berdasarkan data dari video sapi yang diambil di peternakan menghasilkan nilai akurasi dari sudut 0° sebesar 100%, sudut 45° sebesar 66.7%, dan sudut 60° sebesar 50%.

Rata – rata akurasi

$$= \frac{50+50+100+100+66.7+50}{6} = \frac{416.7}{6} = 69.4\%$$

Setelah hasil akurasi masing – masing dari sudut 0°, 45°, dan 60° didapatkan dari percobaan pertama dan kedua, selanjutnya hasil akurasi dari kedua percobaan tersebut di rata – ratakan untuk mendapatkan nilai rata – rata akurasi deteksi YOLOv5 di ketinggian 10 meter. Rata – rata akurasi yang didapat adalah 69.4%.

G. Akurasi Pada Ketinggian 15 Meter Kondisi Diam

Percobaan pertama pada ketinggian 15 meter dilakukan pada tiga sudut kamera berbeda untuk pengambilan data video sapi di peternakan yaitu sudut 0°, sudut 45°, dan sudut 60° dan data dibagi menjadi *True positive*, *True negative*, *False positive*, dan *False negative*.

$$0^\circ \text{ Percobaan 1} = \frac{1+0}{1+0+0+1} = \frac{1}{2} * 100\% = 50\%$$

$$45^\circ \text{ Percobaan 1} = \frac{2+0}{2+0+2+0} = \frac{2}{4} * 100\% = 50\%$$

$$60^\circ \text{ Percobaan 1} = \frac{2+0}{2+0+0+0} = \frac{2}{2} * 100\% = 100\%$$

Dari hasil penghitungan akurasi pada percobaan pertama di ketinggian 15 meter menggunakan rumus akurasi dari *confusion matrix* berdasarkan data dari video sapi yang diambil di peternakan menghasilkan nilai akurasi dari sudut 0° sebesar 50%, sudut 45° sebesar 50%, dan sudut 60° sebesar 100%.

Percobaan kedua pada ketinggian 15 meter dilakukan pada tiga sudut kamera berbeda untuk pengambilan data video sapi di peternakan yaitu sudut 0°, sudut 45°, dan sudut 60° dan data dibagi menjadi *True positive*, *True negative*, *False positive*, dan *False negative*.

$$0^\circ \text{ Percobaan 2} = \frac{1+0}{1+0+1+1} = \frac{1}{3} * 100\% = 33.3\%$$

$$45^\circ \text{ Percobaan 2} = \frac{0+0}{0+0+1+2} = \frac{0}{3} * 100\% = 0\%$$

$$60^\circ \text{ Percobaan 2} = \frac{1+0}{1+0+0+1} = \frac{1}{2} * 100\% = 50\%$$

Dari hasil penghitungan akurasi pada percobaan kedua di ketinggian 15 meter menggunakan rumus akurasi dari *confusion matrix* berdasarkan data dari video sapi yang

diambil di peternakan menghasilkan nilai akurasi dari sudut 0° sebesar 33.3%, sudut 45° sebesar 0%, dan sudut 60° sebesar 50%.

$$\text{Rata - rata akurasi} = \frac{50+50+100+33.3+0+50}{6} = \frac{283.3}{6} = 47.2\%$$

Setelah hasil akurasi masing – masing dari sudut 0°, 45°, dan 60° didapatkan dari percobaan pertama dan kedua, selanjutnya hasil akurasi dari kedua percobaan tersebut di rata – ratakan untuk mendapatkan nilai rata – rata akurasi deteksi YOLOv5 di ketinggian 15 meter. Rata – rata akurasi yang didapat adalah 47.2%.

H. Akurasi Pada Ketinggian 5 Meter Kecepatan 0.1 m/s

Percobaan pertama pada ketinggian 5 meter dilakukan pada kondisi *drone* bergerak dengan kecepatan 0.1 m/s untuk pengambilan data video sapi di peternakan selanjutnya data deteksi yang didapatkan dibagi menjadi *True positive*, *True negative*, *False positive*, dan *False negative*.

$$\text{5 Meter Percobaan 1} = \frac{1+0}{1+0+0+1} = \frac{1}{2} * 100\% = 50\%$$

Dari hasil penghitungan akurasi pada percobaan pertama di ketinggian 5 meter menggunakan rumus akurasi dari *confusion matrix* berdasarkan data dari video sapi yang diambil di peternakan menghasilkan nilai akurasi dari kondisi bergerak dengan kecepatan 0.1 m/s adalah 50%. Percobaan kedua pada ketinggian 5 meter dilakukan pada kondisi *drone* bergerak dengan kecepatan 0.1 m/s untuk pengambilan data video sapi di peternakan selanjutnya data deteksi yang didapatkan dibagi menjadi *True positive*, *True negative*, *False positive*, dan *False negative*.

$$\text{5 Meter Percobaan 2} = \frac{2+0}{2+0+0+0} = \frac{2}{2} * 100\% = 100\%$$

Dari hasil penghitungan akurasi pada percobaan kedua di ketinggian 5 meter menggunakan rumus akurasi dari *confusion matrix* berdasarkan data dari video sapi yang diambil di peternakan menghasilkan nilai akurasi dari kondisi bergerak dengan kecepatan 0.1 m/s adalah 100%.

$$\text{Rata - rata akurasi} = \frac{50+100}{2} = \frac{150}{2} * 100\% = 75\%$$

Setelah hasil akurasi pada kecepatan 0.1 m/s dari percobaan pertama dan kedua, selanjutnya hasil akurasi dari kedua percobaan tersebut di rata – ratakan untuk mendapatkan nilai rata – rata akurasi deteksi YOLOv5 di ketinggian 5 meter. Rata – rata akurasi yang didapat adalah 75%.

I. Akurasi Pada Ketinggian 10 Meter Kecepatan 0.1 m/s

Percobaan pertama pada ketinggian 10 meter dilakukan pada kondisi *drone* bergerak dengan kecepatan 0.1 m/s untuk pengambilan data video sapi di peternakan selanjutnya data deteksi yang didapatkan dibagi menjadi *True positive*, *True negative*, *False positive*, dan *False negative*.

$$\text{10 Meter Percobaan 1} = \frac{0+0}{0+0+0+2} = \frac{0}{2} * 100\% = 0\%$$

Dari hasil penghitungan akurasi pada percobaan pertama di ketinggian 10 meter menggunakan rumus akurasi dari *confusion matrix* berdasarkan data dari video sapi yang diambil di peternakan menghasilkan nilai akurasi dari kondisi bergerak dengan kecepatan 0.1 m/s adalah 0%.

Percobaan kedua pada ketinggian 10 meter dilakukan pada kondisi *drone* bergerak dengan kecepatan 0.1 m/s untuk pengambilan data video sapi di peternakan selanjutnya data deteksi yang didapatkan dibagi menjadi *True positive*, *True negative*, *False positive*, dan *False negative*.

$$\text{10 Meter Percobaan 2} = \frac{0+0}{0+0+1+2} = \frac{0}{3} * 100\% = 0\%$$

Dari hasil penghitungan akurasi pada percobaan kedua di ketinggian 10 meter menggunakan rumus akurasi dari *confusion matrix* berdasarkan data dari video sapi yang diambil di peternakan menghasilkan nilai akurasi dari kondisi bergerak dengan kecepatan 0.1 m/s adalah 0%.

Rata – rata akurasi = 0%

Setelah hasil akurasi pada kecepatan 0.1 m/s dari percobaan pertama dan kedua, selanjutnya hasil akurasi dari kedua percobaan tersebut di rata – ratakan untuk mendapatkan nilai rata – rata akurasi deteksi YOLOv5 di ketinggian 10 meter. Rata – rata akurasi yang didapat adalah 0%.

J. Akurasi Pada Ketinggian 15 Meter Kecepatan 0.1 m/s

Percobaan pertama pada ketinggian 15 meter dilakukan pada kondisi *drone* bergerak dengan kecepatan 0.1 m/s untuk pengambilan data video sapi di peternakan selanjutnya data deteksi yang didapatkan dibagi menjadi *True positive*, *True negative*, *False positive*, dan *False negative*.

$$\text{15 Meter Percobaan 1} = \frac{2+0}{2+0+2+0} = \frac{2}{4} * 100\% = 50\%$$

Dari hasil penghitungan akurasi pada percobaan pertama di ketinggian 15 meter menggunakan rumus akurasi dari *confusion matrix* berdasarkan data dari video sapi yang diambil di peternakan menghasilkan nilai akurasi dari kondisi bergerak dengan kecepatan 0.1 m/s adalah 50%.

Percobaan kedua pada ketinggian 15 meter dilakukan pada kondisi *drone* bergerak dengan kecepatan 0.1 m/s untuk pengambilan data video sapi di peternakan selanjutnya data deteksi yang didapatkan dibagi menjadi *True positive*, *True negative*, *False positive*, dan *False negative*.

$$\text{15 Meter Percobaan 2} = \frac{0+0}{0+0+1+2} = \frac{0}{3} * 100\% = 0\%$$

Dari hasil penghitungan akurasi pada percobaan kedua di ketinggian 15 meter menggunakan rumus akurasi dari *confusion matrix* berdasarkan data dari video sapi yang diambil di peternakan menghasilkan nilai akurasi dari kondisi bergerak dengan kecepatan 0.1 m/s adalah 0%.

$$\text{Rata - rata akurasi} = \frac{50+0}{2} = \frac{50}{2} * 100\% = 25\%$$

Setelah hasil akurasi pada kecepatan 0.1 m/s dari percobaan pertama dan kedua, selanjutnya hasil akurasi dari kedua percobaan tersebut di rata – ratakan untuk mendapatkan nilai rata – rata akurasi deteksi YOLOv5 di ketinggian 15 meter. Rata – rata akurasi yang didapat adalah 25%.

V. KESIMPULAN

A. Kesimpulan

Kesimpulan yang dapat diambil dari hasil pengujian dan analisis pada penelitian Tugas Akhir adalah sebagai berikut:

1. YOLOv5 telah dapat melakukan deteksi dan klasifikasi pada sapi yang ada di peternakan dengan drone pada ketinggian dan posisi yang beragam setelah melakukan training.
2. Akurasi dari YOLOv5 saat percobaan langsung dilapangan didapatkan bahwa algoritma dapat melakukan deteksi dengan akurasi yang cukup tinggi pada ketinggian lima meter saat diam sebesar 75%, ketinggian sepuluh meter memiliki rata – rata akurasi 69,1% dan 15 meter memiliki rata – rata akurasi 47.2%. Dalam kondisi UAV bergerak pada kecepatan 0,1 m/s hasil akurasi yang didapatkan adalah 75% untuk lima meter, 0% untuk sepuluh meter dan 25% untuk 15 meter.

B. Saran

Berikut merupakan saran yang dapat disampaikan agar penelitian selanjutnya bisa mendapatkan hasil yang lebih baik lagi:

1. Memiliki drone dengan waktu operasional yang lebih lama karena waktu operasional dapat membatasi proses pengambilan data.
2. Memiliki hardware dengan spesifikasi GPU lebih baik dari RTX 2060 dan RAM lebih besar dari 32gb untuk dapat melakukan training secara local.
3. Menambahkan sistem agar hasil deteksi YOLOv5 pada citra dari drone menjadi real-time.

REFERENSI

- [1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 779–788, 2016, doi: 10.1109/CVPR.2016.91.
- [2] "YOLOv5 Documentation." <https://ultralytics.github.io/> (accessed Jan. 20, 2023).
- [3] V. Wiley and T. Lucas, "Computer Vision and Image Processing : A Paper Review," vol. 2, no. 1, pp. 28–36, 2018, doi: 10.29099/ijair.v2i1.42.
- [4] P. A. Nugroho, I. Fenriana, R. Arijanto, and M. Kom, "IMPLEMENTASI DEEP LEARNING MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK (CNN) PADA EKSPRESI MANUSIA," vol. 1, 2020.
- [5] L. Alzubaidi *et al.*, *Review of deep learning : concepts, CNN architectures, challenges, applications, future directions*. Springer International Publishing, 2021.
- [6] Z. Q. Zhao, P. Zheng, S. T. Xu, and X. Wu, "Object Detection with Deep Learning: A Review," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 30, no. 11, pp. 3212–3232, 2019, doi: 10.1109/TNNLS.2018.2876865.
- [7] K. O'Shea and R. Nash, "An Introduction to Convolutional Neural Networks," pp. 1–11, 2015, [Online]. Available: <http://arxiv.org/abs/1511.08458>.
- [8] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," 2020, [Online]. Available: <http://arxiv.org/abs/2004.10934>.
- [9] S. Wenkel, K. Alhazmi, T. Liiv, S. Alrshoud, and M. Simon, "Confidence score: The forgotten dimension of object detection performance evaluation," *Sensors*, vol. 21, no. 13, 2021, doi: 10.3390/s21134350.
- [10] "YOLOv4 - An explanation of how it works." <https://blog.roboflow.com/a-thorough-breakdown-of-yolov4/> (accessed Jan. 06, 2023).
- [11] "Overview of model structure about YOLOv5 · Issue #280 · ultralytics/yolov5 · GitHub." <https://github.com/ultralytics/yolov5/issues/280> (accessed Jan. 05, 2023).
- [12] L. Yang and A. Shami, "On hyperparameter optimization of machine learning algorithms: Theory and practice," *Neurocomputing*, vol. 415, pp. 295–316, 2020, doi: 10.1016/j.neucom.2020.07.061.
- [13] Z. Karimi, "Confusion Matrix," no. October, 2021.
- [14] Y. Pang, J. Cao, Y. Li, J. Xie, H. Sun, and J. Gong, "TJU-DHD : A Diverse High-Resolution Dataset for Object Detection," pp. 1–13.
- [15] Y. Xu and R. Goodacre, "On Splitting Training and Validation Set: A Comparative Study of Cross-Validation, Bootstrap and Systematic Sampling for Estimating the Generalization Performance of Supervised Learning," *Journal of Analysis and Testing*, vol. 2, no. 3, pp. 249–262, 2018, doi: 10.1007/s41664-018-0068-2.
- [16] F. Ahmed, J. C. Mohanta, A. Keshari, and P. S. Yadav, "Recent Advances in Unmanned Aerial Vehicles: A Review," *Arab. J. Sci. Eng.*, vol. 47, no. 7, pp. 7963–7984, 2022, doi: 10.1007/s13369-022-06738-0.