

**DETEKSI TELUR BEBEK *FERTIL* DAN  
*INFERTIL* MENGGUNAKAN  
ALGORITMA *CONVOLUTIONAL NEURAL NETWORK*  
(*CNN*)**

**TUGAS AKHIR**

**diajukan untuk memenuhi salah satu syarat  
memperoleh gelar sarjana  
pada Program Studi Teknik Informatika**



oleh:

**RICKY STEVEN CHANDRA**

**19416255201102**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BUANA PERJUANGAN KARAWANG  
2023**

## **LEMBAR PERSETUJUAN**

### **DETEKSI TELUR *BEBEK FERTIL* DAN *INFERTIL* MENGGUNAKAN ALGORITMA *CONVOLUTIONAL NEURAL NETWORK (CNN)***

#### ***DETECTION OF FERTIL AND INFERTIL DUCK EGG USING CONVOLUTIONAL NEURAL NETWORK (CNN) ALGORITHM***

Tugas Akhir diajukan oleh :

**Ricky Steven Chandra**

**19416255201102**

Program Studi Teknik Informatika  
Fakultas Ilmu Komputer  
Universitas Buana Perjuangan Karawang

Karawang, 29 Juli 2023

Menyetujui :

Pembimbing I,

Pembimbing II,



**Dr. Hanny Hikmayanti.H, M.Kom**

NIDN: 0427037305

**Yana Cahyana, M.Kom**

NIDN: 0410077901

## **LEMBAR PENGESAHAN**

### **DETEKSI TELUR *BEBEK FERTIL* DAN *INFERTIL* MENGGUNAKAN ALGORITMA *CONVOLUTIONAL NEURAL NETWORK (CNN)***

#### ***DETECTION OF FERTIL AND INFERTIL DUCK EGG USING CONVOLUTIONAL NEURAL NETWORK (CNN) ALGORITHM***

oleh :

**Ricky Steven Chandra**

**NIM : 19416255201102**

Tugas akhir ini telah diterima dan disahkan untuk memenuhi

sebagian syarat memperoleh gelar sarjana

pada Program Studi Teknik Informatika

Fakultas Ilmu Komputer

Universitas Buana Perjuangan Karawang

Karawang, 22 Agustus 2023

Ketua Penguji,

Anggota Penguji I,

Anggota Penguji II,

**Sutan Faisal, M.Kom**

NIDN: 0428047401

**Tatang Rohana, M.Kom**

NIDN: 0412047201

**Dr. Hanny Hikmayanti.H, M.Kom**

NIDN: 0427037305

Mengetahui:

Dekan Fakultas Ilmu Komputer,

Koordinator Program Studi,

**Dr. Ahmad Fauzi, M.Kom**

NIDN: 0419037701

**Jamaludin Indra, M.Kom**

NIDN: 0405058208

## LEMBAR PERNYATAAN

Saya **Ricky Steven Chandra** menyatakan dengan sesungguhnya bahwa Proposal Tugas Akhir yang saya tulis dengan judul **Deteksi Telur Bebek *Fertil* dan *Infertil* Menggunakan Algoritma *Convolutional Neural Network* (CNN), *Detection Of Fertil And Infertil Duck Egg Using Convolutional Neural Network (CNN) Algorithm***. Beserta dengan seluruh isinya adalah merupakan hasil karya sendiri. Saya tidak melakukan penjiplakan yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan.

Sesuai peraturan yang berlaku saya siap menanggung resiko/sanksi yang diberikan jika dikemudian hari ditemukan pelanggaran terhadap etika keilmuan dalam Proposal Tugas Akhir ini atau jika ada klaim dari pihak lain terhadap keaslian karya.

Karawang, 29 Juli 2023  
Yang Menyatakan,

Ricky Steven Chandra

## KATA PENGANTAR

Puji dan syukur penulis ucapkan kehadiran Tuhan Yang Maha Esa karena yang telah memberikan Rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan Tugas Akhir ini dengan judul DETEKSI TELUR BEBEK FERTIL DAN INFERTIL MENGGUNAKAN ALGORITMA CONVOLUTIONAL NEURAL NETWORK (CNN). Penulis menyadari dalam menyusun Tugas Akhir ini banyak mendapat dukungan, bimbingan bantuan dan kemudahan dari berbagai pihak sehingga tugas akhir ini dapat diselesaikan. Dengan ketulusan hati, penulis menyampaikan ucapan terima kasih kepada:

Penulis mengucapkan terima kasih kepada:

1. Prof. H. Dedi Mulyadi SE, MM, Rektor Universitas Buana Perjuangan Karawang.
2. Dr. Ahmad Fauzi, M. Kom, Dekan Fakultas Ilmu Komputer Universitas Buana Perjuangan Karawang.
3. Jamaludin Indra, M. Kom, Koordinator Program Studi Teknik Informatika Universitas Buana Perjuangan Karawang, yang menerima penulis dengan baik untuk berkonsultasi.
4. Tatang Rohana, M. Kom, Koordinator Tugas Akhir Program Studi Teknik Informatika Universitas Buana Perjuangan Karawang yang nerima penulisan dengan baik untuk berkonsultasi.
5. Dr. Hanny Hikmayanti, M. Kom, Pembimbing I yang telah memberikan bimbingan pembuatan tugas akhir.
6. Yana Cahyana, M.Kom, Pembimbing II yang telah memberikan bimbingan tata cara menulis karya ilmiah dengan benar.
7. Kedua Orang Tua yang telah memberikan otivasi untuk terus semangat.
8. Dimas Maulana yang telah menjadi, motivasi, dan mentor selama mengerjakan Tugas Akhir.
9. Daffa Nazmi Alwan yang telah menjadi teman diskusi selama mengerjakan Tugas Akhir.
10. Riyandi Aditya Fitrah yang telah menjadi teman diskusi selama mengerjakan Tugas Akhir.

11. Yusuf Sopian Maulana yang telah menjadi teman diskusi selama mengerjakan Tugas Akhir.

Semoga Tugas Akhir ini dapat bermanfaat, baik sebagai sumber informasi maupun sumber inspirasi bagi para pembaca.

Karawang, 29 Juli 2023

Penulis,

Ricky Steven Chandra

## **ABSTRAK**

Dalam penetasan telur bebek, telur yang infertil perlu disortir dari mesin tetas agar tidak membusuk di mesin tetas. Proses penyortiran dilakukan dengan meneropong telur menggunakan senter atau lampu yang diletakkan dibalik telur. Tujuan dari penelitian ini adalah untuk mengembangkan deteksi telur bebek fertil dan infertil menggunakan algoritma Convolutional Neural Network (CNN), untuk menggantikan peneropongan secara manual karena tingkat kelelahan manusia akan rentan terjadinya kesalahan pada penyortiran telur bebek fertil dan infertil. Model yang digunakan adalah You Look Only Once (YOLO) merupakan salah satu model deep learning yang dapat digunakan untuk pengenalan objek. Penelitian ini bertujuan untuk pengenalan objek pada citra telur bebek menggunakan YOLO. Penelitian ini terdiri dari beberapa tahap yaitu pengumpulan data, pra-proses data, konfigurasi jaringan YOLO, pelatihan model yolo dan pengujian. Jumlah data citra yang digunakan dalam penelitian yaitu 800 terdiri dari dua jenis telur bebek fertil dan infertil. Hasil pengujian telur bebek fertil dan infertil sebanyak 40 kali dengan menggunakan Algoritma Convolutional Neural Network (CNN), mencapai akurasi sekitar 95% dalam membedakan kedua kategori telur. Hasil ini menunjukkan bahwa cnn memiliki potensi yang signifikan dalam membedakan telur bebek fertil dari infertil, memberikan solusi yang cepat dan efisien bagi peternak bebek.

**Kata Kunci:** CNN, Fertil dan Infertil, Model.

## **ABSTRACT**

*In hatching duck eggs, infertile eggs need to be sorted from the incubator so they don't rot in the incubator. The sorting process is done by observing the eggs using a flashlight or lamp placed behind the eggs. The purpose of this research is to develop the detection of fertile and infertile duck eggs using the Convolutional Neural Network (CNN) algorithm, to replace manual surveillance because the level of human fatigue will be prone to errors in sorting fertile and infertile duck eggs. The model used is You Look Only Once (YOLO) which is a deep learning model that can be used for object recognition. This study aims to recognize objects in duck egg images using YOLO. This research consisted of several stages, namely data collection, data pre-processing, YOLO network configuration, Yolo model training and testing. The amount of image data used in the study was 800 consisting of two types of fertile and infertile duck eggs. The results of testing fertile and infertile duck eggs 40 times using the Convolutional Neural Network (CNN) Algorithm, achieved an accuracy of around 95% in distinguishing the two categories of eggs. These results indicate that CNN has significant potential in differentiating fertile from infertile duck eggs, providing a quick and efficient solution for duck farmers.*

**Keyword:** CNN, Fertile and Barren, Model.



## DAFTAR ISI

LEMBAR PERSETUJUAN.....	i
LEMBAR PENGESAHAN .....	ii
LEMBAR PERNYATAAN .....	iii
ABSTRAK .....	vi
DAFTAR TABEL .....	x
DAFTAR GAMBAR .....	xi
DAFTAR LAMPIRAN .....	xii
BAB I PENDAHULUAN .....	1
1.1. Latar Belakang .....	1
1.2. Rumusan Masalah .....	2
1.3. Tujuan Penelitian.....	2
1.4. Manfaat Penelitian.....	2
BAB II TINJAUAN PUSTAKA.....	3
2.1. Telur Bebek <i>Fertil</i> dan <i>Infertil</i> .....	3
2.2. <i>Computer vision</i> .....	3
2.3. <i>Python</i> .....	4
2.4. <i>Convolutional Neural Network (CNN)</i> .....	4
2.5. <i>Deep Learning</i> .....	5
2.6. <i>OpenCV</i> .....	5
2.7. <i>Google Colab</i> .....	6
2.8. <i>Framework YOLOv8</i> .....	6
2.9. <i>Webcam</i> .....	7
2.10. <i>Visual Studio Code</i> .....	8
2.11. <i>Makesense.ai</i> .....	8
2.12. Penelitian Terkait .....	9
BAB III METODE PENELITIAN.....	11
3.1. Objek Penelitian .....	11
3.2. Bahan Penelitian.....	12
3.3. Prosedur Penelitian.....	13
BAB IV HASIL PEMBAHASAN .....	15
4.1. Perancangan Model.....	15

4.1.1. Pengumpulan Dataset.....	15
4.1.2. Preprocessing Citra .....	16
4.1.3. Membuat File Coco.....	17
4.1.4. Training Model.....	18
4.1.5. Implementasi Model.....	20
4.1.6. Hasil Pengujian .....	20
<b>BAB V KESIMPULAN DAN SARAN .....</b>	<b>24</b>
5.1. Kesimpulan .....	24
5.2. Saran.....	24
<b>DAFTAR PUSTAKA .....</b>	<b>25</b>
<b>LAMPIRAN.....</b>	<b>27</b>
<b>RIWAYAT PENULIS.....</b>	<b>51</b>

## DAFTAR TABEL

Tabel 2. 1 Penelitian Terkait .....	9
Tabel 3. 1 Objek Penelitian.....	11
Tabel 4. 1 Pengujian.....	20

## DAFTAR GAMBAR

Gambar 2. 1 Python .....	4
Gambar 2. 2 OpenCV.....	4
Gambar 2. 3 Colab .....	6
Gambar 2. 4 WebcamV .....	6
Gambar 2. 5 Visual Studio Code.....	6
Gambar 2. 6 Colab .....	6
Gambar 2. 8 Webcam.....	7
Gambar 2. 9 Webcam.....	7
Gambar 2. 10 Visual Studio Code .....	8
Gambar 2. 11 Visual Studio Code .....	8
Gambar 2. 12 Makesense.ai .....	8
Gambar 3. 1 Prosedur Penelitian.....	13
Gambar 3. 2 Kamera Dslr Canon.....	13
Gambar 3. 3 Senter.....	13
Gambar 4. 1 Struktur Folder Images.....	15
Gambar 4. 2 Train,Val,Test.....	15
Gambar 4. 3 Labeling Citra.....	16
Gambar 4. 4 Proses Labeling Citra .....	16
Gambar 4. 5 Hasil ZIP dan Labeling .....	17
Gambar 4. 6 File Coco .....	17
Gambar 4. 7 Training Model.....	18
Gambar 4. 8 Nilai Recall.....	18
Gambar 4. 9 Nilai Kurva F1 dan Precision Terhadap Nilai Confidence .....	18
Gambar 4. 10 Confusion Matrix .....	19
Gambar 4. 11 Hasil Implementasi Model .....	20

## DAFTAR LAMPIRAN

Lampiran 1 Lembar Persetujuan .....	27
Lampiran 2 Lembar Pengesahan .....	27
Lampiran 3 Lembar Pengesahan .....	28
Lampiran 4 Lembar Perbaikan Ketua Penguji .....	29
Lampiran 5 Lembar Perbaikan Anggota penguji 1 .....	30
Lampiran 6 Lembar Perbaikan Anggota Penguji 2 .....	30
Lampiran 7 Lembar Perbaikan Anggota Penguji 2 .....	30
Lampiran 8 Lembar Perbaikan Anggota Penguji 2 .....	31
Lampiran 9 Dokumentasi Wawancara .....	31
Lampiran 10 Dokumentasi Wawancara .....	32
Lampiran 11 Validasi Peternak .....	33
Lampiran 12 Hasil Runing Google Colab .....	36
Lampiran 13 Codingan Yolov8 .....	50



# **BAB I**

## **PENDAHULUAN**

### **1.1. Latar Belakang**

Telur merupakan sumber nutrisi yang sangat baik. Jika dibandingkan dengan nilai gizi telur yang lain seperti ayam, bebek dan puyuh, telur bebek memiliki nilai gizi yang paling unggul. Telur bebek banyak dimanfaatkan dalam berbagai minuman seperti susutelur madu jahe, teh talua (Novra & Ariani, 2020). Kandungan gizi dalam telur sangat lengkap dan baik untuk dikonsumsi setiap hari. Selain harganya yang terjangkau, telur juga mudah didapatkan dan kaya akan protein serta omega-3 (Pencegahan et al., 2021). Meskipun permintaan telur bebek di pasaran tinggi, produksi telur sering menghadapi hambatan karena jumlah bebek petelur. Jika jumlah bebek sedikit, maka produksi telur juga akan sedikit. Sebaliknya, jika bebek petelur banyak, maka produksi telur juga akan tinggi. Untuk meningkatkan produksi telur, sering dilakukan upaya penetasan telur bebek yang berkualitas. Telur dapat dibedakan menjadi fertil dan infertil berdasarkan adanya pembuluh darah dan titik embrio pada telur (Pandy Aldrige Simanungkalit1, 2021). Proses produksi penetasan telur bebek menjadi itik memerlukan strategi seleksi telur terlebih dahulu untuk memastikan telur yang dipilih dapat menetas. Pendeteksian embrio telur saat ini masih banyak dilakukan secara manual, yaitu dengan cara menyinari telur dalam ruangan gelap dan meneropong isi telur menggunakan senter. Metode ini sangat bergantung pada tenaga manusia dan rawan terjadi kesalahan karena kondisi mata dan tingkat kelelahan manusia. Oleh karena itu, penting untuk melakukan penelitian lebih lanjut guna meningkatkan efisiensi dan akurasi dalam proses meneropong telur bebek *fertil* dan *infertil*. Setelah terpilih, telur kemudian ditetaskan dengan proses yang memakan waktu 28 hari dengan suhu pemanas (26-28°C) (Daryatmo et al., 2020).

Menurut (Dewi dkk, 2019) Kegagalan akibat telur itik menetas terlalu dini karena suhu kelembapan di hari pertama hingga hari ke-19 terlalu tinggi. Selain itu, penetasan yang terlalu dini juga bisa disebabkan oleh ukuran telur terlalu kecil. Perkembangan teknologi yang cepat memfasilitasi berbagai aspek kehidupan, termasuk dalam proses pendeteksian telur bebek yang *fertil* dan *infertil*. Tantangan

dan hambatan yang dihadapi bisa diatasi dengan menggunakan teknologi yang ada saat ini. Berdasarkan masalah tersebut, peneliti membangun deteksi telur bebek *fertil* dan *infertil* dengan menggunakan algoritma *Convolutional Neural Network* (*CNN*) melalui proses latihan (*Training*) dan pengujian (*Testing*) model. Dalam penelitian ini, peneliti membangun program deteksi telur bebek yang *fertil* dan *infertil* dengan menggunakan algoritma *Convolutional Neural Network* (*CNN*), dengan menggunakan 900 sampel data yang terdiri dari 450 sampel telur bebek yang *fertil* dan 450 sampel telur bebek yang *infertil*.

### 1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan di atas, makadapat dirumuskan masalah pokok pada penelitian ini yaitu :

1. Bagaimana mendeteksi telur bebek *fertil* dan *infertil* melalui citra menggunakan algoritma *Convolutional Neural Network* (*CNN*) ?
2. Bagaimana cara kerja algoritma *Convolutional Neural Network* (*CNN*) dalam mendeteksi nilai akurasi telur *fertil* dan *infertil* ?

### 1.3. Tujuan Penelitian

Adapun tujuan pada penelitian ini yaitu ?

1. Melakukan deteksi objek pada telur bebek *fertil* dan *infertil* menggunakan algoritma *Convolutional Neural Network* (*CNN*)
2. Mengetahui kinerja algoritma *Convolutional Neural Network* (*CNN*) dalam mendeteksi telur bebek *fertil* dan *infertil*.

### 1.4. Manfaat Penelitian

Manfaat yang diharapkan dalam penelitian ini adalah sebagai berikut :

1. Mempermudah para peternak bebek petelur dalam mendeteksi telur bebek *fertil* dan *infertil*.
2. Membantu peneliti dalam mendeteksi telur bebek *fertil* dan *infertil* menggunakan algoritma *Convolutional Neural Network* (*CNN*) .



## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1. Telur Bebek *Fertil* dan *Infertil***

Telur bebek, yang diproduksi oleh bebek, secara fisik lebih kecil dari telur ayam dan memiliki cangkang yang lebih keras serta bagian kuning telur yang lebih besar. Telur bebek memiliki beragam aplikasi, seperti menjadi bahan dasar makanan, komponen dalam industri kosmetik, atau bahan dalam pembuatan lukisan. Telur bebek juga dikenal kaya akan nutrisi penting, termasuk protein, vitamin, dan mineral, yang menjadikannya pilihan makanan yang sehat dan bergizi. Dengan rasa yang lezat, mudah dicerna, dan tinggi nutrisi, telur bebek adalah sumber protein hewani yang sangat baik. Umumnya, telur bebek memiliki ukuran besar dengan cangkang berwarna putih hingga hijau kebiruan (Agus Wantoro 2019).

Telur *fertil* adalah telur yang memiliki embrio dalam proses perkembangan, yang dapat dikenali dari titik atau noktah yang ada di kuning telur. Telur fertil dapat menetas dan biasanya digunakan sebagai bibit ayam, atau yang dikenal sebagai *DOC (Day Old Chick)*. Di sisi lain, telur infertil adalah telur yang tidak mengalami perkembangan embrio dan tidak dapat menetas. Telur fertil dan infertil dapat dibedakan berdasarkan adanya atau tidaknya pembuluh darah dan titik embrio di dalam telur (Pandy Aldrige Simanungkalit1, 2021).

#### **2.2. *Computer vision***

*Computer vision* adalah cabang ilmu dalam bidang teknologi informasi yang memungkinkan komputer untuk 'melihat' dan menganalisis objek atau gambar di sekitarnya. Dengan kemampuan ini, komputer dapat memahami dan memproses informasi dari gambar yang dihadapinya, yang kemudian dapat digunakan untuk menghasilkan perintah atau aksi tertentu. Dengan kata lain, *computer vision* memberikan kemampuan pada mesin atau komputer untuk mengenali dan memahami citra, bahkan mungkin melebihi kemampuan penglihatan manusia dalam beberapa aspek tertentu. *Computer Vision* ini secara tidak langsung

merupakan kemampuan dari sebuah mesin atau computer dalam melihat atau mengenali sebuah citra dengansama atau bahkan dapat melebihi.

### 2.3. *Python*

*Python* adalah bahasa pemrograman berorientasi objek dan dinamis yang dapat digunakan untuk beragam jenis pengembangan *software*. Keunggulan *Python* terletak pada sintaksis kodenya yang sangat jelas dan mudah dibaca, ditambah dengan pustaka standar yang lengkap. Bahasa ini didesain dengan tujuan utama untuk mempermudah pembacaan kode sumber.

Sebagai bahasa *pemrograman interpretatif*, *Python* bersifat *fleksibel* dan dirancang dengan filosofi yang menekankan pada keterbacaan kode (Syahrudin and Kurniawan, 2018). *Python* mendukung multi-paradigma pemrograman, yang memungkinkan *programmer* memilih gaya pemrograman yang paling sesuai dengan kebutuhan mereka. Selain itu, *Python* adalah bahasa pemrograman dinamis dan memiliki fitur manajemen memori *otomatis*, yang semakin memudahkan penggunaannya.



Gambar 2. 1 Python  
(Sumber : <https://bit.ly/3q54Wt2>)

### 2.4. *Convolutional Neural Network (CNN)*

*Convolutional Neural Network (CNN)* merupakan salah satu jenis dari *Deep Neural Network* yang dikembangkan dari *Multilayer Perceptron (MLP)* dengan tujuan khusus untuk mengolah data berbentuk dua dimensi. *CNN* sangat umum digunakan pada pengolahan data citra karena kedalaman jaringannya yang tinggi (Yohannes, 2020).

Struktur *CNN* biasanya terdiri dari lapisan-lapisan yang disusun dalam tiga dimensi: lebar, tinggi, dan kedalaman. Ukuran dari lapisan ini ditentukan oleh lebar dan tinggi objek, sedangkan jumlah lapisan ditunjukkan oleh kedalaman objek.

Umumnya, *CNN* terbagi menjadi dua bagian utama, yaitu:

1. Layer Ekstraksi: Berada di awal arsitektur, lapisan ini terdiri dari beberapa layer. Setiap layer terhubung dengan region lokal dari layer sebelumnya.
2. Layer Klasifikasi: Layer ini terdiri dari beberapa layer lainnya. Setiap layer di sini memiliki neuron yang saling terhubung secara penuh dengan layer lainnya.

## **2.5. Deep Learning**

*Deep Learning* adalah suatu cabang dari *Machine Learning* dan *Artificial Intelligence* yang menggunakan jaringan syaraf tiruan untuk mempelajari dan memahami pola dari data. Konsep ini berfokus pada proses pembelajaran melalui lapisan tersembunyi (*hidden layers*) yang menerima input metadata dan memprosesnya. Salah satu keunggulan algoritma *deep learning* adalah kemampuannya untuk mengurangi beban pemrograman dengan cara otomatis memilih fitur-fitur yang relevan. Setiap lapisan tersembunyi dalam jaringan dilatih untuk memahami fitur berdasarkan output dari lapisan sebelumnya. Dengan ini, *deep learning* mampu menangani dan memecahkan masalah yang kompleks dan rumit yang melibatkan berbagai transformasi non-linear (Helsaputra et al., 2021).

## **2.6. OpenCV**

*OpenCV* adalah pustaka perangkat lunak yang menyediakan berbagai fungsi pemrograman untuk visi komputer secara *real-time*. Sebagai perangkat open-source, *OpenCV* bisa digunakan untuk mendukung proyek-proyek akademis maupun komersil. *OpenCV* menawarkan antarmuka pemrograman untuk C, C++, *Python*, dan *Java*, bisa dijalankan pada berbagai sistem operasi seperti *Windows*, *Mac*, *Linux*, dan *Android* (Budiarjo 2020). Dengan lebih dari 2500 algoritma yang telah dioptimalkan, *OpenCV* merupakan alat yang sangat efisien dan canggih. Pada Gambar 2.2 merupakan gambar logo *opencv*.



Gambar 2. 3 Colab

(Sumber : <https://bit.ly/3C22iam>)

## 2.7. Google Colab

*Google Colab* adalah layanan berbasis *cloud* yang disediakan oleh *Google*, yang berfungsi mirip seperti *Jupyter Notebook*. *Google Colab* dapat digunakan untuk menjalankan kode *Python* secara interaktif langsung dari *browser*, seperti *Opera*, *Mozilla Firefox*, dan *Google Chrome*, tanpa perlu melakukan instalasi di desktop. Selain itu, *Google Colab* juga menyediakan lingkungan komputasi dengan akses ke *GPU*, yang sangat berguna untuk komputasi yang membutuhkan sumber daya tinggi seperti *deep learning*. *Google Colab* adalah sebuah *IDE* untuk pemrograman *Python* dimana pemrosesan akan dilakukan oleh server *Google* yang memiliki perangkat keras dengan performa yang tinggi (Rangga Gelar Guntara, 2023).

*Google Colab* produk dari *Google Research*. Colab adalah *executable document*, yang biasa digunakan untuk menulis, menyimpan, serta membagikan program yang telah ditulis melalui *Google Drive*. *Google Colab* bisa dikatakan sebagai *Notebook* yang disimpan pada *Google Drive*.



Gambar 2. 6 Colab

(Sumber : <https://bit.ly/45NSAWX>)

## 2.8. Framework YOLOv8

*YOLO (You Only Look Once)* adalah kerangka kerja yang diarahkan untuk mendeteksi objek dalam *real-time*. Model ini diterapkan ke gambar pada berbagai lokasi dan skala. Daerah gambar dengan skor paling tinggi dianggap sebagai hasil

deteksi (Kurniasari dan Sugiono 2021). *YOLO* menjadikan proses pendeteksian objek sebagai satu masalah regresi, yang memproses langsung dari piksel gambar sampai koordinat kotak pembatas dan probabilitas kelas. Dengan *YOLO*, sistem hanya perlu memandang gambar sekali (*You Only Look Once*) untuk memprediksi objek apa yang ada dan di mana lokasinya.

*YOLOv8* adalah model *YOLO* yang paling maju dan terkini, yang dapat digunakan untuk deteksi objek, klasifikasi gambar, dan tugas segmentasi instan. *YOLOv8* dikembangkan oleh Ultralytics, yang juga menciptakan model *YOLOv5* yang berpengaruh dan menentukan standar industri. *YOLOv8* menghadirkan banyak perubahan dan peningkatan dalam hal arsitektur dan pengalaman pengembangan dibandingkan dengan *YOLOv5*.

## 2.9. Webcam

*Webcam* (*Web Camera*) adalah alat yang memungkinkan gambar atau video dapat ditampilkan secara *real-time* melalui *World Wide Web*, aplikasi pesan instan, atau aplikasi panggilan video. Hingga saat ini, penggunaan webcam lebih banyak difokuskan pada perekaman dan penayangan objek, namun belum optimal dalam aplikasi lain seperti sistem keamanan yang berfungsi untuk mendeteksi pergerakan objek. *Webcam* belum sepenuhnya dapat memberikan informasi tentang apakah suatu objek bergerak atau tetap diam. Oleh karenanya, penelitian untuk pengembangan sistem yang dapat mendeteksi gerakan dari gambar yang diambil oleh webcam menjadi penting. Sistem deteksi gerak akan beroperasi jika terdapat perubahan posisi objek, yang secara otomatis akan mengaktifkan alarm. Metode deteksi tepi digunakan dalam pengolahan citra untuk deteksi gerakan, sementara untuk proses deteksi gerakannya sendiri, digunakan proses perbandingan posisi piksel (Julfikar Ali Andre, 2020). Pada Gambar 2.4 merupakan gambar logo *webcam*.



Gambar 2. 8 Webcam

(Sumber: <https://bit.ly/3MGlgII>)

### 2.10. Visual Studio Code

Visual Studio Code adalah editor kode yang dikembangkan oleh Microsoft dan dapat digunakan di platform Windows, Linux, dan MacOS. Editor ini menyediakan berbagai fitur bawaan seperti debugging, kontrol Git yang terintegrasi dengan GitHub, penyorotan sintaksis, penyelesaian kode otomatis, snippet, dan refactoring kode. Visual Studio Code juga sangat dapat disesuaikan dengan preferensi pengguna, termasuk tema, pintasan keyboard, dan ekstensi tambahan yang dapat menambahkan fungsionalitas lebih lanjut.

Visual Studio Code adalah open source dan tersedia di bawah Lisensi MIT yang sangat fleksibel. Binari yang telah dikompilasi dapat digunakan secara gratis untuk keperluan pribadi maupun komersial (R. Kurniawan, 2019). Gambar 2.5 menunjukkan logo dari Visual Studio Code.



Gambar 2. 10 Visual Studio Code

(Sumber : <https://bit.ly/3N1qxM0>)

### 2.11. Makesense.ai

*Makesense.ai* adalah sebuah aplikasi berbasis website yang digunakan untuk berbagai keperluan mengenai AI seperti computer vision yaitu proses membuat label suatu objek. Untuk melakukan deteksi objek tentu-nya membutuhkan sebuah parameter yang digunakan untuk menjadi kunci dari deteksi objek tersebut, parameter yang dimaksud bisa berupa warna, bentuk, tindakan dan lain-lainnya. (Iskandar Mulyana, 2022). Pada Gambar 2.6 merupakan logo *makesense.ai*.



Gambar 2. 12 Makesense.ai

(Sumber : <https://bit.ly/3N176Tv>)

## 2.12. Penelitian Terkait

Review paper dalam Tabel Penelitian 2.1 berfungsi sebagai pondasi dalam menyusun penelitian, dan juga untuk mengetahui hasil dari penerapan metode yang digunakan dalam penelitian. Penelitian terkait dapat menjadi perbandingan dan gambaran untuk mendukung penelitian selanjutnya dengan menggunakan metode yang sama.

Tabel 2. 1 Penelitian Terkait

No	Judul Penelitian	Metode	Hasil
1	Real-time Deteksi Masker Berbasis Deep Learning menggunakan Algoritma CNN YOLOv3 (Lusiana Agustien, Taufikur Rahman, Ahmad Walid Hujairi) 2021	CNN	Objek yang teridentifikasi sebagai tidak memakai masker menunjukkan tingkat kepercayaan atau nilai confidence sebesar 0.53, hal ini didapatkan dari gambar objek dengan kualitas ketajaman yang relatif rendah dan berada pada jarak sekitar 4.5 meter dari kamera. Rentang nilai confidence untuk hasil deteksi dalam ruangan berkisar antara 0.53 hingga 0.98.
2	Deteksi Jenis Beras Menggunakan Algoritma YOLOv3 (Syamsul Ma'arif, Tatang Rohana, Kiki Ahmad Baihaqi)	YOLOv3	Pelatihan yang dilakukan selama 33 jam menghasilkan rata-rata kehilangan (avg loss) sebesar 0,0500 setelah 6000 iterasi batch. Rata-rata loss yang lebih rendah menunjukkan hasil deteksi yang lebih akurat. Jumlah iterasi memiliki pengaruh besar terhadap rata-rata loss, sehingga menentukan jumlah iterasi sangat penting dalam proses deteksi. Dalam pengujian, melakukan 12 kali pendeteksian pada objek citra digital dengan posisi citra beras berurutan atau tidak bertumpuk, menghasilkan nilai akurasi sebesar 100%. Namun, jika citra beras ditumpuk, akurasi yang diperoleh hanya 60%.
3	Sistem klasifikasi telur ayam fertil dan infertil menggunakan fitur tekstur dan metode K-Nearest Neighbor	KNN	Berdasarkan hasil analisis dan pengujian dengan metode K-Fold Cross Validation terhadap fitur GLCM, diketahui bahwa kombinasi fitur terbaik adalah dissimilarity-correlation. Sementara itu,

	Berbasis Raspberry (Pandy Aldrige Simanungkalit <sup>1</sup> , Hurriyatul Fitriyah <sup>2</sup> , Eko Setiawan <sup>3</sup> ) 2021	(K-Nearest Neighbor)	klasifikasi menggunakan metode <i>K-Nearest Neighbor</i> menghasilkan tingkat akurasi sebesar 93,33% ketika jumlah tetangga (K) adalah 7 dan 9.
4	Pengenalan Pola Aksara Sunda dengan Metode Convolutional Neural Network (Alif Kirana, Hanny Hikmayanti H, Jamaludin Indra ) 2020	CNN	Berdasarkan hasil penelitian yang melibatkan empat jenis pengujian, didapatkan bahwa akurasi pengujian gambar yang diambil dari e-book mencapai 72,4%. Untuk gambar yang diambil dari font komputer, akurasi yang diperoleh mencapai 100,0%. Selanjutnya, gambar tulisan tangan yang diambil menggunakan kamera smartphone memiliki akurasi sebesar 84,4%, sementara gambar tulisan tangan yang diambil dengan scanner mencapai akurasi sebesar 85,5%. Jadi, berdasarkan hasil pengujian ini, metode <i>CNN</i> terbukti dapat mengklasifikasikan dan mengenali citra tulisan tangan dalam bahasa Sunda dengan cukup baik.
5	Penerapan Algoritma Convolutional Neural Network dalam Klasifikasi Telur Ayam Fertil dan Infertil Berdasarkan Hasil Candling (Muhammad Rizky Firdaus ) 2020	CNN	<i>Data</i> yang telah dikumpulkan dari telur dibagi menjadi dua kategori, yakni telur <i>fertil</i> dan <i>infertil</i> . <i>Data</i> telur <i>fertil</i> terdiri dari 111 sampel, sementara data telur <i>infertil</i> mencakup 138 sampel. Data ini kemudian dibagi lagi menjadi dua bagian: data latihan ( <i>training</i> ) dan data validasi.





## BAB III METODE PENELITIAN

### 3.1. Objek Penelitian

Objek penelitian yang digunakan yaitu telur bebek *fertil* dan *infertil* dalam membaca sebuah objek citra pada bagian deteksi telur menggunakan algoritma dengan menggunakan algoritma *Convolutional Neural Network (CNN)*, terdapat 1 klasifikasi objek yang akan diteliti yaitu telur bebek *fertil* dan *infertil*, citra tersebut akan ditraining menggunakan *google colab* yang sudah melewati proses *labeling*.

Tabel 3. 1 Objek Penelitian

No	Nama	Gambar	Penjelasan
1.	Fertil		Telur fertil adalah jenis telur yang memiliki potensi untuk menetas. Dengan kata lain telur tersebut telah dibuahi dan memiliki embrio yang sedang berkembang di dalamnya.
2.	Infertil		Telur infertil adalah telur yang tidak dapat menetas karena tidak adanya perkembangan embrio di dalamnya selama proses penetasan.

### 3.2. Bahan Penelitian

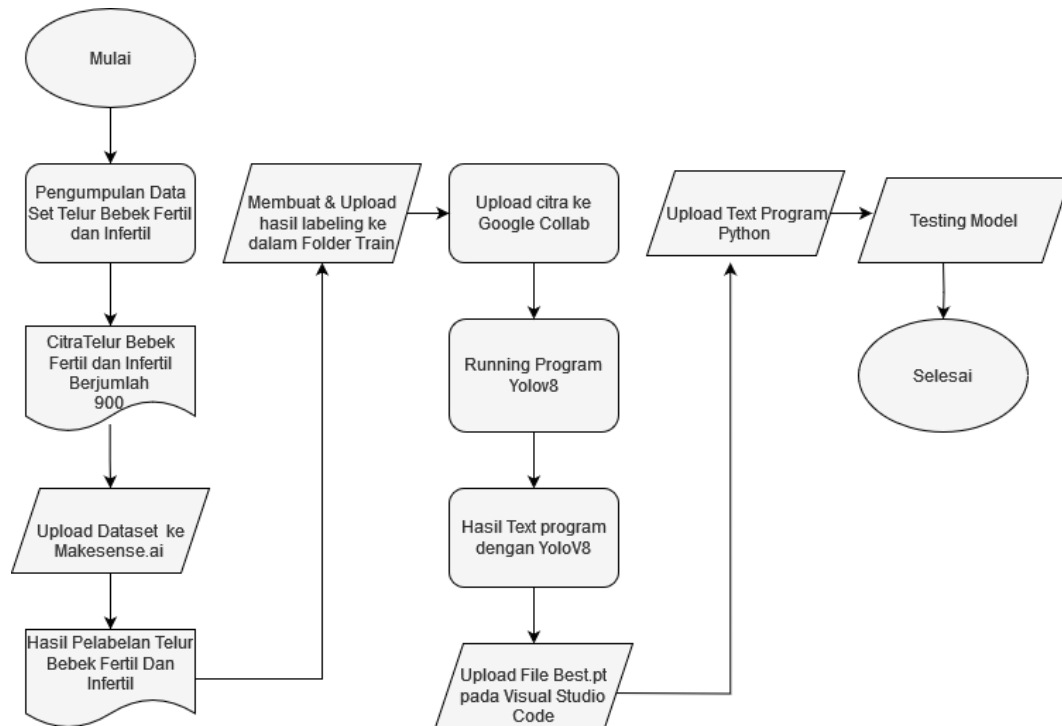
Bahan penelitian yang digunakan berupa perangkat lunak dan juga perangkat keras, serta untuk pengumpulan *dataset* yang diambil menggunakan alat bantu seperti kamera *Dslr Canon Eos 1200D* dan senter. *Dataset* yang dikumpulkan berjumlah 900 *dataset* telur bebek terdiri dari, telur bebek *fertil* berjumlah 450 dan telur bebek *infertil* berjumlah 450 *dataset* yang akan di *training* pada *google colab*.

Perangkat lunak yang digunakan pada penelitian sebagai berikut :

- *Google collab* merupakan *framework* yang digunakan untuk menulis program menggunakan bahasa *python* beserta melakukan eksekusi program, sehingga nantinya program tersebut dapat tersimpan di drive dan mudah dibagikan.
- *Visual Studio Code* merupakan alat editor kode yang dikembangkan oleh *Microsoft* dan dapat digunakan secara gratis di semua perangkat *desktop*. Dengan fitur dan ekstensi yang lengkap, alat editor kode ini menjadi pilihan favorit para pengembang. *Visual Studio Code* mendukung hampir semua sistem operasi, termasuk *Windows*, *Mac OS*, dan *Linux*. *Requirements Visual Studio Code 2019* :
  - *Windows* 10 versi 1703 atau lebih tinggi: Home, Professional, Education, dan Enterprise
  - 1,8 GHz
  - RAM 2 GB – RAM 8 GB
  - 64 Bit
  - *Hard Disk* 800MB – 210GB
- *Makesense.ai* adalah sebuah platform berbasis *web* yang digunakan untuk berbagai keperluan yang berhubungan dengan *AI*, khususnya dalam bidang *computer vision*. Salah satu fungsionalitas utamanya adalah pelabelan objek, yang memudahkan proses pembuatan dan pengelolaan *dataset* untuk keperluan pelatihan model *AI*.

### 3.3. Prosedur Penelitian

Prosedur yang akan dilakukan dalam penelitian ini dapat digambarkan sebagai *flowchart* berikut:



Gambar 3. 1 Prosedur Penelitian

Berikut ini merupakan penjelasan *flowchart* prosedur penelitian:

#### 1. Pengumpulan Dataset

Langkah penelitian yaitu mengumpulkan dataset telur bebek *fertil* dan *infertil* sebanyak 900, proses pengambilan dataset di ambil menggunakan alat bantu kamera *DSLR Canon Eos 1200D* dan alat teropongtelur seperti gambar di bawah ini :



Gambar 3. 3 Senter



Gambar 3. 2 Kamera Dslr Canon

(Sumber : <https://bit.ly/43tL69w>)

(Sumber : <https://bit.ly/3IJg4T3>)

2. *Upload Dataset ke Dalam Makesense.ai*

Setelah dataset sudah terkumpul sebanyak 900 citra telur, langkah kedua penelitian memberi label objek dataset citra telur bebek *fertil* dan *infertil*, Website yang digunakan untuk pemberian label objek citra telur yaitu *makesense.ai*.

3. *Membuat & upload Hasil Labeling ke Dalam Folder Train*

Langkah ketiga penelitian yaitu memasukan dataset telur bebek *fertil* dan *infertil* sebanyak 900 citra yang sudah diberi label ke dalam *folder train*.

4. *Upload Citra ke Google Collab*

Langkah keempat penelitian yaitu meng-*upload folder train* telur bebek *fertil* dan *infertil* ke dalam *google collab*.

5. *Setting Program Dengan Yolov8*

Langkah kelima *setting program* dengan *yolov8* yaitu menentukan *bact size* dan *epochs*, serta menentukan *weights YOLO* yang di inginkan.

6. *Hasil Text Program Dengan Yolov8*

Langkah ke enam, setelah proses training berhasil dilakukan. Selanjutnya menyimpan hasil *file best.pt*. File tersebut berisi *dataset* yang telah ditraining pada *website google colab*.

7. *Upload File Best.pt Pada Visual Studio Code*

Langkah ketujuh, Memasukan file *Best.pt* yang sudah tersimpan kedalam Visual Studio Code.

8. *Upload Text Program Python*

Langkah kedelapan, masukan *text program python* untuk proses testing model.

9. *Implementasi Model*

Langkah terakhir, ini dilakukan setelah model program deteksi sudah terbuat. Implementasi model secara *Real-Time* menggunakan *Webcam*.

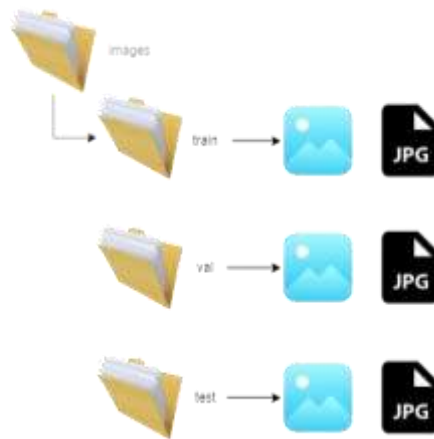
## BAB IV

### HASIL PEMBAHASAN

#### 4.1. Perancangan Model

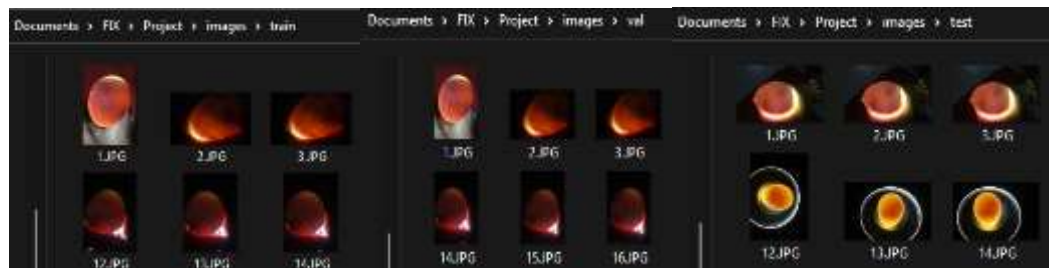
##### 4.1.1. Pengumpulan Dataset

Sebanyak 900 Citra *Dataset* yang berhasil didapatkan dan dikelompokkan kedalam folder *images*. Citra tersebut disimpan ke dalam folder yang diberinama *train*, *test*, dan *val* memiliki jumlah citra yang berbeda. Struktur folder *images* untuk dataset dapat ditunjukkan pada Gambar 4.1



Gambar 4. 1 Struktur Folder Images

Sebanyak 720 atau 80% dataset pada folder train telur bebek *fertil* dan *infertil* dan folder val sebanyak 144 atau 20% dari jumlah dataset. Sedangkan folder test sebanyak 14 atau 10%. Gambar 4.2. menunjukkan isi folder *train*, *val* dan *test*.

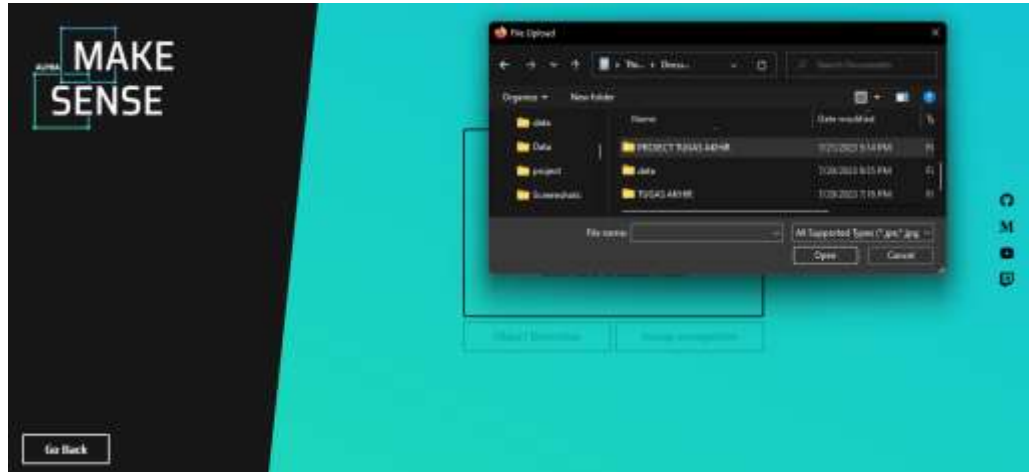


Gambar 4. 2 Train,Val,Test

Pada Gambar 4.2 isi folder *train*, *val*, dan *test* memiliki ciri penamaan yang khusus. Penamaan tersebut disertai dengan angka untuk kemudahan pengelompokan saat membaca berkas.

#### 4.1.2. Preprocessing Citra

*Labeling* Citra menggunakan website *Makesense.ai*. Tahap pelabelan citra untuk memberi *bounding box* terhadap citra yang sudah disusun. Pelabelan ini dilakukan agar model deteksi dapat belajar sesuai ciri telur bebek fertil dan infertil. Citra yang telah tersusun di dalam folder *train* akan dibuka direktorinya melalui *makesense.ai* seperti pada Gambar 4.3.



Gambar 4. 3 Labeling Citra

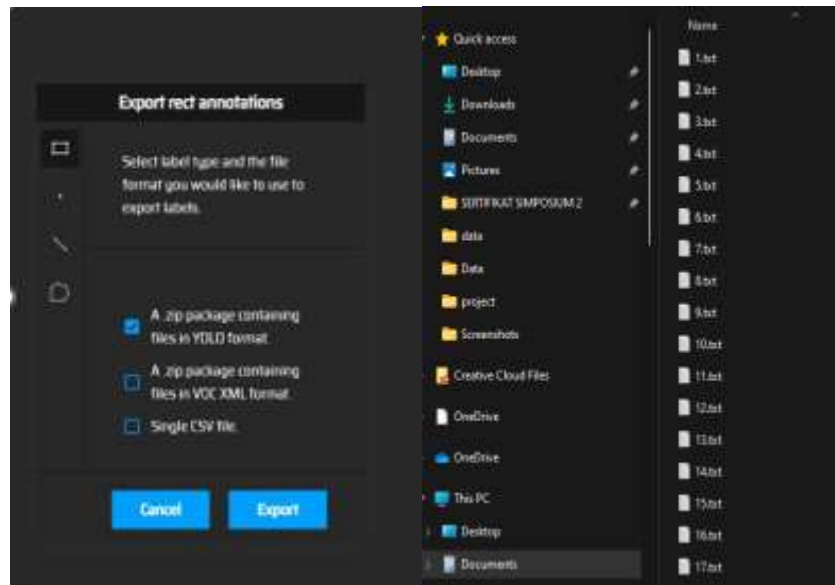
Proses *Labeling* citra dapat dilakukan dengan memberikan *bounding box* terhadap objek yang akan dideteksi. Proses pelabelan dan penamaan citra ditunjukkan pada Gambar 4.4.



Gambar 4. 4 Proses Labeling Citra

Pada Gambar 4.4. Proses *labeling* citra dilakukan ketika memberi bounding box berdasarkan citra yang terdeteksi. Saat proses *labeling* citra, perlu menentukan penamaan label. Penamaan label citra berguna agar citra dapat

dikenali seperti “Fertil”, “Infertil”. Format dari labeling citra menggunakan YOLO, dan hasil dari labeling tersebut berupa file ZIP dan di ekstrak ke dalam folder Train seperti pada Gambar 4.5.



Gambar 4. 5 Hasil ZIP dan Labeling

#### 4.1.3. Membuat File Coco

Mengelola *file Coco*, *File Coco* ini ber-isikan alamat dari folder *train*, *val* & *test* yang telah di simpan untuk diupload pada Google Colab. Format dari file ini adalah YAML. Struktur format pada “nc” berisi dengan nilai integer untuk klasifikasi objek yang terdeteksi, Sementara “names” diisi berdasarkan nama objek yang dideteksi. berisi 2 item dengan “nc” dan “names” yang berisi nama objek “Fertil”, “Infertil”. File Coco dengan format YAML ini digunakan saat proses training model pada Google Colab. Seperti ditunjukkan pada Gambar 4.6

```
1 train: /content/drive/MyDrive/Project/Images/train # train images (relative to 'path') 128 images
2 val: /content/drive/MyDrive/Project/Images/val # val images (relative to 'path') 128 images
3 test: /content/drive/MyDrive/Project/Images/test
4 #class
5 nc: 2
6
7 #classnames
8 names: ['Fertil', 'Infertil']
```

Gambar 4. 6 File Coco

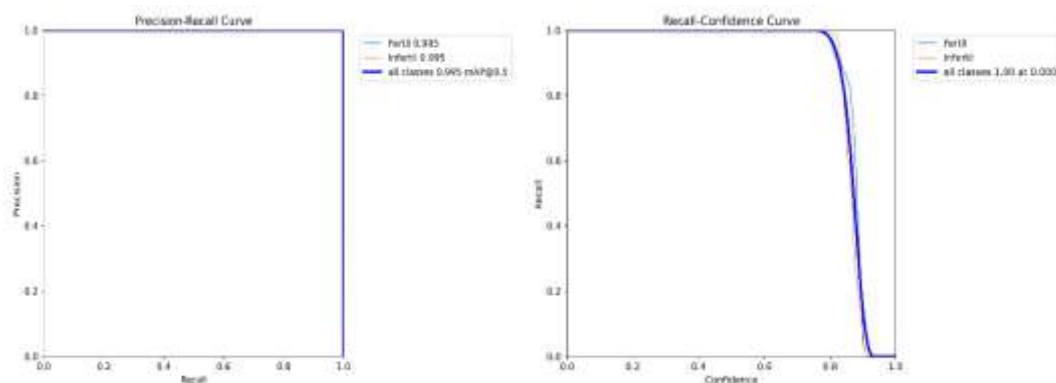
#### 4.1.4. Training Model

Proses ini memiliki tahapan penting dalam melatih model objek deteksi untuk memahami bentuk gambar agar dapat mengetahui objek di dalamnya. Proses training model ini bertujuan untuk mengenali ciri-ciri dari objek kelas yang di deteksi dengan melakukan penyesuaian *epochs*, *bact* dan *imgsize*. Berikut Gambar 4.7. Training Model.

Class	Images	Instances	Box(P	R	mAP50	mAP50-95)
all	144	144	0.999	1	0.995	0.874
Fertil	144	72	0.999	1	0.995	0.862
Infertil	144	72	0.999	1	0.995	0.887

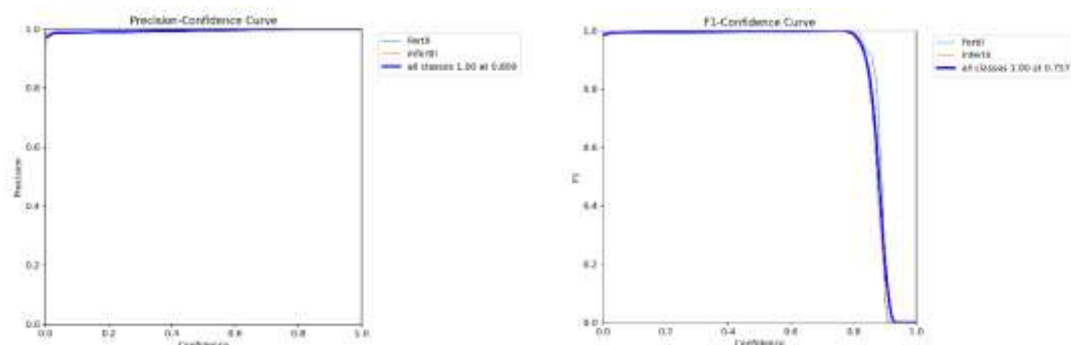
Gambar 4. 7 Training Model

Berdasarkan Gambar 4.7. telah dilakukan training model deteksi objek dengan jumlah *epochs* sebesar 200, sedangkan jumlah untuk *bact* sebesar 16 dan *imgsize* sebesar 640. Hasil yang didapat untuk mAP (Average Precision) sebesar 95%.



Gambar 4. 8 Nilai Recall

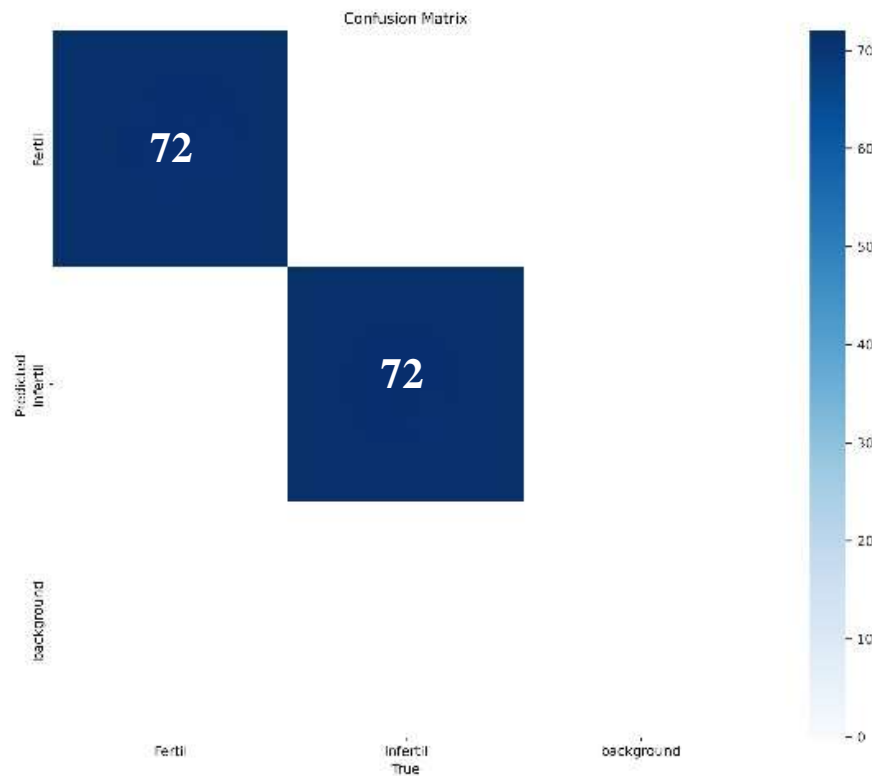
Pada Gambar 4.8. Deteksi telur bebek fertil dan infertil, hasil pelatihan menunjukkan nilai yang sangat baik. Nilai presisi mencapai rata-rata sebesar 0.995 terhadap nilai recall. Pada nilai kepercayaan 0.00, rata-rata nilai recall mencapai puncaknya, yaitu 1.00.



Gambar 4. 9 Nilai Kurva F1 dan Precision Terhadap Nilai Confidence



Pada Gambar 4.9. Rata-rata nilai Kurva F1 mencapai 1.00 dengan nilai confidence 0.709. sementara itu, rata-rata nilai precision mencapai 1.00 pada nilai confidence 0.892.



Gambar 4. 10 Confusion Matrix

Pada Gambar 4.10. merupakan gambar *Confusion Matrix*, Berikut perhitungan dari *confusion matrix* :

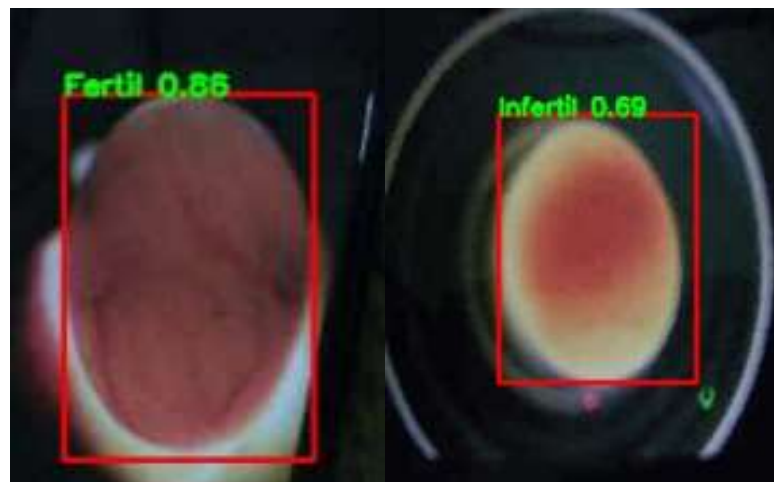
$$\begin{aligned}
 \text{Akurasi} &= \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \\
 &= \frac{72 + 72 + 0}{72 + 72 + 0 + 0} = \frac{144}{144} = 1 \\
 &= 1 \times 100\% \\
 &= 100\%
 \end{aligned}$$

Jadi  $72 + 72 + 0 = 144$  hasil dari 144 di bagi 144 = 1 lalu di kalikan 100 dan mendapatkan hasil 100% nilai tersebut di dapatkan dalam hasil perhitungan pada angka di dalam gambar *confusion matrix*. Pada penelitian deteksi telur

bebek *fertil* dan *infertil* ini mendapatkan nilai akurasi yang cukup tinggi yaitu sebesar 1 atau 100%. Deteksi telur bebek *fertil* dan *infertil* menggunakan YOLOv8 berjalan dengan lancar dan nilai akurasinya cukup tinggi.

#### 4.1.5. Implementasi Model

Setelah proses training berhasil dilakukan tentunya akan menghasilkan model deteksi CNN, model tersebut akan dilakukan uji tes. Tujuannya untuk mengetahui kinerja dari model tersebut. Pada proses ini dilakukan terhadap uji tes video secara realtime menggunakan kamera webcam. Adapun hasil prediksi dari video secara real time menggunakan webcam dapat dilihat pada gambar 4.11.







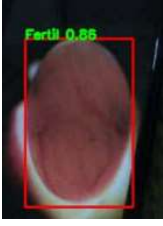




Gambar 4. 11 Hasil Implementasi Model








#### 4.1.6. Hasil Pengujian




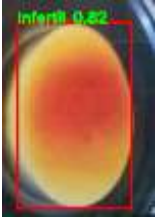



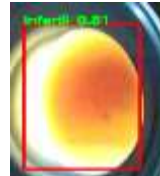
Dilakukan tahap pengujian pada 2 kondisi yang berbeda, yaitu pada saat Gelap dan Terang. Untuk membuktikan nilai akurasi dan tidaknya dalam mendeteksi telur bebek fertil dan infertil. Berikut Tabel 4.1 pengujian









Tabel 4. 1 Pengujian









No.	Hasil Gambar	Hasil Program	Jenis Telur	Kondisi Cahaya	Nilai Prediksi	Keterangan
1.		Terdeteksi	Fertil	Gelap	0,86	Sesuai

2.		Terdeteksi	Fertil	Gelap	0,83	Sesuai
3.		Terdeteksi	Fertil	Gelap	0,81	Sesuai
4.		Terdeteksi	Fertil	Gelap	0,85	Sesuai
5.		Terdeteksi	Fertil	Gelap	0.86	Sesuai
6.		Terdeteksi	Fertil	Gelap	0.81	Sesuai
7.		Terdeteksi	Fertil	Gelap	0.85	Sesuai
8.		Terdeteksi	Fertil	Gelap	0.85	Sesuai
9.		Terdeteksi	Fertil	Gelap	0.86	Sesuai

10.		Terdeteksi	Fertil	Gelap	0.85	Sesuai
11.		Terdeteksi	Fertil	Gelap	0.85	Sesuai
12.		Terdeteksi	Fertil	Gelap	0.82	Sesuai
13.		Terdeteksi	Fertil	Gelap	0.85	Sesuai
14.		Terdeteksi	Fertil	Gelap	0.87	Sesuai
15.		Terdeteksi	Fertil	Gelap	0.82	Sesuai
16.		Terdeteksi	Infertil	Gelap	0.78	Sesuai

17.		Terdeteksi	Infertil	Gelap	0.78	Sesuai
18.		Terdeteksi	Infertil	Gelap	0.81	Sesuai
19.		Terdeteksi	Infertil	Gelap	0.78	Sesuai
20.		Terdeteksi	Infertil	Gelap	0.82	Sesuai
21.		Terdeteksi	\ Infertil	Gelap	0.87	Sesuai
22.		Terdeteksi	\ Infertil	Gelap	0.82	Sesuai
23.		Terdeteksi	Infertil	Gelap	0.79	Sesuai
24.		Terdeteksi	Infertil	Gelap	0.81	Sesuai

25.		Terdeteksi	Infertil	Gelap	0.78	Sesuai
26.		Terdeteksi	Infertil	Gelap	0.81	Sesuai
27.		Terdeteksi	Infertil	Gelap	0.83	Sesuai
28.		Terdeteksi	Infertil	Gelap	0.86	Sesuai
29.		Terdeteksi	Infertil	Gelap	0.81	Sesuai
30.		Terdeteksi	Infertil	Gelap	0.81	Sesuai
31.		Terdeteksi	Fertil	Terang	0.68	Sesuai
32.		Terdeteksi	Fertil	Terang	0.66	Sesuai

33.		Terdeteksi	Fertil	Terang	0.57	Sesuai
34.		Terdeteksi	Fertil	Terang	0.61	Sesuai
35.		Terdeteksi	Fertil	Terang	0.57	Sesuai
36.		Terdeteksi	Infertil	Terang	0.52	Sesuai
37.		Terdeteksi	Infertil	Terang	0.54	Sesuai
38.		Terdeteksi	Infertil	Terang	0.63	Sesuai
39.		Terdeteksi	Infertil	Terang	-	Tidak Sesuai
40.		Terdeteksi	Infertil	Terang	-	Tidak Sesuai

Dari Tabel 4.1. didapatkan hasil nilai pengujian sebagai berikut

TP (True Positive) = 38

TN (True Negative) = 0

FP (False Positive) = 2

FN (False Negative) = 0

Dalam proses pengujian menggunakan rumus dari confusion matriks sebagai berikut :

$$\begin{aligned} \text{Akurasi} &= \frac{TP+TN}{(TP+FP+FN+TN)} \\ &= \frac{(38)+(0)}{(38)+(2)+(0)+(0)} = \frac{38}{40} = 0,95 \\ &= 0,95 \times 100 \% \\ &= 95\% \end{aligned}$$

Jadi hasil yang terdeteksi berjumlah 38 dan yang tidak terdeteksi 2, maka  $38 + 0 = 38$ ,  $38 + 2 = 40$  lalu hasilnya 38 dibagi  $40 = 0,95$  di kalikan 100 dan mendapatkan hasil 95%. Pada pengujian yang telah dilakukan sebanyak 40 kali. Maka didapatkan presentase akurasi dari pengujian dalam mengidentifikasi jenis telur bebek *fertil* dan *infertil* yaitu sebesar 95%



## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1. Kesimpulan**

Berdasarkan hasil pengujian dan analisa pada deteksi telur bebek fertil dan infertil menggunakan algoritma *Convolutional Neural Network (CNN)*, maka peneliian ini telah berhasil mendapatkan kesimpulan bahwa :

1. Penelitian ini telah berhasil dalam membangun model untuk mendeteksi telur bebek *fertil* dan *infertil* dengan menggunakan *algoritma Convolutional Neural Network (CNN)*. Dalam proses pembuatan model ini, diperlukan *dataset* yang sudah dilengkapi dengan *label* berupa kotak pembatas (*bounding box*) dan nama kelas untuk setiap objek yang ada di dalam citra tersebut.
2. Cara kerja metode CNN yaitu, CNN akan melatih dan menguji setiap gambar melalui serangkaian proses. Dimulai dari pemecahan gambar menjadi gambar yang lebih kecil, kemudian memasukkan setiap gambar yang lebih kecil ke neural network yang lebih kecil, menyimpan hasil dari masing-masing gambar kecil ke dalam array baru, downsampling atau mengurangi ukuran spasial untuk mengurangi jumlah parameter dan perhitungan ketika ukuran citra terlalu besar, dan membuat prediksi. . Hasil pengujian telur bebek fertil dan infertil sebanyak 40 kali dengan menggunakan CNN, mencapai akurasi sekitar 95% dalam membedakan kedua kategori telur.

#### **5.2. Saran**

Berdasarkan hasil pengujian dan analisa pada deteksi telur bebek fertil dan infertil menggunakan algoritma *Convolutional Neural Network (CNN)*, maka peneliian ini telah berhasil mendapatkan kesimpulan bahwa :

1. Jumlah dataset perlu diperbanyak untuk meningkatkan perfoma model deteksi.
2. Mengembangkan model ini kedalam android agar lebih mudah saat digunakan.



## DAFTAR PUSTAKA

- Agustien, L., Rohman, T., & Hujairi, A. W. (2021). Real-time Deteksi Masker Berbasis Deep Learning menggunakan Algoritma CNN YOLOv3. *Jurnal Teknologi Informasi Dan Terapan*, 8(2), 129–137. <https://doi.org/10.25047/jtit.v8i2.246>
- Ali Andre, J. (2016). Sistem Security Webcam Dengan Menggunakan Microsoft Visual Basic (6.0). *Rabit : Jurnal Teknologi Dan Sistem Informasi Univrab*, 1(2), 46–58. <https://doi.org/10.36341/rabit.v1i2.23>
- Amiril Danur Rahmah, S. (2020). Implementasi Algoritma Convolutional Neural Network Pada Klasifikasi Penyakit Padi Melalui Citra Daun. *Dspace.Uii.Ac.Id*. [https://dspace.uui.ac.id/handle/123456789/30189%0Ahttps://dspace.uui.ac.id/bitstream/handle/123456789/30189/16611043 Siti Rahmah Danur Amiril.pdf?sequence=1](https://dspace.uui.ac.id/handle/123456789/30189%0Ahttps://dspace.uui.ac.id/bitstream/handle/123456789/30189/16611043_Siti_Rahmah_Danur_Amiril.pdf?sequence=1)
- Brier, J., & lia dwi jayanti. (2020). *No 主観的健康感を中心とした在宅高齢者における健康関連指標に関する共分散構造分析Title* (Vol. 21, Issue 1). <http://journal.um-surabaya.ac.id/index.php/JKM/article/view/2203>
- Daya, M., Telur, T., Di, B., & Beki, U. K. M. (2022). 5 1-4 5. 5, 3504–3513.
- Dewi, S. R. (2018). Deep Learning Object Detection Pada Video. *Deep Learning Object Detection Pada Video Menggunakan Tensorflow Dan Convolutional Neural Network*, 1–60. [https://dspace.uui.ac.id/bitstream/handle/123456789/7762/14611242\\_Syarifah Rosita Dewi\\_Statistika.pdf?sequence=1](https://dspace.uui.ac.id/bitstream/handle/123456789/7762/14611242_SyarifahRositaDewi_Statistika.pdf?sequence=1)
- Harani, N. H., Prianto, C., & Hasanah, M. (2019). Deteksi Objek Dan Pengenalan Karakter Plat Nomor Kendaraan Indonesia Menggunakan Metode Convolutional Neural Network (CNN) Berbasis Python. *Jurnal Teknik Informatika*, 11(3), 47–53. <https://ejurnal.ulbi.ac.id/index.php/informatika/article/view/658>
- Hartiwi, Y., Rasywir, E., Pratama, Y., & Jusia, P. A. (2020). Eksperimen Pengenalan Wajah dengan fitur Indoor Positioning System menggunakan Algoritma CNN. *Paradigma - Jurnal Komputer Dan Informatika*, 22(2), 109–116. <https://doi.org/10.31294/p.v22i2.8906>
- Ihsan, C. N. (2021). Klasifikasi Data Radar Menggunakan Algoritma Convolutional Neural Network (CNN). *DoubleClick: Journal of Computer and Information Technology*, 4(2), 115. <https://doi.org/10.25273/doubleclick.v4i2.8188>
- Irfan Nugraha Pratama, Tatang Rohana, T. A. M. (2020). Pengenalan Sampah Plastik Dengan Model Convolutional Neural Network. *Conference on Innovation and Application of Science and Technology (CIASTECH 2020)*, *Ciastech*, 691–698.

- Kurniasari, N., & Sugiono, J. P. (2021). Deteksi Jalur Yang Terputus Pada Rangkaian Listrik Dalam Pcb Menggunakan Metode Convolutional Neural Network (Cnn). *Surabaya Jurnal Sistem Cerdas Dan Rekayasa (JSCR)*, 3(1), 2656–7504.
- Ma, S. (2022). *Deteksi Jenis Beras Menggunakan Algoritma YOLOv3. III*, 219–226.
- Perkasa, B. R., Sularsa, A., Pratondo, A., & Telkom, U. (2022). *IMPLEMENTASI KLASIFIKASI CITRA UNTUK MENDETEKSI EMBRIO BEBEK PADA APLIKASI MOBILE MENGGUNAKAN ARTIFICIAL INTELLIGENCE IMAGE CLASSIFICATION IMPLEMENTATION FOR DETECTING DUCK EMBRYOS*. 8(1), 1–7.
- Suartika E. P, I. W., Wijaya, A. Y., & Soelaiman, R. (2016). Klasifikasi Citra Menggunakan Convolutional Neural Network (Cnn) Pada Caltech 101Dewa, C. K., Fadhilah, A. L., & Afiahayati, A. (2018). Convolutional Neural Networks for Handwritten Javanese Character Recognition. *IJCCS (Indonesian Journal of Computing an. Jurnal Teknik ITS)*, 5(1), 76. <http://repository.its.ac.id/48842/>
- Sujita, S., Sari, N. H., Sinarep, S., Zainuri, A., & Kaliwantoro, N. (2022). *Aplikasi Alat Penetas Telur Kontrol Suhu dan Kelembaban di Desa Batu Tulis Kecamatan Jonggat Lombok Tengah*. 4(2), 103–108.
- Wantoro, A., Muludi, K., & Sukisno. (2019). Penerapan Logika Fuzzy pada Sistem Pendukung Keputusan Penentuan Kelayakan Kualitas Telur Bebek. *Jutis*, 7(1), 1–6.

## LAMPIRAN

### Lampiran 1 Lembar Persetujuan

**LEMBAR PERSETUJUAN**  
**DETEKSI TELUR BEBEK FERTIL DAN INFERTIL MENGGUNAKAN**  
**ALGORITMA CONVOLUTIONAL NEURAL NETWORK (CNN)**  
  
***DETECTION OF FERTIL AND INFERTIL DUCK EGG***  
***USING CONVOLUTIONAL NEURAL NETWORK***  
***(CNN) ALGORITHM***  
  
Tugas Akhir diajukan oleh :  
**Ricky Steven Chandra**  
**19416255201102**  
  
Program Studi Teknik Informatika  
Fakultas Ilmu Komputer  
Universitas Buana Perjuangan Karawang  
  
Karawang, 29 Juli 2023  
Menyetujui :  
  
Pembimbing I, Pembimbing II,  
   
  
**Dr. Hanny Hikmayanti.H, M.Kom** **Yana Cahyana, M.Kom**  
**NIDN: 0427037305** **NIDN: 0410077901**  
  
i

### Lampiran 2 Lembar Pengesahan

Lampiran 3 Lembar Pengesahan




**LEMBAR PENGESAHAN**

**DETEKSI TELUR BEBEK FERTIL DAN INFERTIL MENGGUNAKAN  
ALGORITMA CONVOLUTIONAL NEURAL NETWORK (CNN)**



***DETECTION OF FERTIL AND INFERTIL DUCK EGG  
USING CONVOLUTIONAL NEURAL NETWORK  
(CNN) ALGORITHM***

oleh :  
**Ricky Steven Chandra**  
**NIM : 19416255201102**

Tugas akhir ini telah diterima dan disahkan untuk memenuhi  
sebagian syarat memperoleh gelar sarjana  
pada Program Studi Teknik Informatika  
Fakultas Ilmu Komputer  
Universitas Buana Perjuangan Karawang  
Karawang, 22 Agustus 2023

Ketua Penguji,	Anggota Penguji I,	Anggota Penguji II,
		
<b>Sutan Faisal, M.Kom</b> NIDN: <u>0428047401</u>	<b>Tatang Rohana, M.Kom</b> NIDN: <u>0412047201</u>	<b>Dr. Hanny Hikmayanti, H, M.Kom</b> NIDN: <u>0427037305</u>

Mengetahui:

Dekan Fakultas Ilmu Komputer,	Koordinator Program Studi,
	
<b>Dr. Ahmad Fauzi, M.Kom</b> NIDN: <u>0419037701</u>	<b>Jamaludin Indra, M.Kom</b> NIDN: <u>0405058208</u>

ii

Lampiran 4 Lembar Perbaikan Ketua Penguji



UNIVERSITAS BUANA PERJUANGAN KARAWANG  
FAKULTAS ILMU KOMPUTER  
Terakreditasi BAN-PT

Jl. H.S. Ronggowaluyo Telukjambe Timur Karawang 41361 Telp./Fax. (0267) 8403140  
Site: <http://fik.ubpkarawang.ac.id> email: [fik@ubpkarawang.ac.id](mailto:fik@ubpkarawang.ac.id)

LEMBAR PERBAIKAN KETUA PENGUJI  
SIDANG TUGAS AKHIR

1. Nama Mahasiswa: Ricky Steven Chandra
2. NIM: 19416255201102
3. Program Studi: Teknik Informatika
4. Judul Tugas Akhir: Deteksi Telur Bebek fertil dan infertil Menggunakan Algoritma Convolutional Neural Network (CNN)

No.	Perbaikan	Paraf
1.	Tabel pengisian data yang ditambahkan	<i>st</i>
2.	Validasi hasil pengisian kelayakan	<i>st</i>

Demikian untuk dipergunakan sebagaimana mestinya.

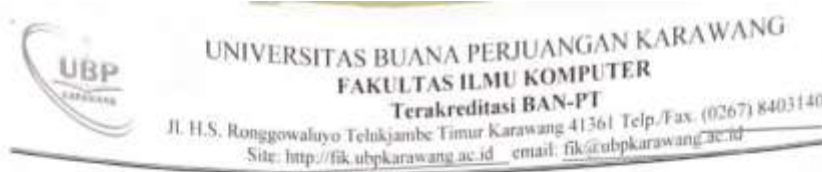
Karawang, 22 Agustus 2023  
Ketua Penguji,

*[Signature]*  
(Sulton Faisal, M.Kom)

Batas akhir perbaikan : \_\_\_\_\_

(Maksimal revisi hingga 15 (lima belas) hari kalender. Jika lebih dari 15 (lima belas) hari maka indeks nilai TA turun menjadi satu tingkat dari nilai indeks hasil keputusan sidang TA. Jika lebih dari 30 (tiga puluh) hari maka sidang ulang)

Lampiran 5 Lembar Perbaikan Anggota penguji 1



LEMBAR PERBAIKAN ANGGOTA PENGUJI I  
SIDANG TUGAS AKHIR

1. Nama Mahasiswa Ricky Steven Chandra
2. NIM 19416255201102
3. Program Studi Teknik Informatika
4. Judul Tugas Akhir Deteksi Telur Bebek Fertile dan Infertile Menggunakan Algoritma Convolutional Neural Network (CNN)

No.	Perbaikan	Paraf
1.	Abstrak	✓
2.	Latar belakang (urgensi penelitian)	✓
3.	Pemecahan masalah hasil penelitian dgn kesimpulan.	✓
4.	prosedur → prosed. akronis	✓
5.	Daftar Pustaka min 15 jurnal	✓
6.	Tabel perbaiki	✓

Demikian untuk dipergunakan sebagaimana mestinya.

Karawang, 16-8-2023  
Anggota Penguji I  
*[Signature]*  
(Ditandatangani)


Batas akhir perbaikan :

(Maksimal revisi hingga 15 (lima belas) hari kalender. Jika lebih dari 15 (lima belas) hari maka indeks nilai TA turun menjadi satu tingkat dari nilai indeks hasil keputusan sidang TA. Jika lebih dari 30 (tiga puluh) hari maka sidang ulang)

Lampiran 6 Lembar Perbaikan Anggota Penguji 2

Lampiran 7 Lembar Perbaikan Anggota Penguji 2

## Lampiran 8 Lembar Perbaikan Anggota Penguji 2



**UNIVERSITAS BUANA PERJUANGAN KARAWANG**  
**FAKULTAS ILMU KOMPUTER**  
**Terakreditasi BAN-PT**  
 Jl. H.S. Ronggowaluyo Telukjambe Timur Karawang 41361 Telp./Fax. (0267) 8403140  
 Site: <http://fik.ubpkarawang.ac.id> email: [fik@ubpkarawang.ac.id](mailto:fik@ubpkarawang.ac.id)


---

**LEMBAR PERBAIKAN ANGGOTA PENGUJI II**  
**SIDANG TUGAS AKHIR**

1. Nama Mahasiswa : Picky Steven Chandra  
 2. NIM : \_\_\_\_\_  
 3. Program Studi : \_\_\_\_\_  
 4. Judul Tugas Akhir : \_\_\_\_\_

No.	Perbaikan	Paraf
1	<div style="border: 1px solid black; height: 150px; margin: 5px;"></div>	<div style="border: 1px solid black; height: 150px; margin: 5px;"></div>

Densikan untuk dipergunakan sebagaimana mestinya.

Karawang, 16/8-2023  
 Anggota Penguji 2,  
  
Dr. HANNY HIKMAHYANTI

**Batas akhir perbaikan :** \_\_\_\_\_  
 (Maksimal revisi hingga 15 (lima belas) hari kalender. Jika lebih dari 15 (lima belas) hari maka indeks nilai TA turun menjadi satu tingkat dari nilai indeks hasil keputusan sidang TA. Jika lebih dari 30 (tiga puluh) hari maka sidang ulang)
















Lampiran 10 Dokumentasi Wawancara









### Lampiran 11 Validasi Peternak

✓ : Sesuai  
X : Tidak Sesuai

No.	Gambar	Jenis	Validasi Peternak	Tanda Tangan Peternak
1.		Fertil	✓	
2.		Fertil	✓	
3.		Fertil	✓	
4.		Fertil	✓	
5.		Fertil	✓	
6.		Fertil	✓	

7.		Fertil	✓	
8.		Fertil	✓	
9.		Fertil	✓	
10.		Fertil	✓	
11.		Infertil	X	
12.		Infertil	X	
13.		Infertil	X	
14.		Infertil	X	

15.		Infertil	X	
16.		Infertil	X	
17.		Infertil	X	
18.		Infertil	X	
19.		Infertil	X	
20.		Infertil	X	

Mengetahui,

Karawang 21 Agustus 2023

*Odah*

IBU ODAH  
(.....)

## Lampiran 12 Hasil Runing Google Colab

The screenshot shows a Google Colab notebook titled 'RikyDane.ipynb'. The code cell contains the following commands:

```
!pip install ultralytics
import ultralytics
ultralytics.checks()

!python train model=yolov5.pt data=custom.yaml epochs=150 imgsz=640
```

The output shows the installation of Ultralytics YOLOv5.0.139 and the training process. A table lists the model's parameters:

	from	n	param	module	argument1
0	-1	1	920	ultralytics.nn.modules.conv.Conv	[3, 32, 3, 2]
1	-1	1	38360	ultralytics.nn.modules.conv.Conv	[32, 64, 3, 2]
2	-1	1	20950	ultralytics.nn.modules.block.C2F	[64, 64, 1, True]
3	-1	1	72040	ultralytics.nn.modules.conv.Conv	[64, 128, 3, 2]
4	-1	2	107632	ultralytics.nn.modules.block.C2F	[128, 128, 2, True]
5	-1	1	296420	ultralytics.nn.modules.conv.Conv	[128, 256, 3, 2]
6	-1	2	786400	ultralytics.nn.modules.block.C2F	[256, 256, 2, True]
7	-1	1	1180672	ultralytics.nn.modules.conv.Conv	[256, 512, 3, 2]

The screenshot shows the continuation of the training process. The code cell contains the following commands:

```
!python train model=yolov5.pt data=custom.yaml epochs=150 imgsz=640
```

The output shows the training progress, including the number of layers, parameters, and gradients. The model summary is as follows:

```
Model summary: 225 layers, 11136374 parameters, 11136358 gradients
```

The training process is completed, and the model is saved. The output shows the final results of the training process.

Using 8 dataloader workers  
Logging results to runs/detect/traja  
Starting training for 350 epochs...

Epoch	GPU_mem	box_loss	cls_loss	df1_loss	Instances	Size	map50	map50-95
1/150	2.940	1.377	1.540	1.584	10	640: 100% 57/57 [02:29:00:00, 1.83s/it]	0.77	0.429
Class		Images	Instances	Box(P)	8			
all	60	60	60	0.033				
Epoch	GPU_mem	box_loss	cls_loss	df1_loss	Instances	Size	map50	map50-95
2/150	2.990	1.402	0.8020	1.528	11	640: 100% 57/57 [01:01:00:00, 1.00s/it]	0.104	0.0001
Class		Images	Instances	Box(P)	8			
all	60	60	60	0.098	0.403			
Epoch	GPU_mem	box_loss	cls_loss	df1_loss	Instances	Size	map50	map50-95
3/150	40	1.438	0.8020	1.543	9	640: 100% 57/57 [01:07:00:00, 1.10s/it]	0.475	0.101
Class		Images	Instances	Box(P)	8			
all	60	60	60	0.4	0.927			
Epoch	GPU_mem	box_loss	cls_loss	df1_loss	Instances	Size	map50	map50-95
4/150	4.050	1.30	0.780	1.495	10	640: 100% 57/57 [01:09:00:00, 1.22s/it]	0.001	0.001
Class		Images	Instances	Box(P)	8			
all	60	60	60	0.005	0.978			
Epoch	GPU_mem	box_loss	cls_loss	df1_loss	Instances	Size	map50	map50-95
5/150	2.880	1.000	0.7541	1.524	10	640: 100% 57/57 [01:05:00:00, 1.10s/it]	0.978	0.725
Class		Images	Instances	Box(P)	8			
all	60	60	60	0.947	0.95			

Epoch	GPU_mem	box_loss	cls_loss	df1_loss	Instances	Size	map50	map50-95
6/150	4.020	1.334	0.7182	1.447	10	640: 100% 57/57 [01:04:00:00, 1.13s/it]	0.004	0.101
Class		Images	Instances	Box(P)	8			
all	60	60	60	0.010	0.987			
Epoch	GPU_mem	box_loss	cls_loss	df1_loss	Instances	Size	map50	map50-95
7/150	7.40	1.342	0.6787	1.471	8	640: 100% 57/57 [01:06:00:00, 1.10s/it]	0.344	0.000
Class		Images	Instances	Box(P)	8			
all	60	60	60	0.757	0.980			
Epoch	GPU_mem	box_loss	cls_loss	df1_loss	Instances	Size	map50	map50-95
8/150	4.030	1.371	0.6090	1.447	4	640: 100% 57/57 [01:06:00:00, 1.10s/it]	0.404	0.025
Class		Images	Instances	Box(P)	8			
all	60	60	60	0.030	0.730			
Epoch	GPU_mem	box_loss	cls_loss	df1_loss	Instances	Size	map50	map50-95
9/150	4.020	1.335	0.6884	1.478	7	640: 100% 57/57 [01:05:00:00, 1.10s/it]	0.005	0.110
Class		Images	Instances	Box(P)	8			
all	60	60	60	0.003	1			
Epoch	GPU_mem	box_loss	cls_loss	df1_loss	Instances	Size	map50	map50-95
10/150	4.050	1.31	0.6002	1.434	15	640: 100% 57/57 [01:06:00:00, 1.17s/it]	0.005	0.700
Class		Images	Instances	Box(P)	8			
all	60	60	60	0.984	1			
Epoch	GPU_mem	box_loss	cls_loss	df1_loss	Instances	Size	map50	map50-95
11/150	40	1.305	0.6943	1.441	0	640: 100% 57/57 [01:05:00:00, 1.15s/it]	0.005	0.701
Class		Images	Instances	Box(P)	8			
all	60	60	60	0.900	1			



Memorandum Abstrak

translate - Penulisan Gens

Ukuta Penulisan Tata

YOLOv4 Tutorial - Colab

RikyDane.ipynb - Colab

https://colab.research.google.com/drive/1T5tACDUMWvCNATuCCVWvQZ3msDnR8Vtwhuqz14mcDfXmH1

RikyDane.ipynb

File Edit View Insert Runtime Tools Help Last edited on July 22

Connect

Code + Test

Epoch	GPU_mem	box_loss	cls_loss	obj_loss	Instances	Size
12/150	4.020	1.283	0.8197	1.489	10	640: 100% 57/57 [01:04:00:00, 1.13s/it]
Class	Images	Instances	Box(P)	8	mAP50 mAP50-95: 100% 2/2 [00:00:00:00, 2.01it/s]	
all	60	60	0.007	1	0.995 0.743	
Epoch	GPU_mem	box_loss	cls_loss	obj_loss	Instances	Size
13/150	4.020	1.275	0.8880	1.453	10	640: 100% 57/57 [01:04:00:00, 1.10s/it]
Class	Images	Instances	Box(P)	8	mAP50 mAP50-95: 100% 2/2 [00:00:00:00, 2.06it/s]	
all	60	60	0.005	1	0.995 0.755	
Epoch	GPU_mem	box_loss	cls_loss	obj_loss	Instances	Size
14/150	4.020	1.270	0.830	1.433	1	640: 100% 57/57 [01:05:00:00, 1.14s/it]
Class	Images	Instances	Box(P)	8	mAP50 mAP50-95: 100% 2/2 [00:00:00:00, 2.10it/s]	
all	60	60	0.008	1	0.995 0.708	
Epoch	GPU_mem	box_loss	cls_loss	obj_loss	Instances	Size
15/150	3.980	1.247	0.5880	1.397	11	640: 100% 57/57 [01:05:00:00, 1.15s/it]
Class	Images	Instances	Box(P)	8	mAP50 mAP50-95: 100% 2/2 [00:00:00:00, 2.16it/s]	
all	60	60	0.008	1	0.995 0.755	
Epoch	GPU_mem	box_loss	cls_loss	obj_loss	Instances	Size
16/150	4.020	1.243	0.5880	1.39	10	640: 100% 57/57 [01:06:00:00, 1.17s/it]
Class	Images	Instances	Box(P)	8	mAP50 mAP50-95: 100% 2/2 [00:00:00:00, 2.10it/s]	
all	60	60	0.005	1	0.995 0.756	
Epoch	GPU_mem	box_loss	cls_loss	obj_loss	Instances	Size
17/150	4.020	1.287	0.574	1.388	7	640: 100% 57/57 [01:05:00:00, 1.15s/it]
Class	Images	Instances	Box(P)	8	mAP50 mAP50-95: 100% 2/2 [00:00:00:00, 2.10it/s]	
all	60	60	0.005	1	0.995 0.742	

Search

3:40 PM 8/17/2023

Memorandum Abstrak

translate - Penulisan Gens

Ukuta Penulisan Tata

YOLOv4 Tutorial - Colab

RikyDane.ipynb - Colab

https://colab.research.google.com/drive/1T5tACDUMWvCNATuCCVWvQZ3msDnR8Vtwhuqz14mcDfXmH1

RikyDane.ipynb

File Edit View Insert Runtime Tools Help Last edited on July 22

Connect

Code + Test

Epoch	GPU_mem	box_loss	cls_loss	obj_loss	Instances	Size
18/150	4.020	1.251	0.6054	1.39	11	640: 100% 57/57 [01:05:00:00, 1.10s/it]
Class	Images	Instances	Box(P)	8	mAP50 mAP50-95: 100% 2/2 [00:00:00:00, 2.05it/s]	
all	60	60	0.000	1	0.995 0.767	
Epoch	GPU_mem	box_loss	cls_loss	obj_loss	Instances	Size
19/150	3.990	1.250	0.6000	1.388	11	640: 100% 57/57 [01:03:00:00, 1.12s/it]
Class	Images	Instances	Box(P)	8	mAP50 mAP50-95: 100% 2/2 [00:00:00:00, 2.12it/s]	
all	60	60	0.000	1	0.995 0.757	
Epoch	GPU_mem	box_loss	cls_loss	obj_loss	Instances	Size
20/150	4.020	1.227	0.5784	1.344	14	640: 100% 57/57 [01:05:00:00, 1.10s/it]
Class	Images	Instances	Box(P)	8	mAP50 mAP50-95: 100% 2/2 [00:00:00:00, 2.13it/s]	
all	60	60	0.002	1	0.995 0.74	
Epoch	GPU_mem	box_loss	cls_loss	obj_loss	Instances	Size
21/150	4.020	1.244	0.5075	1.390	10	640: 100% 57/57 [01:05:00:00, 1.15s/it]
Class	Images	Instances	Box(P)	8	mAP50 mAP50-95: 100% 2/2 [00:00:00:00, 2.40it/s]	
all	60	60	0.000	1	0.995 0.774	
Epoch	GPU_mem	box_loss	cls_loss	obj_loss	Instances	Size
22/150	4.020	1.234	0.5750	1.382	11	640: 100% 57/57 [01:06:00:00, 1.10s/it]
Class	Images	Instances	Box(P)	8	mAP50 mAP50-95: 100% 2/2 [00:00:00:00, 2.18it/s]	
all	60	60	0.001	1	0.995 0.766	
Epoch	GPU_mem	box_loss	cls_loss	obj_loss	Instances	Size
23/150	3.980	1.291	0.5700	1.371	12	640: 100% 57/57 [01:03:00:00, 1.12s/it]
Class	Images	Instances	Box(P)	8	mAP50 mAP50-95: 100% 2/2 [00:00:00:00, 2.02it/s]	
all	60	60	0.000	1	0.995 0.700	

Search

3:40 PM 8/17/2023

Memberat Abriak x Translate - Panduan Gens x Ushat Panduan Tata R... x YOLOv8 Tutorial - Colabon x RikyDane.pyth - Colabon x

https://colab.research.google.com/drive/1EzYCOUWwCHM7uCWVwZ3msDuP8/withdata?usp=sharing

RikyDane.pyth

File Edit View Insert Runtime Tools Help Last edited on July 22

Connect

Epoch	GPU_mem	box_loss	cls_loss	obj_loss	Instances	Size
10/150	4.836	1.231	0.5642	1.367	11	Size: 100% 57/57 [01:05:00:00, 1.16s/it] Data: 100% 57/57 [01:05:00:00, 1.16s/it] mAP50 mAP50-95: 100% 2/2 [00:00:00:00, 2.40it/s] 0.995 0.996
Class	Images	Instances	Box(P)	8		
all	60	60	0.996	1		
Epoch	GPU_mem	box_loss	cls_loss	obj_loss	Instances	Size
25/150	4.020	1.190	0.5500	1.360	11	Size: 100% 57/57 [01:05:00:00, 1.12s/it] Data: 100% 57/57 [01:05:00:00, 1.12s/it] mAP50 mAP50-95: 100% 2/2 [00:01:00:00, 1.54it/s] 0.995 0.977
Class	Images	Instances	Box(P)	8		
all	60	60	0.996	1		
Epoch	GPU_mem	box_loss	cls_loss	obj_loss	Instances	Size
26/150	4.010	1.19	0.5310	1.360	11	Size: 100% 57/57 [01:05:00:00, 1.14s/it] Data: 100% 57/57 [01:05:00:00, 1.14s/it] mAP50 mAP50-95: 100% 2/2 [00:00:00:00, 2.40it/s] 0.995 0.978
Class	Images	Instances	Box(P)	8		
all	60	60	0.996	1		
Epoch	GPU_mem	box_loss	cls_loss	obj_loss	Instances	Size
27/150	3.990	1.200	0.5451	1.378	10	Size: 100% 57/57 [01:04:00:00, 1.14s/it] Data: 100% 57/57 [01:04:00:00, 1.14s/it] mAP50 mAP50-95: 100% 2/2 [00:01:00:00, 1.70it/s] 0.995 0.983
Class	Images	Instances	Box(P)	8		
all	60	60	0.996	1		
Epoch	GPU_mem	box_loss	cls_loss	obj_loss	Instances	Size
28/150	4.020	1.207	0.5437	1.37	9	Size: 100% 57/57 [01:05:00:00, 1.15s/it] Data: 100% 57/57 [01:05:00:00, 1.15s/it] mAP50 mAP50-95: 100% 2/2 [00:01:00:00, 1.80it/s] 0.995 0.985
Class	Images	Instances	Box(P)	8		
all	60	60	0.994	1		
Epoch	GPU_mem	box_loss	cls_loss	obj_loss	Instances	Size
29/150	4.000	1.216	0.5455	1.367	10	Size: 100% 57/57 [01:04:00:00, 1.12s/it] Data: 100% 57/57 [01:04:00:00, 1.12s/it] mAP50 mAP50-95: 100% 2/2 [00:01:00:00, 1.85it/s] 0.995 0.977
Class	Images	Instances	Box(P)	8		
all	60	60	0.996	1		

Search

3:40 PM 8/17/2023

Memberat Abriak x Translate - Panduan Gens x Ushat Panduan Tata R... x YOLOv8 Tutorial - Colabon x RikyDane.pyth - Colabon x

https://colab.research.google.com/drive/1EzYCOUWwCHM7uCWVwZ3msDuP8/withdata?usp=sharing

RikyDane.pyth

File Edit View Insert Runtime Tools Help Last edited on July 22

Connect

Epoch	GPU_mem	box_loss	cls_loss	obj_loss	Instances	Size
30/150	4.020	1.180	0.5434	1.342	11	Size: 100% 57/57 [01:05:00:00, 1.13s/it] Data: 100% 57/57 [01:05:00:00, 1.13s/it] mAP50 mAP50-95: 100% 2/2 [00:00:00:00, 2.15it/s] 0.995 0.992
Class	Images	Instances	Box(P)	8		
all	60	60	0.997	1		
Epoch	GPU_mem	box_loss	cls_loss	obj_loss	Instances	Size
31/150	3.980	1.183	0.5272	1.337	8	Size: 100% 57/57 [01:04:00:00, 1.14s/it] Data: 100% 57/57 [01:04:00:00, 1.14s/it] mAP50 mAP50-95: 100% 2/2 [00:01:00:00, 1.67it/s] 0.995 0.993
Class	Images	Instances	Box(P)	8		
all	60	60	0.996	1		
Epoch	GPU_mem	box_loss	cls_loss	obj_loss	Instances	Size
32/150	4.010	1.200	0.5374	1.36	9	Size: 100% 57/57 [01:04:00:00, 1.14s/it] Data: 100% 57/57 [01:04:00:00, 1.14s/it] mAP50 mAP50-95: 100% 2/2 [00:00:00:00, 2.11it/s] 0.995 0.995
Class	Images	Instances	Box(P)	8		
all	60	60	0.998	1		
Epoch	GPU_mem	box_loss	cls_loss	obj_loss	Instances	Size
33/150	4.020	1.182	0.5385	1.341	11	Size: 100% 57/57 [01:05:00:00, 1.14s/it] Data: 100% 57/57 [01:05:00:00, 1.14s/it] mAP50 mAP50-95: 100% 2/2 [00:00:00:00, 1.42it/s] 0.995 0.993
Class	Images	Instances	Box(P)	8		
all	60	60	0.997	1		
Epoch	GPU_mem	box_loss	cls_loss	obj_loss	Instances	Size
34/150	4.020	1.176	0.5380	1.353	6	Size: 100% 57/57 [01:04:00:00, 1.14s/it] Data: 100% 57/57 [01:04:00:00, 1.14s/it] mAP50 mAP50-95: 100% 2/2 [00:00:00:00, 2.07it/s] 0.995 0.9
Class	Images	Instances	Box(P)	8		
all	60	60	0.999	1		
Epoch	GPU_mem	box_loss	cls_loss	obj_loss	Instances	Size
35/150	4	1.150	0.511	1.32	9	Size: 100% 57/57 [01:00:00:00, 1.16s/it] Data: 100% 57/57 [01:00:00:00, 1.16s/it] mAP50 mAP50-95: 100% 2/2 [00:00:00:00, 2.83it/s] 0.995 0.988
Class	Images	Instances	Box(P)	8		
all	60	60	0.998	1		

Search

3:40 PM 8/17/2023



The image displays two screenshots of the RStudio IDE, showing the 'Environment' pane with a table of model metrics. The top screenshot shows metrics for epochs 36 to 41, and the bottom screenshot shows metrics for epochs 42 to 47. The table includes columns for Epoch, GPU\_mem, box\_loss, cls\_loss, df1\_loss, Instances, Size, and various performance metrics like GAB, MAP50, and mAP50-95.

Epoch	GPU_mem	box_loss	cls_loss	df1_loss	Instances	Size	GAB	MAP50	mAP50-95
36/150	4.820	1.165	0.5343	1.33	13	640: 100% 57/57 [01:03:00:00, 1.11s/it]	100%	2/2	[00:00:00:00, 2.20it/s]
37/150	4.816	1.170	0.5185	1.35	8	640: 100% 57/57 [01:05:00:00, 1.14s/it]	100%	2/2	[00:00:00:00, 2.40it/s]
38/150	4.826	1.18	0.5183	1.35	10	640: 100% 57/57 [01:06:00:00, 1.15s/it]	100%	2/2	[00:00:00:00, 2.64it/s]
39/150	3.980	1.182	0.5423	1.35	9	640: 100% 57/57 [01:08:00:00, 1.15s/it]	100%	2/2	[00:00:00:00, 2.00it/s]
40/150	4.816	1.151	0.5197	1.33	9	640: 100% 57/57 [01:04:00:00, 1.12s/it]	100%	2/2	[00:00:00:00, 2.20it/s]
41/150	4.820	1.148	0.5224	1.33	14	640: 100% 57/57 [01:06:00:00, 1.16s/it]	100%	2/2	[00:00:00:00, 2.33it/s]
42/150	4.816	1.167	0.5150	1.345	8	640: 100% 57/57 [01:04:00:00, 1.12s/it]	100%	2/2	[00:00:00:00, 2.54it/s]
43/150	3.996	1.182	0.5141	1.327	12	640: 100% 57/57 [01:06:00:00, 1.17s/it]	100%	2/2	[00:00:00:00, 2.40it/s]
44/150	4.826	1.14	0.5079	1.325	10	640: 100% 57/57 [01:08:00:00, 1.15s/it]	100%	2/2	[00:00:00:00, 1.88it/s]
45/150	4.820	1.157	0.5032	1.332	8	640: 100% 57/57 [01:05:00:00, 1.13s/it]	100%	2/2	[00:00:00:00, 2.51it/s]
46/150	4.826	1.147	0.4987	1.323	8	640: 100% 57/57 [01:08:00:00, 1.30s/it]	100%	2/2	[00:00:00:00, 2.30it/s]
47/150	3.996	1.157	0.4928	1.332	8	640: 100% 57/57 [01:03:00:00, 1.12s/it]	100%	2/2	[00:00:00:00, 2.41it/s]

The image displays two screenshots of a Jupyter Notebook interface, showing the execution of a training script. The notebook is titled "RikyDane.ipynb" and is running on a Google Colab environment. The interface includes a menu bar (File, Edit, View, Insert, Runtime, Tools, Help) and a toolbar with various icons for file operations and runtime management. The main content area shows a series of tables representing the output of the training process, organized by epoch.

The top screenshot shows epochs 40 to 53. The bottom screenshot shows epochs 54 to 59. Each epoch's data is presented in a table with columns for GPU mem, box loss, cls loss, df1 loss, instances, size, and various metrics.

Epoch	GPU mem	box_loss	cls_loss	df1_loss	Instances	Size	Other Metrics
40/150	4.030	1.138	0.5853	1.204	9	0.00%	100% 37/37 [01:03:00:00, 1.11s/it]
41/150	4.030	1.141	0.5184	1.337	10	0.00%	100% 37/37 [01:03:00:00, 1.11s/it]
42/150	4.030	1.137	0.5080	1.338	8	0.00%	100% 37/37 [01:03:00:00, 1.11s/it]
43/150	4.030	1.137	0.5080	1.338	8	0.00%	100% 37/37 [01:03:00:00, 1.11s/it]
44/150	4.030	1.137	0.5080	1.338	8	0.00%	100% 37/37 [01:03:00:00, 1.11s/it]
45/150	4.030	1.137	0.5080	1.338	8	0.00%	100% 37/37 [01:03:00:00, 1.11s/it]
46/150	4.030	1.137	0.5080	1.338	8	0.00%	100% 37/37 [01:03:00:00, 1.11s/it]
47/150	4.030	1.137	0.5080	1.338	8	0.00%	100% 37/37 [01:03:00:00, 1.11s/it]
48/150	4.030	1.137	0.5080	1.338	8	0.00%	100% 37/37 [01:03:00:00, 1.11s/it]
49/150	4.030	1.137	0.5080	1.338	8	0.00%	100% 37/37 [01:03:00:00, 1.11s/it]
50/150	4.030	1.137	0.5080	1.338	8	0.00%	100% 37/37 [01:03:00:00, 1.11s/it]
51/150	4.030	1.137	0.5080	1.338	8	0.00%	100% 37/37 [01:03:00:00, 1.11s/it]
52/150	4.030	1.137	0.5080	1.338	8	0.00%	100% 37/37 [01:03:00:00, 1.11s/it]
53/150	4.030	1.137	0.5080	1.338	8	0.00%	100% 37/37 [01:03:00:00, 1.11s/it]
54/150	4.030	1.135	0.4858	1.337	10	0.00%	100% 37/37 [01:06:00:00, 1.17s/it]
55/150	4.030	1.135	0.4858	1.337	10	0.00%	100% 37/37 [01:06:00:00, 1.17s/it]
56/150	4.030	1.135	0.4858	1.337	10	0.00%	100% 37/37 [01:06:00:00, 1.17s/it]
57/150	4.030	1.135	0.4858	1.337	10	0.00%	100% 37/37 [01:06:00:00, 1.17s/it]
58/150	4.030	1.135	0.4858	1.337	10	0.00%	100% 37/37 [01:06:00:00, 1.17s/it]
59/150	4.030	1.135	0.4858	1.337	10	0.00%	100% 37/37 [01:06:00:00, 1.17s/it]

The image displays two screenshots of the RStudio IDE interface, showing the 'RStudio.ipynb' file. The top screenshot shows the first five epochs of training, and the bottom screenshot shows epochs 6 through 10. The table displays various metrics including GPU memory, box loss, cls loss, df1 loss, instances, size, and mAP50 for each epoch.

Epoch	GPU_mem	box_loss	cls_loss	df1_loss	Instances	Size	mAP50	mAP50-95	
60/150	4.810	1.183	0.4796	1.31	8	640: 100% 57/57 [01:00:00:00, 1.28s/it]	100% 2/2 [00:00:00:00, 1.38it/s]	0.995	0.825
61/150	4.810	1.886	0.4647	1.282	13	640: 100% 57/57 [01:00:00:00, 1.10s/it]	100% 2/2 [00:00:00:00, 2.60it/s]	0.995	0.827
62/150	4.810	1.195	0.4824	1.302	6	640: 100% 57/57 [01:00:00:00, 1.10s/it]	100% 2/2 [00:00:00:00, 2.47it/s]	0.995	0.82
63/150	3.996	1.07	0.4646	1.285	8	640: 100% 57/57 [01:00:00:00, 1.15s/it]	100% 2/2 [00:00:00:00, 1.92it/s]	0.995	0.815
64/150	4.810	1.885	0.4583	1.273	13	640: 100% 57/57 [01:00:00:00, 1.10s/it]	100% 2/2 [00:00:00:00, 2.30it/s]	0.995	0.810
65/150	4.810	1.186	0.4745	1.304	16	640: 100% 57/57 [01:00:00:00, 1.11s/it]	100% 2/2 [00:00:00:00, 2.35it/s]	0.995	0.825
66/150	4.810	1.082	0.4613	1.284	11	640: 100% 57/57 [01:00:00:00, 1.17s/it]	100% 2/2 [00:00:00:00, 1.38it/s]	0.995	0.782
67/150	3.996	1.888	0.4660	1.292	13	640: 100% 57/57 [01:00:00:00, 1.17s/it]	100% 2/2 [00:00:00:00, 2.30it/s]	0.995	0.828
68/150	4.810	1.894	0.4647	1.3	10	640: 100% 57/57 [01:00:00:00, 1.10s/it]	100% 2/2 [00:00:00:00, 1.52it/s]	0.995	0.827
69/150	4.810	1.084	0.4723	1.294	11	640: 100% 57/57 [01:00:00:00, 1.19s/it]	100% 2/2 [00:00:00:00, 2.37it/s]	0.995	0.801
70/150	4.810	1.06	0.4832	1.27	16	640: 100% 57/57 [01:00:00:00, 1.11s/it]	100% 2/2 [00:00:00:00, 2.30it/s]	0.995	0.81
71/150	3.996	1.881	0.4681	1.287	11	640: 100% 57/57 [01:00:00:00, 1.19s/it]	100% 2/2 [00:00:00:00, 2.75it/s]	0.995	0.845

The image displays two screenshots of a Jupyter notebook titled 'RikyDane.ipynb' running on a Google Colab instance. The notebook shows the progress of a machine learning training process across multiple epochs. The table below summarizes the data presented in the screenshots.

Epoch	GPU_mem	box_loss	cls_loss	df1_loss	Instances	Size	GPU_mem	box_loss	cls_loss	df1_loss	Instances	Size
72/150	4.020	1.094	0.888	1.291	0	0.095	4.020	1.094	0.888	1.291	0	0.095
73/150	4.020	1.094	0.888	1.291	0	0.095	4.020	1.094	0.888	1.291	0	0.095
74/150	4.020	1.094	0.888	1.291	0	0.095	4.020	1.094	0.888	1.291	0	0.095
75/150	4.020	1.094	0.888	1.291	0	0.095	4.020	1.094	0.888	1.291	0	0.095
76/150	4.020	1.094	0.888	1.291	0	0.095	4.020	1.094	0.888	1.291	0	0.095
77/150	4.020	1.094	0.888	1.291	0	0.095	4.020	1.094	0.888	1.291	0	0.095
78/150	4.020	1.094	0.888	1.291	0	0.095	4.020	1.094	0.888	1.291	0	0.095
79/150	4.020	1.094	0.888	1.291	0	0.095	4.020	1.094	0.888	1.291	0	0.095
80/150	4.020	1.094	0.888	1.291	0	0.095	4.020	1.094	0.888	1.291	0	0.095
81/150	4.020	1.094	0.888	1.291	0	0.095	4.020	1.094	0.888	1.291	0	0.095
82/150	4.020	1.094	0.888	1.291	0	0.095	4.020	1.094	0.888	1.291	0	0.095
83/150	4.020	1.094	0.888	1.291	0	0.095	4.020	1.094	0.888	1.291	0	0.095



The image displays two screenshots of a Jupyter Notebook titled 'RikyDane.ipynb' running on a Google Colab environment. The notebook is showing the training progress of a model, with data presented in a table format for each epoch.

**Top Screenshot (Epochs 84-89):**

Epoch	GPU_mem	box_loss	cls_loss	obj_loss	Instances	Size	DA0	DA0	DA0	DA0
84/150	4.036	1.044	0.4598	1.252	13	100%	57/57	[01:00:00:00, 1.22s/it]	100%	2/2 [00:00:00:00, 2.10it/s]
85/150	4.020	1.044	0.4521	1.205	10	100%	57/57	[01:00:00:00, 1.23s/it]	100%	2/2 [00:01:00:00, 1.50it/s]
86/150	4.020	1.020	0.4382	1.249	14	100%	57/57	[01:07:00:00, 1.30s/it]	100%	2/2 [00:01:00:00, 1.91it/s]
87/150	3.990	1.041	0.4550	1.265	14	100%	57/57	[01:07:00:00, 1.30s/it]	100%	2/2 [00:00:00:00, 2.91it/s]
88/150	4.020	1.05	0.4475	1.27	13	100%	57/57	[01:10:00:00, 1.24s/it]	100%	2/2 [00:00:00:00, 2.14it/s]
89/150	4.020	1.056	0.4500	1.274	11	100%	57/57	[01:00:00:00, 1.21s/it]	100%	2/2 [00:01:00:00, 1.91it/s]

**Bottom Screenshot (Epochs 90-95):**

Epoch	GPU_mem	box_loss	cls_loss	obj_loss	Instances	Size	DA0	DA0	DA0	DA0
90/150	4.020	1.040	0.4400	1.303	8	100%	57/57	[01:00:00:00, 1.22s/it]	100%	2/2 [00:00:00:00, 2.10it/s]
91/150	3.990	1.03	0.44	1.247	6	100%	57/57	[01:10:00:00, 1.23s/it]	100%	2/2 [00:00:00:00, 2.01it/s]
92/150	4.020	1.000	0.4202	1.235	7	100%	57/57	[01:05:00:00, 1.25s/it]	100%	2/2 [00:01:00:00, 1.83it/s]
93/150	4.020	1.000	0.4375	1.248	8	100%	57/57	[01:00:00:00, 1.18s/it]	100%	2/2 [00:00:00:00, 2.31it/s]
94/150	4.020	1.030	0.4433	1.261	9	100%	57/57	[01:09:00:00, 1.22s/it]	100%	2/2 [00:00:00:00, 2.14it/s]
95/150	3.980	1.027	0.4404	1.235	7	100%	57/57	[01:07:00:00, 1.19s/it]	100%	2/2 [00:00:00:00, 2.10it/s]

The image displays two sequential screenshots of a Jupyter Notebook titled 'Rikydane.ipynb'. The notebook is running on a Google Colab environment, as indicated by the URL in the browser's address bar: <https://colab.research.google.com/drive/1GyYCDUWwCvMwTucCWwWQ3msDvR5/v/ufu5uazr1KqoDfzcc0>.

The notebook interface includes a menu bar (File, Edit, View, Insert, Runtime, Tools, Help) and a toolbar with icons for file operations, search, and execution. The main content area shows a series of code cells, each representing an epoch of training. The output of each cell is a table of performance metrics.

**Epoch 96 to 101 (Top Screenshot):**

Epoch	GPU_mem	box_loss	cls_loss	df1_loss	Instances	Size
96/150	4.010	1.821	0.4380	1.258	11	640: 100% 57/57 [01:00:00:00, 1.39x/it]
Class	Images	Instances	Box(P)	8	MAP50	MAP50-95): 100% 2/2 [00:00:00:00, 2.24it/s]
all	60	60	0.995	1	0.995	0.883
97/150	4.010	1.803	0.4341	1.230	12	640: 100% 57/57 [01:11:00:00, 1.25x/it]
Class	Images	Instances	Box(P)	8	MAP50	MAP50-95): 100% 2/2 [00:00:00:00, 2.55it/s]
all	60	60	0.995	1	0.995	0.882
98/150	4.010	1.827	0.435	1.268	8	640: 100% 57/57 [01:10:00:00, 1.34x/it]
Class	Images	Instances	Box(P)	8	MAP50	MAP50-95): 100% 2/2 [00:00:00:00, 2.47it/s]
all	60	60	0.996	1	0.995	0.836
99/150	3.980	1.810	0.4259	1.25	12	640: 100% 57/57 [01:10:00:00, 1.73x/it]
Class	Images	Instances	Box(P)	8	MAP50	MAP50-95): 100% 2/2 [00:00:00:00, 2.38it/s]
all	60	60	0.998	1	0.995	0.889
100/150	4.0	1.822	0.4334	1.251	9	640: 100% 57/57 [01:09:00:00, 1.28x/it]
Class	Images	Instances	Box(P)	8	MAP50	MAP50-95): 100% 2/2 [00:01:00:00, 1.72it/s]
all	60	60	0.998	1	0.995	0.885
101/150	4.010	1.818	0.4278	1.251	12	640: 100% 57/57 [01:08:00:00, 1.39x/it]
Class	Images	Instances	Box(P)	8	MAP50	MAP50-95): 100% 2/2 [00:00:00:00, 2.10it/s]
all	60	60	0.997	1	0.995	0.836

**Epoch 102 to 107 (Bottom Screenshot):**

Epoch	GPU_mem	box_loss	cls_loss	df1_loss	Instances	Size
102/150	4.010	1.800	0.4315	1.22	12	640: 100% 57/57 [01:11:00:00, 1.35x/it]
Class	Images	Instances	Box(P)	8	MAP50	MAP50-95): 100% 2/2 [00:00:00:00, 2.22it/s]
all	60	60	0.996	1	0.995	0.886
103/150	3.980	1.809	0.427	1.241	10	640: 100% 57/57 [01:10:00:00, 1.28x/it]
Class	Images	Instances	Box(P)	8	MAP50	MAP50-95): 100% 2/2 [00:00:00:00, 2.58it/s]
all	60	60	0.998	1	0.995	0.851
104/150	4.010	0.9954	0.4275	1.23	8	640: 100% 57/57 [01:09:00:00, 1.21x/it]
Class	Images	Instances	Box(P)	8	MAP50	MAP50-95): 100% 2/2 [00:01:00:00, 1.72it/s]
all	60	60	0.998	1	0.995	0.835
105/150	4.010	0.9877	0.4246	1.227	11	640: 100% 57/57 [01:08:00:00, 1.39x/it]
Class	Images	Instances	Box(P)	8	MAP50	MAP50-95): 100% 2/2 [00:00:00:00, 2.80it/s]
all	60	60	0.998	1	0.995	0.856
106/150	4.010	0.9857	0.4181	1.228	5	640: 100% 57/57 [01:12:00:00, 1.28x/it]
Class	Images	Instances	Box(P)	8	MAP50	MAP50-95): 100% 2/2 [00:00:00:00, 2.35it/s]
all	60	60	0.998	1	0.995	0.885
107/150	4.0	0.9888	0.4285	1.224	9	640: 100% 57/57 [01:07:00:00, 1.19x/it]
Class	Images	Instances	Box(P)	8	MAP50	MAP50-95): 100% 2/2 [00:01:00:00, 1.90it/s]
all	60	60	0.998	1	0.995	0.855

The image displays two sequential screenshots of a Google Colab notebook titled "RikyDane.ipynb". The interface shows the "Runtime" tab, which provides a detailed log of training epochs. Each epoch's data is organized into a table with the following columns: Epoch, GPU mem, box\_loss, cls\_loss, df1\_loss, Instances, Size, and a set of performance metrics (DAP, mAP50, mAP50-95, and FPS).

**Epoch 100 Data (Top Screenshot):**

Epoch	GPU mem	box_loss	cls_loss	df1_loss	Instances	Size	DAP	mAP50	mAP50-95	FPS
100/150	4.81G	0.4307	0.4307	1.344	7	640	100% 57/57	[01:00:00:00, 1.39s/it]	100% 2/2	[00:01:00:00, 1.42it/s]
Class	Images	Instances	Box(P)							
all	68	68	0.997	1	0.995	0.863				

**Epoch 114 Data (Bottom Screenshot):**

Epoch	GPU mem	box_loss	cls_loss	df1_loss	Instances	Size	DAP	mAP50	mAP50-95	FPS
114/150	4.82G	0.3880	0.4168	1.221	24	640	100% 57/57	[01:00:00:00, 1.38s/it]	100% 2/2	[00:01:00:00, 1.52it/s]
Class	Images	Instances	Box(P)							
all	68	68	0.996	1	0.995	0.863				

The image shows two sequential screenshots of a Jupyter Notebook titled 'RikyDane.ipynb' running on a Google Colab environment. The notebook is displaying the output of a training loop, with each epoch's results printed to the console. The top screenshot shows epochs 119 through 123, and the bottom screenshot shows epochs 124 through 131. The output for each epoch includes GPU memory usage, loss values for box, cls, and df1, the number of instances, size, and mAP50/95 metrics.

**Epoch 119/119:**

GPU_mem	box_loss	cls_loss	df1_loss	Instances	Size	mAP50	mAP50-95
4.03G	0.9749	0.4088	1.212	7	640: 100% 57/57 [01:00:00:00, 1.71s/it]	100%	2/2 [00:01:00:00, 1.81it/s]
all	0.0	0.0	0.007	1	0.0%	0.070	

**Epoch 120/119:**

GPU_mem	box_loss	cls_loss	df1_loss	Instances	Size	mAP50	mAP50-95
4.03G	0.9678	0.4090	1.206	9	640: 100% 57/57 [01:00:00:00, 1.17s/it]	100%	2/2 [00:00:00:00, 2.43it/s]
all	0.0	0.0	0.007	1	0.0%	0.084	

**Epoch 121/119:**

GPU_mem	box_loss	cls_loss	df1_loss	Instances	Size	mAP50	mAP50-95
4.03G	0.9589	0.4049	1.21	12	640: 100% 57/57 [01:00:00:00, 1.22s/it]	100%	2/2 [00:00:00:00, 2.73it/s]
all	0.0	0.0	0.007	1	0.0%	0.067	

**Epoch 122/119:**

GPU_mem	box_loss	cls_loss	df1_loss	Instances	Size	mAP50	mAP50-95
3.99G	0.9400	0.4042	1.217	9	640: 100% 57/57 [01:00:00:00, 1.20s/it]	100%	2/2 [00:00:00:00, 2.52it/s]
all	0.0	0.0	0.006	1	0.0%	0.058	

**Epoch 123/119:**

GPU_mem	box_loss	cls_loss	df1_loss	Instances	Size	mAP50	mAP50-95
4.02G	0.9537	0.4052	1.211	7	640: 100% 57/57 [01:00:00:00, 1.19s/it]	100%	2/2 [00:01:00:00, 1.73it/s]
all	0.0	0.0	0.006	1	0.0%	0.084	

**Epoch 124/119:**

GPU_mem	box_loss	cls_loss	df1_loss	Instances	Size	mAP50	mAP50-95
4.03G	0.955	0.4033	1.212	7	640: 100% 57/57 [01:00:00:00, 1.22s/it]	100%	2/2 [00:00:00:00, 2.54it/s]
all	0.0	0.0	0.007	1	0.0%	0.058	

**Epoch 125/119:**

GPU_mem	box_loss	cls_loss	df1_loss	Instances	Size	mAP50	mAP50-95
4.03G	0.9559	0.4105	1.22	9	640: 100% 57/57 [01:11:00:00, 1.20s/it]	100%	2/2 [00:00:00:00, 2.13it/s]
all	0.0	0.0	0.006	1	0.0%	0.057	

**Epoch 126/119:**

GPU_mem	box_loss	cls_loss	df1_loss	Instances	Size	mAP50	mAP50-95
3.99G	0.9418	0.4018	1.200	9	640: 100% 57/57 [01:10:00:00, 1.23s/it]	100%	2/2 [00:00:00:00, 2.18it/s]
all	0.0	0.0	0.007	1	0.0%	0.075	

**Epoch 127/119:**

GPU_mem	box_loss	cls_loss	df1_loss	Instances	Size	mAP50	mAP50-95
4.03G	0.9517	0.401	1.218	9	640: 100% 57/57 [01:07:00:00, 1.19s/it]	100%	2/2 [00:01:00:00, 1.88it/s]
all	0.0	0.0	0.006	1	0.0%	0.088	

**Epoch 128/119:**

GPU_mem	box_loss	cls_loss	df1_loss	Instances	Size	mAP50	mAP50-95
4.02G	0.9422	0.3932	1.200	11	640: 100% 57/57 [01:09:00:00, 1.21s/it]	100%	2/2 [00:00:00:00, 2.52it/s]
all	0.0	0.0	0.006	1	0.0%	0.093	

**Epoch 129/119:**

GPU_mem	box_loss	cls_loss	df1_loss	Instances	Size	mAP50	mAP50-95
4.03G	0.9509	0.4002	1.224	13	640: 100% 57/57 [01:07:00:00, 1.10s/it]	100%	2/2 [00:00:00:00, 2.27it/s]
all	0.0	0.0	0.006	1	0.0%	0.063	

**Epoch 130/119:**

GPU_mem	box_loss	cls_loss	df1_loss	Instances	Size	mAP50	mAP50-95
3.99G	0.9183	0.4027	1.204	9	640: 100% 57/57 [01:00:00:00, 1.16s/it]	100%	2/2 [00:01:00:00, 1.88it/s]
all	0.0	0.0	0.006	1	0.0%	0.057	

**Epoch 131/119:**

GPU_mem	box_loss	cls_loss	df1_loss	Instances	Size	mAP50	mAP50-95
3.99G	0.9183	0.4027	1.204	9	640: 100% 57/57 [01:00:00:00, 1.16s/it]	100%	2/2 [00:01:00:00, 1.88it/s]
all	0.0	0.0	0.006	1	0.0%	0.057	



The image displays two screenshots of a Jupyter notebook titled 'RikyDane.ipynb' running on a Google Colab environment. The notebook is showing the progress of a training process, with each epoch's results printed out.

**Top Screenshot (Epochs 132-137):**

Epoch	GPU_mem	box_loss	cls_loss	df1_loss	Instances	Size	DAP	AP50	AP50-95
132/150	4.03G	0.9962	0.3818	1.18	9	100% 57/57 [01:07:00:00, 1.19s/it]	100%	2/2	[00:00:00:00, 1.22it/s]
133/150	4.03G	0.9101	0.3828	1.188	10	100% 57/57 [01:07:00:00, 1.19s/it]	100%	2/2	[00:00:00:00, 1.80it/s]
134/150	4.03G	0.9395	0.3943	1.193	8	100% 57/57 [01:08:00:00, 1.19s/it]	100%	2/2	[00:01:00:00, 1.72it/s]
135/150	3.99G	0.900	0.3883	1.188	8	100% 57/57 [01:08:00:00, 1.21s/it]	100%	2/2	[00:00:00:00, 1.67it/s]
136/150	4.02G	0.9134	0.3883	1.200	3	100% 57/57 [01:00:00:00, 1.10s/it]	100%	2/2	[00:01:00:00, 1.55it/s]
137/150	4.02G	0.9228	0.3911	1.200	8	100% 57/57 [01:07:00:00, 1.19s/it]	100%	2/2	[00:00:00:00, 1.57it/s]

**Bottom Screenshot (Epochs 138-143):**

Epoch	GPU_mem	box_loss	cls_loss	df1_loss	Instances	Size	DAP	AP50	AP50-95
138/150	4.02G	0.8118	0.3813	1.184	9	100% 57/57 [01:10:00:00, 1.21s/it]	100%	2/2	[00:00:00:00, 1.69it/s]
139/150	3.98G	0.9492	0.4006	1.201	8	100% 57/57 [01:00:00:00, 1.20s/it]	100%	2/2	[00:00:00:00, 1.87it/s]
140/150	4.02G	0.9029	0.3797	1.19	13	100% 57/57 [01:00:00:00, 1.20s/it]	100%	2/2	[00:01:00:00, 1.40it/s]
Closing dataloader model:									
augmentations: Blur(p=0.01, blur_limit=(3, 7)), PadAndJitter(p=0.01, blur_limit=(3, 7), togray(p=0.01), CLAHE(p=0.01, clip_limit=(1, 4.0), tile_grid_size=(8,									
141/150	4.01G	0.8123	0.3021	1.201	4	100% 57/57 [01:12:00:00, 1.20s/it]	100%	2/2	[00:00:00:00, 1.53it/s]
142/150	4.02G	0.9354	0.3229	1.202	4	100% 57/57 [01:00:00:00, 1.21s/it]	100%	2/2	[00:00:00:00, 1.53it/s]
143/150	3.98G	0.9232	0.3233	1.207	4	100% 57/57 [01:00:00:00, 1.20s/it]			

The image displays two screenshots of a Google Colab notebook titled 'RikyDane.ipynb'. The notebook is running a YOLOv8 training script. The top screenshot shows the training progress from epoch 144 to 149. The bottom screenshot shows epoch 150 and the final training summary.

**Epoch 144/150**

Epoch	GPU_mem	box_loss	cls_loss	df1_loss	Instances	Size
144/150	4.830	0.0125	0.3231	1.388	4	640: 100% 57/57 [01:00:00:00, 1.17s/it]
Class	Images	Instances	Box(P)	F	AP50	AP50-95): 100% 2/2 [00:01:00:00, 1.08it/s]
all	60	60	0.997	1	0.995	0.874

**Epoch 145/150**

Epoch	GPU_mem	box_loss	cls_loss	df1_loss	Instances	Size
145/150	4.830	0.9951	0.3148	1.202	4	640: 100% 57/57 [01:00:00:00, 1.13s/it]
Class	Images	Instances	Box(P)	F	AP50	AP50-95): 100% 2/2 [00:00:00:00, 2.31it/s]
all	60	60	0.997	1	0.995	0.875

**Epoch 146/150**

Epoch	GPU_mem	box_loss	cls_loss	df1_loss	Instances	Size
146/150	4.830	0.8952	0.3128	1.278	4	640: 100% 57/57 [01:00:00:00, 1.20s/it]
Class	Images	Instances	Box(P)	F	AP50	AP50-95): 100% 2/2 [00:00:00:00, 2.28it/s]
all	60	60	0.997	1	0.995	0.877

**Epoch 147/150**

Epoch	GPU_mem	box_loss	cls_loss	df1_loss	Instances	Size
147/150	3.990	0.9928	0.3141	1.285	4	640: 100% 57/57 [01:00:00:00, 1.13s/it]
Class	Images	Instances	Box(P)	F	AP50	AP50-95): 100% 2/2 [00:00:00:00, 2.43it/s]
all	60	60	0.997	1	0.995	0.873

**Epoch 148/150**

Epoch	GPU_mem	box_loss	cls_loss	df1_loss	Instances	Size
148/150	4.056	0.8880	0.3094	1.250	4	640: 100% 57/57 [01:00:00:00, 1.20s/it]
Class	Images	Instances	Box(P)	F	AP50	AP50-95): 100% 2/2 [00:00:00:00, 2.79it/s]
all	60	60	0.997	1	0.995	0.873

**Epoch 149/150**

Epoch	GPU_mem	box_loss	cls_loss	df1_loss	Instances	Size
149/150	4.830	0.990	0.3078	1.260	4	640: 100% 57/57 [01:00:00:00, 1.16s/it]
Class	Images	Instances	Box(P)	F	AP50	AP50-95): 100% 2/2 [00:01:00:00, 1.81it/s]
all	60	60	0.997	1	0.995	0.875

**Epoch 150/150**

Epoch	GPU_mem	box_loss	cls_loss	df1_loss	Instances	Size
150/150	4.830	0.8880	0.308	1.200	4	640: 100% 57/57 [01:05:00:00, 1.15s/it]
Class	Images	Instances	Box(P)	F	AP50	AP50-95): 100% 2/2 [00:01:00:00, 1.29it/s]
all	60	60	0.997	1	0.995	0.877

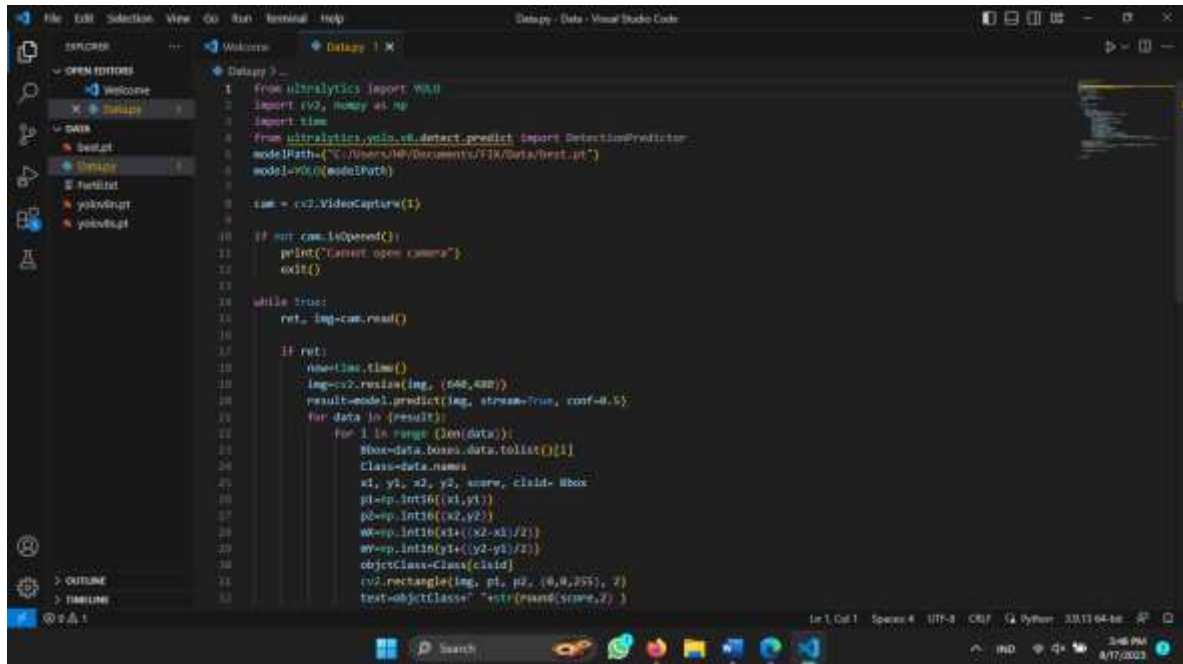
150 epochs completed in 2.912 hours.  
 Optimizer stripped from runs/detect/train/weights/last.pt, 22.5MB  
 Optimizer stripped from runs/detect/train/weights/best.pt, 22.5MB

Validating runs/detect/train/weights/best.pt...  
 Ultralytics YOLOv8 8.1.30 Python-3.10.6 torch-2.0.1+cu118 CUDA-0 (Tesla T4, 15102MiB)  
 Model summary (fused): 188 layers, 11130350 parameters, 0 gradients

Class	Images	Instances	Box(P)	F	AP50	AP50-95)
all	60	60	0.997	1	0.995	0.879
Turtill	60	30	0.998	1	0.995	0.879
Infortill	60	30	0.998	1	0.995	0.8

Speed: 2.1ms preprocess, 1.0ms inference, 0.0ms loss, 1.1ms postprocess per image  
 Results saved to runs/detect/train

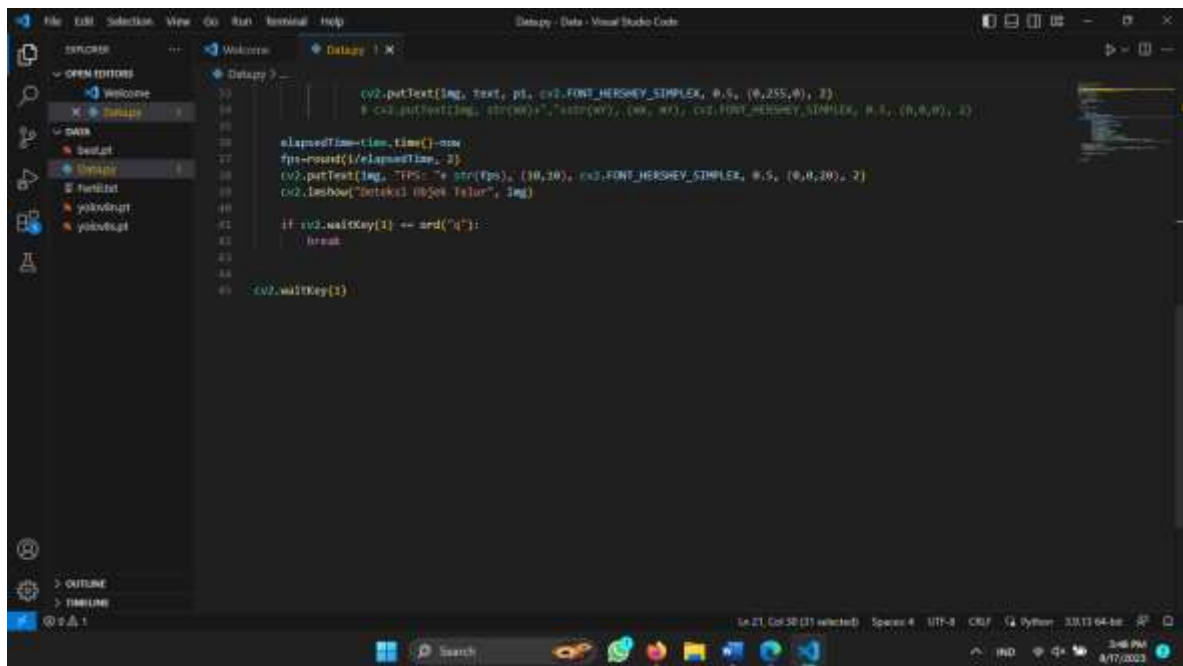
## Lampiran 13 Codingan Yolov8



```

1 from ultralytics import YOLO
2 import cv2, numpy as np
3 import time
4 from ultralytics.yolo.v8.detect_predict import DetectionPredictor
5 modelPath = ("C:/Users/140/Documents/Y18/Data/test.pt")
6 model=YOLO(modelPath)
7
8 cam = cv2.VideoCapture(1)
9
10 if not cam.isOpened():
11     print("Canst open camera")
12     exit()
13
14 while True:
15     ret, img=cam.read()
16
17     if ret:
18         now=time.time()
19         img=cv2.resize(img, (640,480))
20         results=model.predict(img, stream=True, conf=0.5)
21         for data in results:
22             for i in range(len(data)):
23                 bbox=data.bboxes.data.tolist()[i]
24                 Class=data.names
25                 x1, y1, x2, y2, score, class= Bbox
26                 pt=np.int16(x1,y1)
27                 pt=np.int16(x2,y2)
28                 w=np.int16((x2-x1)/2)
29                 w=np.int16((y2-y1)/2)
30                 objctClass=Class[class]
31                 cv2.rectangle(img, pt, pt2, (0,0,255), 2)
32                 text=objctClass+ " "+str(round(score,2))

```



```

33         cv2.putText(img, text, pt, cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0,255,0), 2)
34         # cv2.putText(img, str(w)+", "+str(h), (w, h), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0,0,0), 2)
35
36         elapsedTime=time.time()-now
37         fps=round(1/elapsedTime, 2)
38         cv2.putText(img, "FPS: "+ str(fps), (10,10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0,0,20), 2)
39         cv2.imshow("Deteksi Objek Jalur", img)
40
41         if cv2.waitKey(1) && ord("q"):
42             break
43
44
45 cv2.waitKey(1)

```

## RIWAYAT PENULIS



Ricky Steven Chandra adalah nama penulis laporan tugas akhir ini. Lahir pada tanggal 24 Agustus 2001 di Karawang Provinsi Jawa Barat. Penulis merupakan Anak ke 3 dari 3 bersaudara, dari pasangan (Alm) Edy Chandra dan Ipah Hanipah. Penulis pertama kali masuk ke pendidikan di SD Negeri 1 Karawang Barat pada tahun 2008 dan tamat 2013, pada tahun yang sama penulis melanjutkan ke SMP Negeri 2 Karawang Barat pada tahun 2013 dan tamat pada tahun 2015. Setelah tamat di SMP, penulis melanjutkan ke SMA Negeri 1 Karawang Barat pada tahun 2016 dan tamat pada tahun 2019. Dan pada tahun yang sama penulis terdaftar sebagai Mahasiswa di Universitas Buana Perjuangan Karawang.

Dengan ketekunan, motivasi tinggi untuk terus belajar dan berusaha. Penulis telah berhasil menyelesaikan pengerjaan tugas laporan tugas akhir ini. Semoga dengan penulisan laporan tugas akhir ini mampu memberikan kontribusi positif bagi dunia pendidikan.

Akhir kata penulis mengucapkan rasa syukur yang sebesar-besarnya atas terselesaikannya tugas akhir yang berjudul “Deteksi Telur Bebek Fertil dan Inferti Menggunakan Algoritma Convolutional Neural Network (CNN)”.