

**DETEKSI DAN KLASIFIKASI JALAN RUSAK DI BANDAR LAMPUNG  
MENGUNAKAN YOLOv8**

**(Skripsi)**

**Oleh**

**RENDY FITRA ADI PRATAMA  
NPM 1917051041**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS LAMPUNG  
BANDAR LAMPUNG  
2024**

**DETEKSI DAN KLASIFIKASI JALAN RUSAK DI BANDAR LAMPUNG  
MENGUNAKAN YOLOv8**

**Oleh**

**RENDY FITRA ADI PRAATAMA**

**Skripsi**

**Sebagai Salah Satu Syarat untuk Mencapai Gelar  
SARJANA KOMPUTER**

**Pada**

**Fakultas Matematika dan Ilmu Pengetahuan Alam  
Universitas Lampung**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS LAMPUNG  
BANDAR LAMPUNG  
2024**

## **ABSTRAK**

### **DETEKSI DAN KLASIFIKASI JALAN RUSAK DI BANDAR LAMPUNG MENGUNAKAN YOLOv8**

**Oleh**

**RENDY FITRA ADI PRATAMA**

Jalan merupakan salah satu infrastruktur yang sangat penting dalam berbagai sektor kehidupan manusia. Jalan sering kali mengalami kerusakan yang diakibatkan oleh kesalahan konstruksi ataupun pemakaian yang tidak sesuai standar, permasalahan berupa kerusakan jalan sering terjadi di kota Bandar Lampung kurangnya tenaga ahli dan sulitnya keterjangkauan menyebabkan banyak kerusakan jalan yang sulit untuk diketahui oleh pihak pemerintah kota. Penelitian ini bertujuan untuk membuat sebuah model deteksi objek menggunakan algoritma *Convolutional Neural Network* (CNN) yang merupakan teknologi *Computer Vision* (CV). Metode yang akan digunakan adalah *You Only Look Once* v8 (YOLOv8) yang akan digunakan untuk melakukan deteksi objek pada kerusakan jalan dengan menggunakan berbagai ukuran gambar saat melakukan proses *training* pada model. Model yang dihasilkan memiliki nilai mAP sebesar 87.1% dan akurasi 86.7% setelah diimplementasikan ke dalam aplikasi *mobile*. Ini memungkinkan deteksi secara *realtime* melalui kamera ponsel bahkan tanpa koneksi internet.

Kata Kunci: Jalan Rusak, Kecerdasan Buatan, Convolutional Neural Network, You Only Look Once v8, Mobile.

## **ABSTRACT**

### **DETECTION AND CLASSIFICATION OF ROADS DAMAGED IN BANDAR LAMPUNG USING YOLOv8**

**By**

**RENDY FITRA ADI PRATAMA**

Road is one of the most important infrastructures in various sectors of human life. Roads often experience damage caused by construction errors or usage that is not in accordance with standards, problems in the form of road damage often occur in the city of Bandar Lampung, the lack of experts and the difficulty of affordability causes a lot of road damage that is difficult to know by the city government. This research aims to create an object detection model using the Convolutional Neural Network (CNN) algorithm which is a Computer Vision (CV) technology. The method to be used is You Only Look Once v8 (YOLOv8) which will be used to perform object detection on road damage using various image sizes when performing the training process on the model. The resulting model has a mAP value of 87.1% and an accuracy of 86.7% after being implemented into a mobile application. This allows realtime detection via cell phone camera even without internet connection.

**Keywords:** Road, Damaged Road, Artificial Intelligence, Convolutional Neural Network, You Only Look Once v8, Android, Mobile.

Judul Skripsi : **DETEKSI DAN KLASIFIKASI JALAN  
RUSAK DI BANDAR LAMPUNG  
MENGUNAKAN yoloV8**

Nama Mahasiswa : **Rendy Fitra Adi Pratama**

Nomor Pokok Mahasiswa : **1917051041**

Program Studi : **S1 Ilmu Komputer**

Jurusan : **Ilmu Komputer**

Fakultas : **Matematika dan Ilmu Pengetahuan Alam**

**MENYETUJUI**

1. **Komisi Pembimbing**

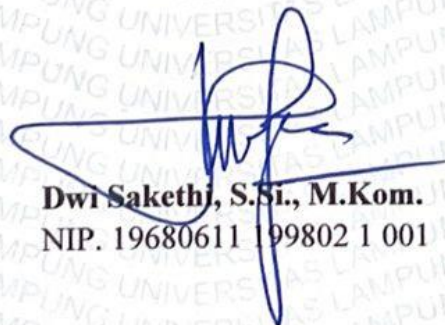


**Dr. rer. nat. Akmal Junaidi, M.Sc.**  
NIP. 19710129 199702 1 001



**Ridho Sholehurrohman, M.Mat.**  
NIK. 232111970128101

2. **Ketua Jurusan Ilmu Komputer**



**Dwi Sakethj, S.Si., M.Kom.**  
NIP. 19680611 199802 1 001

**MENGESAHKAN**

**1. Tim Penguji**

**Ketua Penguji : Dr. rer. nat. Akmal Junaidi, M.Sc.**



**Sekretaris Penguji : Ridho Sholehurrohman, M.Mat.**



**Penguji Utama : Prof. Admi Syarif, Ph.D**

**2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam**



**Dr. Eng. Heri Satria, S.Si., M.Si.**  
NIP. 197110012005011002

**Tanggal Lulus Ujian Skripsi: 15 Agustus 2024**

## PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Rendy Fitra Adi Pratama

NPM : 1917051041

Dengan ini menyatakan bahwa skripsi saya yang berjudul **“DETEKSI DAN KLASIFIKASI JALAN RUSAK DI BANDAR LAMPUNG MENGGUNAKAN YOLOv8”** merupakan karya saya sendiri, bukan karya orang lain. Semua tulisan yang tertulis dalam skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Jika dikemudian hari terbukti bahwa karya tulis ilmiah saya terbukti hasil menjiplak karya orang lain, maka saya siap menerima sanksi berupa pencabutan gelar yang saya peroleh.

Bandar Lampung, 21 Agustus 2024



Rendy Fitra Adi Pratama  
NPM. 1917051041

## RIWAYAT HIDUP



Lahir pada hari Senin, 18 Juni 2001. Anak pertama dari Bapak Nasir, S. Ag. Dan Ibu Hzami, S. Pd.. Penulis menyelesaikan pendidikan Sekolah Dasar (SD) di SD Negeri 1 Way Empulau Ulu pada tahun 2012, melanjutkan ke bangku pendidikan Madrasah Tsanawiyah (MTs) di MTs Negeri 1 Lampung Barat lulus pada tahun 2016 dan Sekolah Menengah Atas di SMA Negeri 1 Liwa lulus pada tahun 2019.

Pada tahun 2019, penulis terdaftar sebagai mahasiswa Jurusan Ilmu Komputer Universitas Lampung melalui jalur SBMPTN. Kegiatan yang dilakukan penulis selama menjadi mahasiswa sebagai berikut.

1. Menjadi anggota Adapter Himpunan Mahasiswa Jurusan Ilmu Komputer pada periode 2019/2020.
2. Menjadi anggota Kaderisasi Himpunan Mahasiswa Jurusan Ilmu Komputer periode 2019/2020.
3. Menjadi anggota Kaderisasi Himpunan Mahasiswa Jurusan Ilmu Komputer periode 2020/2021.
4. Melaksanakan Kerja Praktik pada bulan Desember periode 2022/2023 di KSP Sehati Makmur Abadi
5. Melakukan KKN (Kuliah Kerja Nyata) di Desa Argopeni Kecamatan Sumberrejo Kabupaten Tanggamus dengan Program Kerja Pelatihan Aplikasi Desa Digital.



## **MOTTO**

“Apa yang memang ditakdirkan untukmu, pasti akan menjadi milikmu, bahkan jika itu berada di bawah dua gunung. Dan apa yang memang tidak ditakdirkan untukmu, tidak akan pernah menjadi milikmu bahkan jika itu berada tepat di antara dua bibirmu.”

**(Imam Al-Ghazali)**

“Kusesui kik pak jaoh kutului kik pak sakik, kik lapah nyak mak muloh mak kheno nyak mak hukhik.”

**(Rendy Fitra Adi Pratama)**

*“Yesterday is history, tomorrow is mystery and today is a gift of God, which is why we call it the present.”*

**(Bill Keane)**

## **PERSEMBAHAN**

Alhamdulillah puji dan syukur kepada ke hadirat Allah SWT atas segala rahmat dan karunia-Nya sehingga dapat menyelesaikan skripsi ini dengan sebaik-baiknya.

Shalawat serta salam selalu saya sanjung agungkan kepada Nabi Muhammad SAW yang syafaatnya selalu senantiasa dinantikan di yaumul akhir kelak.

Kupersembahkan karya ini kepada  
**Kedua Orang Tuaku Tersayang**

Yang selalu mendukung dan mendoakan dalam setiap langkah yang saya lalui.  
Saya ucapkan terimakasih sebesar-besarnya atas kasih sayang, didikan dan pengorbanan yang telah diberikan kepada saya dan tak akan mungkin bisa terbalaskan.

**Seluruh Keluarga Besar Ilmu Komputer 2019**

**Jurusan Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan  
Alam, Universitas Lampung**

## SANWACANA

Alhamdulillah, puji syukur kehadirat Allah SWT, atas berkat rahmat, karunia, hidayah, dan kesehatan yang diberikan sehingga penulis dapat menyelesaikan skripsi yang berjudul “DETEKSI DAN KLASIFIKASI JALAN RUSAK DI BANDAR LAMPUNG MENGGUNAKAN YOLOv8”. Dalam penyusunan skripsi ini, penulis mendapatkan semangat, dukungan, dan bantuan dari berbagai pihak. Dalam melaksanakan penelitian dan pembuatan skripsi ini, sehingga pada kesempatan ini penulis ingin menyampaikan ungkapan terima kasih ini kepada

1. Kedua orang tua, Nasir, S. Ag. Dan Hazami, S. Pd. serta keluarga yang telah memberi dukungan dan motivasi kepada penulis untuk menyelesaikan skripsi dengan baik.
2. Bapak Dr. rer. nat. Akmal Junaidi, M.Sc. selaku Dosen Pembimbing 1 yang telah membimbing sehingga penulis dapat menyelesaikan skripsi dengan baik.
3. Bapak Ridho Sholehurrohman, M. Mat. selaku Dosen Pembimbing 2 yang telah membimbing serta memberi masukan dalam proses pembuatan skripsi.
4. Bapak Prof. Admi Syarif, Ph.D. selaku Dosen Pembahas yang telah memberikan masukan dalam penelitian skripsi ini.
5. Bapak Dr. Eng. Heri Satria, S.Si., M.Si. selaku Dekan FMIPA Universitas Lampung.
6. Bapak Dwi Sakethi, S.Si., M.Kom. selaku Ketua Jurusan Ilmu Komputer Universitas Lampung.

7. Ibu Yohana Tri Utami, M. Kom. selaku Dosen Pembimbing Akademik.
8. Bapak dan Ibu Dosen Jurusan Ilmu Komputer FMIPA Universitas Lampung yang telah memberikan ilmu serta pengalaman semasa perkuliahan.
9. Ibu Ade Nora Maela dan seluruh staf di Jurusan Ilmu Komputer yang telah sabar membantu segala urusan administrasi di masa perkuliahan.
10. Teman-teman Jurusan Ilmu Komputer Universitas Lampung angkatan 2019 yang senantiasa memberikan dukungan dan telah berjuang bersama menjalankan studi perkuliahan.
11. Teman-teman Pemuda Hilang Arah selaku rekan seperjuangan dan teman-teman yang sangat berarti di bangku perkuliahan.
12. Teman-teman penghuni *House of Gintoki* selaku sahabat sekaligus keluarga bagi penulis.
13. Teman-teman Grup Kebanyakan Pola yang menjadi teman seperjuangan dalam memperjuangkan nilai mata kuliah yang mengulang.

Bandar Lampung, 15 Agustus 2024



Rendy Fitra Adi Pratama  
NPM.1917051041

## DAFTAR ISI

	<b>Halaman</b>
<b>DAFTAR ISI</b> .....	<b>i</b>
<b>DAFTAR GAMBAR</b> .....	<b>iii</b>
<b>DAFTAR TABEL</b> .....	<b>v</b>
<b>I. PENDAHULUAN</b> .....	<b>1</b>
1.1. Latar Belakang .....	1
1.2. Rumusan masalah.....	4
1.3. Batasan Masalah.....	5
1.4. Tujuan Penelitian.....	5
1.5. Manfaat Penelitian.....	5
<b>II. TINJAUAN PUSTAKA</b> .....	<b>7</b>
2.1. Penelitian Terdahulu.....	7
2.2. Jalan.....	8
2.3. Citra Digital .....	12
2.4. <i>Convolutional Neural Network</i> (CNN) .....	13
2.4.1. <i>You Only Look Once</i> v8 (YOLOv8).....	15
2.4.2. <i>Hyperparameter</i> .....	18
2.5. <i>Mean Average Precision</i> (mAP) .....	19
2.5.1. <i>Intersection over Union</i> (IoU).....	20
2.6. Aplikasi <i>Mobile</i> .....	21
2.7. Dart.....	21
2.8. <i>Framework flutter</i> .....	21
2.9. Visual Studio Code.....	22
2.10. Python.....	22
2.11. Roboflow .....	23

2.12. Google Colab.....	23
2.13. <i>Confusion Matrix</i> .....	24
2.13.1. <i>Precision</i> .....	24
2.13.2. <i>Recall</i> .....	25
2.13.3. <i>Accuracy</i> .....	25
2.13.4. <i>F1-Score</i> .....	26
<b>III. METODOLOGI PENELITIAN.....</b>	<b>27</b>
3.1. Waktu dan Tempat Penelitian .....	27
3.2. Perangkat Penelitian .....	28
3.2.1. Perangkat Lunak ( <i>Software</i> ).....	28
3.2.2. Perangkat Keras ( <i>Hardware</i> ) .....	28
3.3. Bahan.....	29
3.4. Tahapan Penelitian .....	29
3.4.1. Pengumpulan Data .....	30
3.4.2. <i>Pre-Processing Images</i> .....	30
3.4.3. <i>Training and Evaluate Model YOLOv8</i> .....	31
3.4.4. <i>Deploy Model to Mobile Application</i> .....	32
<b>IV. PEMBAHASAN .....</b>	<b>36</b>
4.1. Pengumpulan Data .....	36
4.2. <i>Pre-Processing Images</i> .....	36
4.2.1. <i>Images Annotation</i> .....	37
4.2.2. <i>Splitting Images Datasets</i> .....	37
4.3. <i>Training and Evaluate Model YOLOv8</i> .....	38
4.4. <i>Converting Model to Mobile Model</i> .....	41
4.5. <i>Deploy Model to Mobile</i> .....	42
4.5.1. Tampilan <i>User Interface</i> Aplikasi.....	43
4.5.2. <i>Testing Performa</i> Aplikasi .....	46
<b>V. KESIMPULAN DAN PEMBAHASAN .....</b>	<b>52</b>
5.1. Kesimpulan.....	52
5.2. Saran.....	52
<b>DAFTAR PUSTAKA .....</b>	<b>54</b>

## DAFTAR GAMBAR

Gambar 1. Jalan Mulus (Indonesia.go.id, 2024) .....	8
Gambar 2. <i>Alligator Crack</i> (Olahan Pribadi).....	9
Gambar 3. <i>Longitudinal Crack</i> (Wang et al., 2023). .....	10
Gambar 4. <i>Transversal Crack</i> (Khare et al., 2023).....	11
Gambar 5. <i>Potholes</i> (Jiang et al., 2023).....	12
Gambar 6. Citra Digital (Jumadi et al., 2021).....	13
Gambar 7. Struktur Proses Pada Algoritma CNN (Putra, 2023).....	14
Gambar 8. Cara Kerja CNN(Simonyan & Zisserman, 2015) .....	15
Gambar 9. Arsitektur YOLOv8 (Khare et al., 2023) .....	16
Gambar 10. Arsitektur CSPDarknet53 (Khare et al., 2023) .....	17
Gambar 11. FPN (Wang et al., 2023) .....	18
Gambar 12. <i>Intersection over Union</i> (Brilliant et al., 2018).....	20
Gambar 13. Tahapan Penelitian .....	29
Gambar 14. Prototipe <i>Home Screen</i> .....	33
Gambar 15. Prototipe Halaman <i>Realtime Detection</i> .....	33
Gambar 16. Prototipe Halaman <i>Image Detection</i> .....	34
Gambar 17. <i>Annotation Dataset</i> .....	37
Gambar 18. <i>Splitting Datasets</i> .....	37
Gambar 19. <i>Images Size 160x160</i> .....	38
Gambar 20. <i>Image Size 320x320</i> .....	38
Gambar 21. <i>Image Size 480x480</i> .....	39
Gambar 22. <i>Image Size 640x640</i> .....	39
Gambar 23. Perbandingan Precision, Recall dan mAP Tiap Model Berdasarkan Image Size.....	39

Gambar 24. Grafik Hasil Proses <i>Training</i> .....	40
Gambar 25. Perbandingan Label dan Prediksi Model .....	41
Gambar 26. Hasil Konversi Model ke .tflite .....	42
Gambar 27. Tampilan <i>Home Screen</i> .....	43
Gambar 28. Halaman <i>In Frame Detection</i> .....	44
Gambar 29. Halaman <i>Upload Images</i> .....	45
Gambar 30. Halaman Hasil <i>Images Detection</i> .....	46
Gambar 31. Contoh <i>Output</i> Jalan Berlubang .....	47
Gambar 32. Contoh <i>Output</i> Jalan Retak .....	48



## DAFTAR TABEL

Tabel 1. Penelitian Terdahulu .....	7
Tabel 2. <i>Confusion Matrix</i> .....	24
Tabel 3. Waktu Penelitian .....	27
Tabel 4. Komposisi <i>Dataset</i> .....	29
Tabel 5. <i>Hyperparameter</i> .....	31
Tabel 6. Kebutuhan Fitur .....	32
Tabel 7. Skenario Pengujian <i>Home Screen</i> .....	35
Tabel 8. Skenario Pengujian <i>Realtime Detection</i> .....	35
Tabel 9. Skenario Pengujian <i>Image Detection</i> .....	35
Tabel 10. Detail <i>Dataset</i> .....	36
Tabel 11. <i>Average Precision</i> dan <i>Average Recall</i> .....	40
Tabel 12. Ringkasan Hasil Pengujian Menggunakan <i>Data Test</i> .....	49
Tabel 13. Ringkasan Hasil Pengujian di Lapangan .....	50

## I. PENDAHULUAN

### 1.1. Latar Belakang

Infrastruktur merupakan sebuah fasilitas fisik, perangkat keras, dan perangkat lunak yang diperlukan untuk pelayanan kepada masyarakat serta dapat mendukung pertumbuhan Ekonomi (Afriani, 2014). Jenis Infrastruktur Fisik yang sering kita temui dalam kehidupan sehari-hari contohnya bangunan Perkantoran, gedung Rumah Sakit, bangunan Sekolah, Pasar, Jalan Raya, dan lain-lain (Afriyana *et al.*, 2023). Dalam kehidupan sehari-hari keberadaan Infrastruktur yang memadai dapat memudahkan kita dalam melakukan aktivitas. Infrastruktur sangat dibutuhkan dalam pertumbuhan dalam sektor Ekonomi, Pendidikan, Sosial, Politik dan banyak sektor lain yang perkembangannya sangat bergantung pada berbagai infrastruktur yang ada oleh karena itu, penanganan dan perawatan serta antisipasi dini terhadap kerusakan sangat diperlukan.

Infrastruktur sangat berperan penting dalam kehidupan masyarakat, dari banyaknya infrastruktur yang ada. Jalan adalah salah satu infrastruktur yang berperan penting dalam pertumbuhan tiap-tiap sektor kehidupan masyarakat. Jalan merupakan Infrastruktur yang sangat riskan akan adanya kerusakan karena penggunaan yang terus menerus serta sulitnya keterjangkauan dalam melakukan perawatan (Rawansyah *et al.*, 2020). Jalan merupakan salah satu Infrastruktur yang berperan penting, maka dari itu Jalan harus bisa menyediakan kenyamanan, kelancaran dan keamanan kepada para penggunanya (Mukhtar *et al.*, 2022), Salah satu masalah yang kerap dihadapi oleh Pemerintah Kota Bandar Lampung adalah semakin banyaknya jalan yang rusak sehingga sulit untuk mengetahui lokasi dan prioritas perbaikan jalan rusak di Kota Bandar Lampung.

Metode deteksi dan klasifikasi manual mengenai kondisi suatu ruas jalan dapat diketahui dengan menggunakan nilai IRI (*International Roughness Index*). IRI merupakan nilai ketidakrataan permukaan jalan dan ditentukan oleh kumulatif panjang undulasi permukaan jalan per satuan Panjang (Umi *et al.*, 2016) sebagaimana telah ditetapkan pada Permen PUPR Nomor 3 Tahun 2019. Metode deteksi dan klasifikasi secara otomatis dapat dilakukan dengan banyak cara, dalam pelaksanaannya dibutuhkan teknologi yang dapat menunjang proses deteksi dan klasifikasi tersebut. Dengan perkembangan zaman yang sangat pesat terutama pada bidang Teknologi dan Informasi yang dapat membantu dalam identifikasi objek salah satunya adalah *Convolutional Neural Network* (CNN) yang merupakan salah satu teknologi *Computer Vision* (CV). CNN menggunakan metode pembelajaran *Deep Learning* hal tersebut dapat dimanfaatkan untuk membantu dalam proses deteksi dan klasifikasi jalan rusak di Kota Bandar Lampung secara otomatis.

Proses klasifikasi jalan rusak dapat dilakukan dengan pengamatan menggunakan mata dan hal ini lebih efisien dibandingkan menggunakan sistem pendeteksian objek namun dalam prakteknya kurangnya tenaga ahli untuk melakukan klasifikasi secara akurat sehingga sistem deteksi dan klasifikasi objek dibutuhkan sebagai alat bantu untuk melakukan hal tersebut, *You Only Look Once* adalah salah satu metode yang dapat digunakan untuk melakukan deteksi dan klasifikasi objek.

*You Only Look Once* (YOLO) adalah sekumpulan model pendeteksian objek yang telah menjadi model pendeteksian objek tercepat untuk pendeteksian waktu nyata sejak diperkenalkan pada tahun 2015 (Eriksson, 2023). Dalam praktiknya metode *You Only Look Once* (YOLO) terbukti sebagai metode yang jauh lebih efisien dibandingkan dengan metode *Computer Vision* yang lain seperti SSD, R-CNN, *Faster R-CNN*, dan lain-lain (Wang *et al.*, 2023). Sebagai salah satu algoritma pendeteksian klasik, algoritma YOLO telah berevolusi menjadi YOLOv8, yang menawarkan keuntungan signifikan dalam hal akurasi dan kecepatan pendeteksian pada objek secara *realtime* (Pasaribu *et al.*, 2023), dimana kecepatan pendeteksian ini mempengaruhi jumlah *frame* yang dapat diolah sebuah algoritma pendeteksian. Dalam penelitian yang dilakukan oleh (Sharma *et al.*, 2022) yang membahas

tentang pendeteksian pada media video dengan membandingkan dua buah algoritma berbeda yaitu SVM dan YOLO, algoritma YOLO dapat menghasilkan *detection frame* sebesar 45 fps sedangkan SVM hanya menghasilkan 2 fps pada pendeteksian.

Saat melakukan pengamatan di lapangan panjang jalan yang diamati bisa saja berkilo-kilometer sehingga sulit jika hanya mengandalkan mata manusia, selain itu faktor mobilitas manusia yang menyebabkan keterbatasan dalam menjangkau lokasi yang jauh dalam waktu yang relatif berdekatan oleh karena itu dibutuhkan sistem yang dapat membantu sekaligus menggantikan peran manusia untuk melakukan pendeteksian dan pengamatan jalan rusak. YOLO mampu menganalisis gambar secara cepat dan mengidentifikasi berbagai jenis kerusakan jalan dalam waktu nyata, tanpa dipengaruhi oleh faktor-faktor seperti kelelahan atau variasi subjektif dalam penilaian. Ini memungkinkan proses deteksi dan klasifikasi kerusakan dilakukan secara konsisten, terutama saat menangani volume data yang besar atau dalam kondisi yang tidak ideal, hal ini membuat YOLO sangat baik untuk diterapkan menjadi alat bantu untuk melakukan pendeteksian dan klasifikasi jalan rusak di Bandar Lampung.

Berdasarkan Penelitian yang dilakukan oleh (Wang *et al.*, 2023) dengan menggunakan *dataset* RDD2022, algoritma YOLOv8 menghasilkan nilai akurasi yang cukup tinggi yaitu 90,7% mengungguli metode lain seperti SSD yang hanya menghasilkan nilai akurasi di angka 72,7% dan *Faster* R-CNN yang hanya menghasilkan nilai akurasi sebesar 73,2%. Dalam penelitian lain oleh (Widiyanto, 2023) dengan menggunakan *dataset* RDD2022 algoritma YOLOv8 menghasilkan nilai akurasi sebesar 96%. Pada penelitian yang dilakukan oleh (Khare *et al.*, 2023) yang bertujuan membandingkan *processing time* terhadap beberapa algoritma, YOLOv8 membutuhkan *processing time* sebesar 8,8 ms dimana angka tersebut memiliki perbedaan yang sangat signifikan dibandingkan algoritma lain dimana YOLOv5 membutuhkan waktu hingga 38 ms dan YOLOv7 membutuhkan waktu hingga 35 ms.

Menurut penelitian lain yang dilakukan oleh (Pratama, 2021) algoritma R-CNN menghasilkan kecepatan deteksi pada video hanya pada angka 50s, nilai ini sangat jauh lebih lambat dibandingkan dengan kecepatan deteksi pada video yang dihasilkan oleh algoritma YOLO pada penelitian sebelumnya, Algoritma *fast* R-CNN hanya menghasilkan kecepatan deteksi hanya sebesar 2s dan *faster* R-CNN menghasilkan kecepatan deteksi paling baik dari kedua algoritma deteksi sebelumnya dengan kecepatan 200ms yang dimana nilai tersebut masih jauh lebih lambat apabila dibandingkan dengan kecepatan deteksi yang dihasilkan oleh algoritma pada penelitian sebelumnya, berdasarkan penjelasan yang telah dijabarkan di atas maka dalam penelitian ini akan digunakan YOLOv8 untuk melakukan identifikasi dan klasifikasi jalan rusak Kota Bandar Lampung.

Dalam penelitian ini sistem akan di *training* dengan menggunakan *dataset* yang berisikan data kerusakan jalan yang bertujuan untuk mendeteksi kerusakan jalan berupa lubang ataupun retak. Testing pada sistem dilakukan secara *realtime* pada objek kerusakan jalan di Kota Bandar Lampung menggunakan model yang telah di *deploy* ke dalam aplikasi *mobile*. Evaluasi dilakukan dengan menggunakan dengan menggunakan *Confusion Matrix*, variabel pengukuran yang akan digunakan adalah *accuracy*, *precision*, *recall*, dan *F1-Score*.

## 1.2. Rumusan masalah

Berdasarkan latar belakang rumusan masalah pada penelitian ini adalah :

1. Bagaimana implementasi metode YOLOv8 untuk melakukan deteksi dan klasifikasi jalan rusak Kota Bandar Lampung berbasis *mobile*.
2. Bagaimana Tingkat keakuratan metode YOLOv8 dalam mendeteksi dan mengklasifikasikan kerusakan jalan di Kota Bandar Lampung berbasis *mobile*.

### 1.3. Batasan Masalah

Berdasarkan latar belakang yang telah dijabarkan, maka penelitian ini dibatasi pada:

1. Metode yang digunakan adalah YOLOv8 dengan menggunakan bahasa pemrograman Python 3.10.12.
2. Input yang digunakan berupa gambar citra digital yang berisi objek antara lain : Jalan Protokol, Jalan Lintas, Jalan Perumahan.
3. Sistem akan di-*training* menggunakan *dataset* yang berisikan dua kelas yaitu : berlubang dan retak.
4. Model YOLOv8 yang telah dibuat akan dilakukan *deploy* ke dalam aplikasi *mobile* sederhana yang hanya berfokus kepada pendeteksian objek.

### 1.4. Tujuan Penelitian

Adapun tujuan penelitian ini adalah sebagai berikut :

1. Melakukan deteksi dan klasifikasi jalan rusak Kota Bandar Lampung dengan mengimplementasikan metode YOLOv8.
2. Mengetahui akurasi metode YOLOv8 dalam mendeteksi dan mengklasifikasikan jalan rusak Kota Bandar Lampung.
3. Mengimplementasikan model YOLOv8 pada aplikasi sederhana yang berbasis *mobile*.
4. Mengetahui apakah *image size* yang digunakan saat melakukan *training* pada model dapat berpengaruh pada akurasi dari model YOLOv8.

### 1.5. Manfaat Penelitian

Adapun manfaat penelitian ini adalah sebagai berikut:

1. Sebagai sumber informasi terkait pengembangan model *Machine Learning* khususnya metode YOLOv8.
2. Sebagai rujukan untuk mengetahui jenis kerusakan jalan untuk penentuan prioritas perbaikan jalan rusak Kota Bandar Lampung.
3. Penelitian ini dapat menjadi sumber informasi terkait pengembangan

aplikasi *mobile* Android khususnya untuk *Framework Flutter* pada Jurusan Ilmu Komputer.

## II. TINJAUAN PUSTAKA

### 2.1. Penelitian Terdahulu

Penelitian terdahulu akan digunakan sebagai pembandingan bagi penelitian ini, penelitian terdahulu akan menjadi acuan untuk penelitian yang akan dilakukan. Penelitian terdahulu tersebut yaitu :

Tabel 1. Penelitian Terdahulu.

No.	Judul Penelitian	Metode	Hasil Penelitian
1.	Deteksi dan Klasifikasi Kerusakan Jalan Aspal Menggunakan Metode YOLO Berbasis Citra Digital (Pramestya, 2018).	YOLO	Sistem pendeteksian kerusakan jalan aspal dengan algoritma YOLOv1- <i>tiny</i> .
2.	BL-YOLOv8: <i>An Improved Road Defect Detection Model Based on YOLOv8</i> (Wang <i>et al.</i> , 2023).	YOLOv8	Sistem deteksi jalan rusak dengan algoritma YOLOv8 pada <i>dataset</i> RDD2022.
3.	<i>YOLOv8-Based Visual Detection of Road Hazards: Potholes, Sewer Covers, and Manholes</i> (Khare <i>et al.</i> , 2023).	YOLOv8	Model deteksi YOLOv8 untuk mendeteksi <i>Potholes, Sewer Cover, dan Manholes</i> .
4.	<i>Automated Pavement Defect Detection Using YOLOv8 Object Detection</i> (Widiyanto, 2023).	YOLOv8	Sistem deteksi kecacatan pada trotoar dengan algoritma YOLOv8.



No.	Judul Penelitian	Metode	Hasil Penelitian
5.	<i>Road damage detection with Yolov8 on Swedish roads</i> <i>Civilingenjörprogrammet i informationsteknologi</i> (Eriksson, 2023).	YOLOv5, YOLOv6, YOLOv8	Mengungguli kedua metode lainnya dalam mendeteksi kerusakan jalan di Swedia.

## 2.2. Jalan

Jalan merupakan komponen krusial dalam kehidupan masyarakat, jalan diperuntukkan sebagai perlintasan kendaraan seperti mobil, motor, truk, bus, becak, dan lain-lain, hal tersebut sangat berpengaruh terhadap perkembangan ekonomi, sosial, dan pembangunan suatu daerah (Adityah, 2021; Bertarina *et al.*, 2022). Jalan yang baik adalah jalan yang dapat memenuhi kebutuhan fungsional dan standar struktural sehingga dapat dipergunakan sebagaimana mestinya, naiknya infrastruktur jalan sangat berpengaruh terhadap kelancaran transportasi dan pertumbuhan ekonomi serta dapat mendukung sektor lain seperti pendidikan, sosial dan politik.



Gambar 1. Jalan Mulus (Indonesia.go.id, 2024)

Jalan merupakan salah satu komponen infrastruktur yang riskan akan adanya kerusakan, terdapat berbagai jenis kerusakan pada jalan. Menurut (Adityah, 2021) kerusakan jalan secara umum dapat dibedakan menjadi 2 yaitu kerusakan struktural dan juga kerusakan fungsional. Kerusakan struktural merupakan kerusakan yang melibatkan satu atau lebih komponen perkerasan jalan atau disebut juga dengan kegagalan perkerasan. Kerusakan fungsional adalah kerusakan yang berdampak pada keamanan dan kenyamanan pengguna jalan (Hidayat, 2020). Di bawah ini adalah beberapa kerusakan fungsional jalan :

a. *Crack*

*Crack* atau retakan adalah jenis kerusakan jalan yang berupa pecahnya bagian perkerasan jalan yang disebabkan oleh berbagai faktor seperti beban yang berlebihan atau kurangnya lapisan perkerasan pada ruas jalan. *Crack* memiliki beberapa jenis yaitu :

- *Alligator Crack*



Gambar 2. *Alligator Crack* (Olahan Pribadi).

*Alligator Crack* (Retak Kulit Buaya) adalah kondisi dimana retakan yang terjadi berbentuk jaring-jaring menyerupai kulit buaya pada sebagian atau keseluruhan permukaan ruas jalan, retak jenis ini memiliki penyebab berupa beban lalu lintas yang berlebihan secara terus-menerus, selain itu bisa juga disebabkan oleh kesalahan pada konstruksi jalan (Adityah, 2021). Selengkapnya mengenai *Alligator Crack* dapat dilihat pada Gambar 1.

- *Longitudinal Crack*

*Longitudinal Crack* (Retak Membujur) adalah jenis kerusakan jalan yang berupa keretakan pada ruas jalan dimana retakan tersebut memanjang searah dengan arus dari ruas jalan, sama seperti jenis kerusakan jalan yang sebelumnya keretakan ini disebabkan oleh beban lalu lintas yang berlebihan secara berkesinambungan (Sasmito *et al.*, 2023). Bentuk dari *Longitudinal Crack* dapat dilihat langsung pada Gambar 2.



Gambar 3. *Longitudinal Crack* (Wang *et al.*, 2023).

- *Transversal/Lateral Crack*



Gambar 4. *Transversal Crack* (Khare *et al.*, 2023)

*Transversal/Lateral Crack* (Retak Melintang) merupakan kondisi dimana terdapat keretakan yang membentang dari satu sisi ke sisi lain jalan, hal ini dapat disebabkan oleh beban lalu lintas yang melebihi kapasitas yang seharusnya atau dapat juga disebabkan oleh permukaan perkerasan jalan yang tidak sesuai dengan ketentuan yang seharusnya (Widiyanto, 2023). Mengenai *Transversal Crack* dapat dilihat melalui Gambar 5.

b. *Potholes*

*Potholes* (Lubang) kerusakan jenis ini berbentuk seperti mangkok dan dapat menampung dan menyerap air pada badan jalan. Kerusakan jenis ini biasanya disebabkan oleh kadar aspal pada lapisan perkerasan terlalu sedikit sehingga bisa

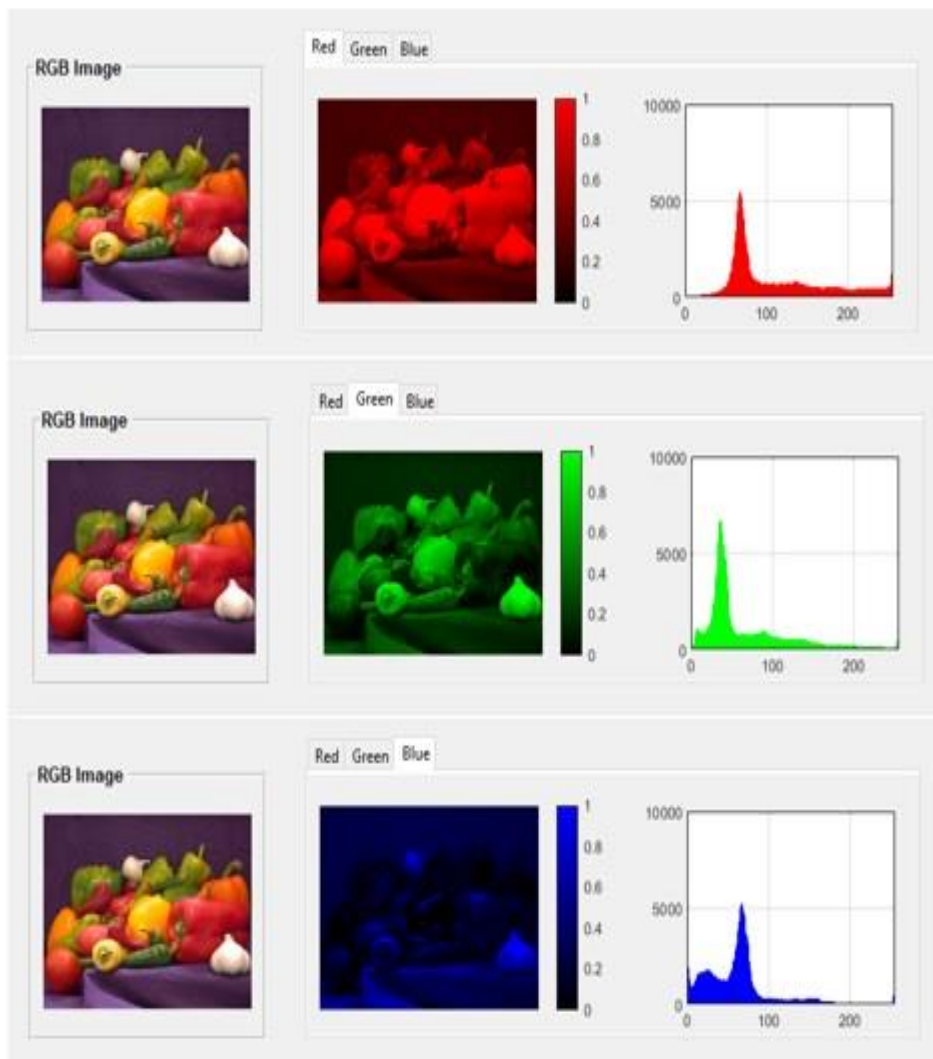
menyebabkan lapisan agregat terlepas saat tergenang oleh air secara terus – menerus (Pramestya, 2018). Lebih detail mengenai *Potholes* dapat dilihat pada Gambar 4.



Gambar 5. *Potholes* (Jiang *et al.*, 2023)

### 2.3. Citra Digital

Menurut (Jumadi *et al.*, 2021) Citra digital merupakan entitas visual yang memainkan peran penting dalam berbagai aspek kehidupan modern, mulai dari hiburan hingga industri teknologi. Citra digital adalah representasi visual dari objek atau *scene* yang direkam dalam format digital, diwakili oleh sejumlah besar piksel dengan nilai-nilai warna atau *grayscale* yang terdefinisi. Keteraturan ini memungkinkan komputer untuk memproses, menyimpan, dan menampilkan gambar secara elektronik dengan presisi yang tinggi .

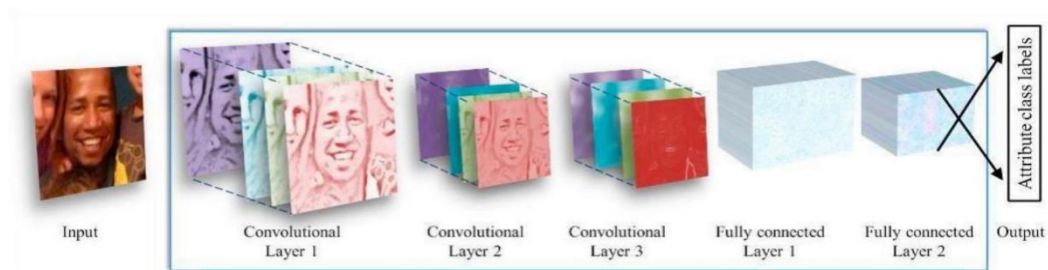


Gambar 6. Citra Digital (Pamungkas, 2024)

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

*Convolutional Neural Network* (CNN) adalah bagian dari jaringan saraf tiruan, dan. Jaringan konvolusional (CNN) adalah jenis jaringan saraf tiruan (neural network) yang dirancang untuk mengolah data gambar. CNN telah menjadi salah satu jenis jaringan saraf tiruan yang paling populer dan sukses, dan telah digunakan untuk berbagai aplikasi, termasuk pengenalan objek, klasifikasi gambar, dan segmentasi gambar. Konsep dasar CNN telah ada sejak lama, tetapi baru pada tahun 1990-an CNN mulai dikembangkan secara serius. Salah satu pelopor pengembangan CNN adalah (Lecun *et al.*, 1998), yang mengembangkan CNN untuk pengenalan karakter

tulisan tangan. Secara luas algoritma ini biasanya digunakan untuk identifikasi suatu objek dan klasifikasi gambar. *Input layer*, *hidden layers*, dan *output layer* adalah bagian terpenting dari *neural network*. Setiap *node* pada algoritma CNN mengambil nilai dari *input layer*, memproses hasil *input*, dan hasil dari proses ini menghasilkan *output*. Gambar yang dijadikan data diolah dengan tumpukan *hidden layer* algoritma CNN dan selanjutnya digunakan sebagai ekstraksi fitur dari gambar untuk menghitung fitur algoritma CNN (Mubarok, 2019).

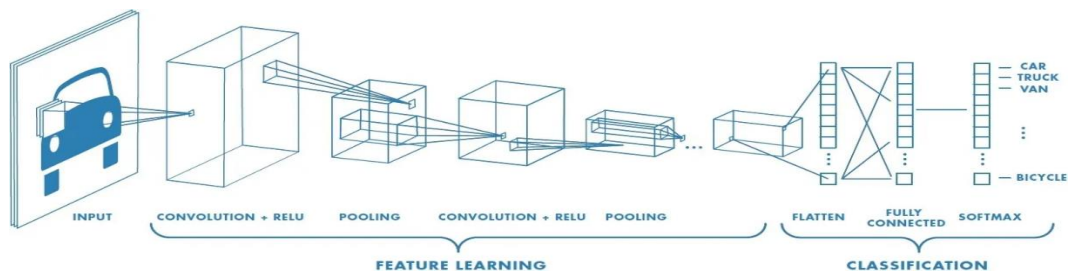


Gambar 7. Struktur Proses Pada Algoritma CNN (Putra, 2023)

*Convolutional Neural Network* (CNN) adalah arsitektur jaringan saraf yang dirancang khusus untuk tugas pemrosesan dan pengenalan gambar. Dasar-dasar CNN mencakup beberapa komponen utama yang bekerja sama untuk mengekstrak dan memahami fitur dari data *grid* (Jensen *et al.*, 2021). Pertama, *Convolutional layer* menggunakan filter atau kernel untuk melakukan operasi konvolusi pada *input*, memungkinkan jaringan mendeteksi fitur seperti tepi, sudut, dan tekstur. Hasilnya kemudian diteruskan melalui fungsi aktivasi ReLU untuk menghasilkan non-linieritas. *Pooling layer* kemudian digunakan untuk mengurangi dimensi spasial pada representasi gambar, sehingga mengurangi jumlah parameter dan meningkatkan kecepatan komputasi. *Feature map* yang dihasilkan dari proses konvolusi dan *pooling* kemudian dilakukan *flattening* menjadi vektor satu dimensi.

Setelah proses tersebut, vektor hasil *flattening* dihubungkan dengan *Fully Connected Layer*. *Layer* ini menghubungkan setiap *neuron* di satu *layer* dengan setiap *neuron* di *layer* berikutnya, mengubah representasi spasial menjadi representasi yang dapat dihubungkan langsung. Terakhir, *Output Layer* menghasilkan prediksi atau kelas berdasarkan *input* yang diberikan. Dengan demikian, arsitektur CNN memungkinkan pembelajaran fitur-fitur hierarkis dari

data *grid*, seperti struktur hierarkis dalam gambar, yang dapat digunakan untuk tugas-tugas seperti klasifikasi gambar (Simonyan & Zisserman, 2015).



Gambar 8. Cara Kerja CNN(Simonyan & Zisserman, 2015)

#### 2.4.1. *You Only Look Once* v8 (YOLOv8)

YOLOv8 adalah algoritma *object detection* yang dikembangkan oleh (Bochkovskiy et al., 2020). Algoritma ini merupakan pengembangan dari algoritma YOLOv5 yang dirilis pada tahun 2020. YOLOv8 dirancang untuk dapat mendeteksi dan melacak objek dengan cepat dan akurat, bahkan dalam kondisi yang kompleks. Dalam penelitian yang dilakukan oleh (Wang *et al.*, 2023) arsitektur YOLOv8 terdiri dari 3 bagian, yaitu:

a. *Backbone Network*

*Backbone Network* atau Jaringan Tulang Belakang berfungsi untuk mengekstrak fitur dari gambar. *Backbone network* yang digunakan pada YOLOv8 adalah CSPDarknet53.

b. *Neck Network*

*Neck Network* atau Jaringan Leher adalah bagian yang berfungsi untuk menghubungkan *backbone network* dengan *head network*. *Neck network* yang digunakan pada YOLOv8 adalah FPN (*Feature Pyramid Network*).

c. *Head Network*

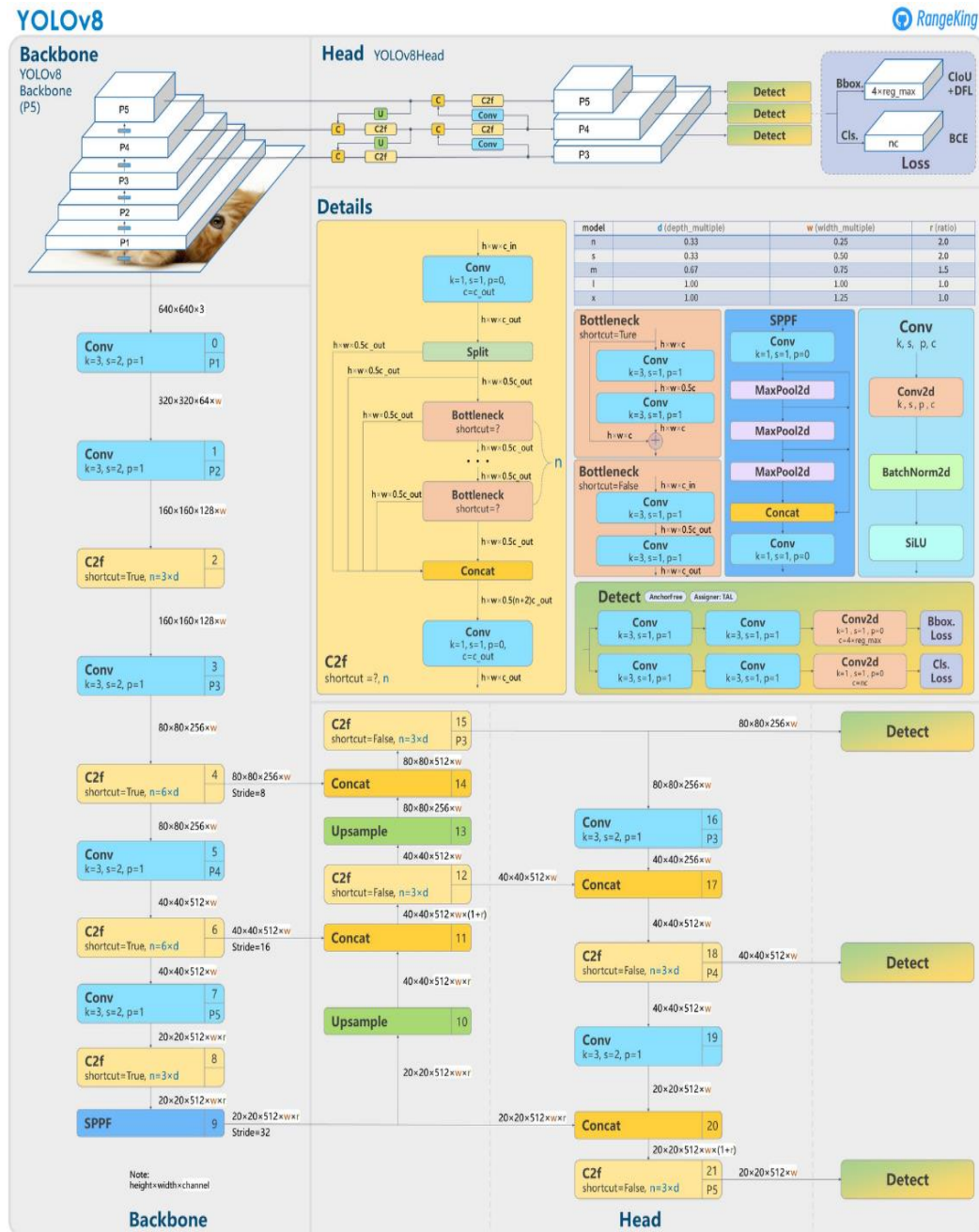
*Head Network* atau jaringan kepala adalah bagian yang berfungsi untuk mendeteksi dan melacak objek. *Head network* yang digunakan pada YOLOv8 terdiri dari 3 bagian, yaitu:

- Bagian pertama adalah YOLOv8-S yang berfungsi untuk mendeteksi objek dengan ukuran kecil.



- Bagian kedua adalah YOLOv8-M yang berfungsi untuk mendeteksi objek dengan ukuran sedang.
- Bagian ketiga adalah YOLOv8-L yang berfungsi untuk mendeteksi objek dengan ukuran besar.

Arsitektur dari YOLOv8 secara keseluruhan dapat dilihat pada Gambar 8.

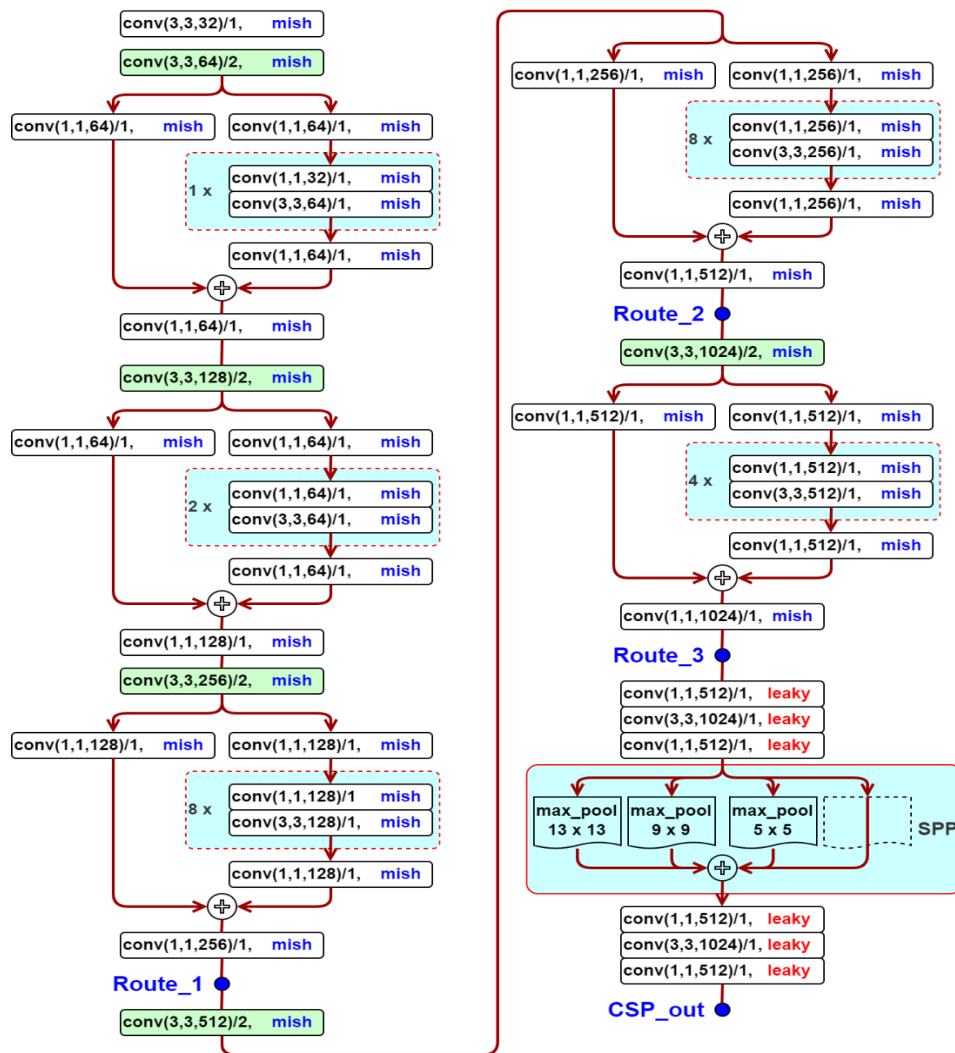


Gambar 9. Arsitektur YOLOv8 (Khare et al., 2023)

Dalam penelitian yang dilakukan oleh (Khare *et al.*, 2023) dijelaskan bahwa dalam pengembangannya YOLOv8 memiliki beberapa peningkatan dibandingkan dengan pendahulunya, beberapa peningkatan yang diberikan kepada YOLOv8 yaitu:

a. CSPDarknet53

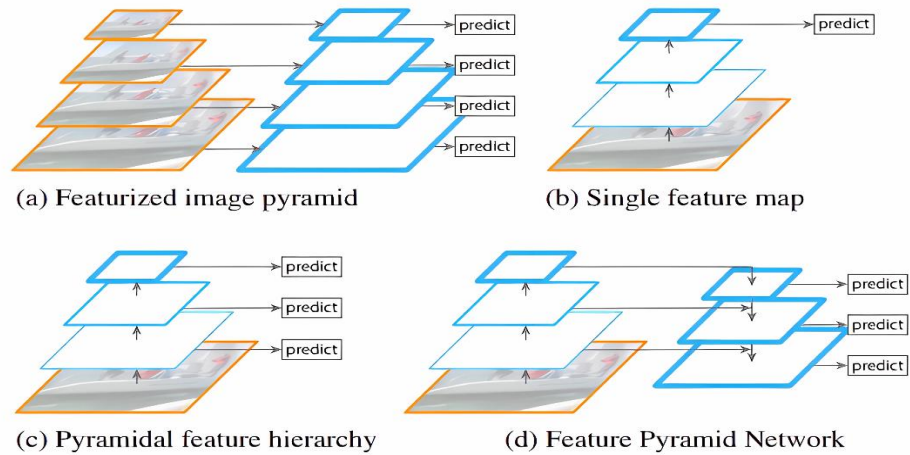
*Backbone network* YOLOv8 menggunakan CSPDarknet53 yang menggantikan Darknet53 yang digunakan pada YOLOv5. CSPDarknet53 memiliki beberapa keunggulan, Seperti kecepatan *processing* yang lebih cepat dan efisien serta memiliki akurasi yang lebih tinggi, selain itu CSPDarknet dapat mengatasi masalah gradien hilang (Khare *et al.*, 2023). Arsitektur CSP Darknet dapat dilihat pada Gambar 10.



Gambar 10. Arsitektur CSPDarknet53 (Khare *et al.*, 2023)

b. FPN (Feature Pyramid Network)

*Neck network* YOLOv8 menggunakan FPN yang menggantikan PAN (*Path Aggregation Network*) yang digunakan pada YOLOv5. Keunggulan FPN adalah dapat mendeteksi objek dengan ukuran yang berbeda-beda (Wang et al., 2023).



Gambar 11. FPN (Wang et al., 2023)

c. *Head*

Merujuk pada Gambar 9 *head network* Yolov8 terdiri dari 3 bagian, yaitu YOLOv8-S digunakan untuk melakukan deteksi terhadap objek yang berukuran kecil, YOLOv8-M digunakan untuk mendeteksi objek yang berukuran sedang, dan YOLOv8-L digunakan untuk melakukan deteksi pada objek yang berukuran besar (Khare et al., 2023).

Pengujian YOLOv8. YOLOv8 telah diuji dengan menggunakan *dataset* COCO 2017. Hasil pengujian menunjukkan bahwa Yolov8 memiliki akurasi yang lebih tinggi dibandingkan dengan YOLOv5, YOLOv8 memiliki *precision* sebesar 94,9%, *recall* sebesar 92,1%, dan *F1 score* sebesar 93,5% (Khare et al., 2023).

#### 2.4.2. *Hyperparameter*

*Hyperparameter* adalah variabel konfigurasi yang berada di luar model yang digunakan untuk membantu kinerja dari model yang akan dibuat (Feurer et al.,

2019). *Hyperparameter* yang digunakan adalah *epoch*, *batch size*, *optimizer* AdamW, dan *Learning rate*.

a. *Epoch*

*Epoch* adalah sebuah parameter yang menentukan jumlah perulangan suatu model mempelajari sebuah *dataset* yang ada. Dalam satu kali *epoch* berarti model telah mempelajari sebuah *dataset* secara keseluruhan sebanyak satu kali (Wasil *et al.*, 2022).

b. *Batch Size*

*Batch size* adalah jumlah dari sampel data pada setiap iterasi. *Batch size* juga dapat mempengaruhi waktu *training* dan akurasi sebuah model, semakin kecil *Batch Size* yang digunakan maka waktu *training* akan semakin lama dan akurasi akan semakin tinggi, sebaliknya jika *Batch Size* yang digunakan semakin besar maka waktu *training* akan semakin cepat namun akurasi akan semakin kecil (Rochmawati *et al.*, 2023).

c. *Optimizer* AdamW

*Optimizer* AdamW bekerja dengan cara melakukan penurunan gradien stokastik yang didasarkan pada estimasi yang bersifat adaptif yang bertujuan untuk meluruhkan bobot (Loshchilov & Hutter, 2019).

d. *Learning rate*

*Learning rate* memiliki sistem kerja dengan melakukan koreksi pada bobot yang ada saat melakukan *training* dengan tujuan mencapai nilai akurasi yang lebih baik (Rochmawati *et al.*, 2023).

## **2.5. Mean Average Precision (mAP)**

*Mean Average Precision* (mAP) adalah sebuah nilai yang menjadi tolak ukur untuk mengetahui seberapa baik sebuah sistem dalam menemukan sesuatu. *Mean Average Precision* menghitung seberapa sering sebuah sistem melakukan deteksi dengan benar dan sebaliknya. Nilai dari *Mean Average Precision* didapat dengan cara menggabungkan nilai *accuracy* dan nilai *recall* (Harun, 2020).

*Mean Average Precision* dapat dihitung dengan menjumlahkan nilai dari *Average Precision* dari setiap kelas kemudian dibagi dengan jumlah kelas yang ada (Brilliant et al., 2018). Rumus dari *Mean Average Precision* adalah sebagai berikut:

$$\text{Mean Average Precision} = \frac{1}{n} \sum_{k=1}^n AP_{k \rightarrow n}$$

Keterangan :

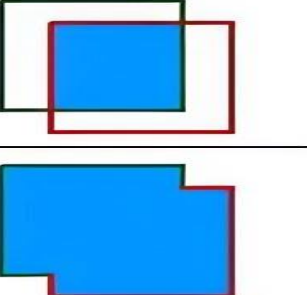
$n$  = Jumlah Kelas

$AP$  = *Average Precision*

*Mean Average Precision* adalah cara terbaik untuk mengukur kinerja sebuah sistem deteksi objek jika dibandingkan dengan *accuracy* dan pengukuran dengan satu IoU.

### 2.5.1. *Intersection over Union (IoU)*

IoU (*Intersection over Union*) adalah salah satu parameter paling penting dalam melakukan evaluasi performa deteksi objek. IoU menghitung tingkat tumpang tindih antara *bounding box* yang dihasilkan oleh algoritma dengan *bounding box* sebenarnya yang terdapat dalam referensi (*dataset*) (Brilliant et al., 2018).

$$IOU = \frac{\text{area of overlap}}{\text{area of union}} = \frac{\text{Diagram 1}}{\text{Diagram 2}}$$


Gambar 12. *Intersection over Union* (Brilliant et al., 2018).

Dalam deteksi objek terdapat sebuah nilai ambang batas yang digunakan untuk menentukan apakah sebuah *bounding box* benar atau salah yang disebut dengan IoU

*threshold*. Nilai IoU *threshold* biasanya digunakan dalam perhitungan mAP untuk menilai ketepatan sebuah model deteksi objek. Pada umumnya nilai IoU *threshold* yang digunakan adalah 0,5 sebagai standar, hal ini mengharuskan area prediksi dan referensi yang tumpang tindih minimal setengah dari seluruh area agar dapat diklasifikasikan benar (Harun, 2020).

## **2.6. Aplikasi Mobile**

Aplikasi Mobile atau biasa disebut dengan *Mobile Apps* adalah perangkat lunak yang dirancang untuk dijalankan pada perangkat seluler, seperti *Smartphone* dan tablet. Aplikasi *mobile* beroperasi pada perangkat seluler dan bertujuan untuk menjalankan fungsi tertentu untuk memenuhi kebutuhan pengguna. Aplikasi *mobile* sendiri memiliki keunggulan yang memudahkan penggunaannya yaitu kemudahan dalam mengakses aplikasi secara portabel sehingga hal ini dapat dilakukan dimana saja dan kapan saja (Hiremath et al., 2021).

## **2.7. Dart**

Dart merupakan bahasa pemrograman yang dirancang guna mendukung pengembangan aplikasi web dan seluler dengan lebih cepat sehingga dapat menghemat waktu *developing*. Dart sendiri menerapkan berbagai konsep pendekatan dari pemrograman berorientasi objek, seperti kelas, abstraksi, pewarisan, enkapsulasi, dan polimorfisme, sehingga memungkinkan pengembang untuk menggunakan konsep pemrograman berorientasi objek yang diterapkan oleh dart (Nagaraj et al., 2022).

## **2.8. Framework flutter**

Flutter merupakan *framework* pengembangan aplikasi *mobile* berperforma tinggi yang berbasis lintas platform. Flutter sendiri dirilis oleh Google pada tahun 2016. Aplikasi flutter dapat berjalan pada beberapa platform yaitu Android, iOS, dan web. Flutter memiliki fitur yang disebut sebagai *statefull hot reload* yang dimana hal ini

adalah salah satu keunggulan utama dari *framework flutter*. *Statefull hot reload* bekerja dengan cara mengirimkan *source code* yang telah diperbarui melalui *Dart Virtual Machine* (Dart VM) hal ini memungkinkan aplikasi untuk bisa bekerja dengan optimal bahkan setelah dilakukannya *hot reload* (Nagaraj et al., 2022).

## 2.9. Visual Studio Code

Visual studio code merupakan *software editor* yang cukup *powerfull*, namun cukup ringan untuk dijalankan. Visual studio code dapat digunakan untuk mengedit berbagai Bahasa pemrograman. *Software* ini memiliki beberapa fitur yang dapat memudahkan penggunanya dalam melakukan pembuatan *source code* fitur-fitur tersebut adalah *auto complete*, *debugging*, *auto save* dan *hot exit* serta pengguna juga dapat menambahkan berbagai ekstensi sehingga mempermudah pembuatan dan pengeditan *source code* (Nagaraj et al., 2022).

## 2.10. Python

Python adalah bahasa pemrograman tingkat tinggi yang serbaguna, dirancang dengan fokus pada kemudahan penggunaan dan keterbacaan kode. Dengan sintaksis yang sederhana dan mudah dipahami, Python menjadi pilihan utama bagi banyak pengembang perangkat lunak untuk berbagai keperluan, mulai dari pengembangan web hingga ilmu data dan kecerdasan buatan (Raihan & Yulianto, 2023).

Python memiliki sebuah kelebihan berupa beragamnya pustaka dan kerangka kerja yang tersedia, mempercepat proses pengembangan dengan menyediakan alat yang diperlukan untuk menangani berbagai tugas dan tantangan. Selain itu, Python bersifat *cross platform*, dapat berjalan diberbagai sistem operasi, menjadikannya pilihan ideal untuk proyek-proyek yang memerlukan fleksibilitas dan portabilitas. Dengan komunitas yang luas dan aktif, Python terus berkembang dan memperkuat posisinya sebagai salah satu bahasa pemrograman paling populer di dunia.

### 2.11. Roboflow

Roboflow adalah platform yang memungkinkan pengembang dan peneliti untuk dengan mudah mengelola, menganalisis, dan menyempurnakan data citra untuk aplikasi kecerdasan buatan. Dengan alat-alat yang canggih, Roboflow memfasilitasi proses pra-pemrosesan data, termasuk *augmentasi* citra, pengaturan label, dan integrasi dengan berbagai kerangka kerja pembelajaran mesin. Dengan demikian, platform ini membantu mengatasi beberapa tantangan utama dalam pengembangan model pembelajaran mesin berbasis citra, seperti kekurangan data yang berkualitas dan kesulitan dalam mengatur dan mengelola *dataset* yang besar.

Selain itu, Roboflow juga menawarkan fitur kolaborasi yang memungkinkan tim untuk bekerja sama secara efisien dalam mengelola dan memperbaiki *dataset* citra. Dengan fitur ini, pengguna dapat dengan mudah berbagi proyek, melacak revisi, dan mengelola versi *dataset*, sehingga mempercepat siklus pengembangan dan meningkatkan kualitas model yang dihasilkan. Dengan demikian, Roboflow menjadi solusi yang penting bagi para praktisi kecerdasan buatan yang mengutamakan efisiensi, kualitas, dan kolaborasi dalam pengembangan model berbasis citra (Roboflow, 2023).

### 2.12. Google Colab

Google Colab adalah platform komputasi berbasis *cloud* yang disediakan oleh Google, dirancang khusus untuk pengembangan dan penelitian dalam bidang ilmu data, pembelajaran mesin, dan kecerdasan buatan. Colab memungkinkan pengguna untuk menulis dan mengeksekusi kode Python dalam lingkungan *notebook* interaktif yang serupa dengan Jupyter *Notebook*, tanpa memerlukan instalasi atau konfigurasi tambahan. Salah satu fitur utama Colab adalah kemampuannya untuk memberikan akses ke GPU dan TPU (*Tensor Processing Unit*) secara gratis, mempercepat waktu pelatihan model pembelajaran mesin yang kompleks. Selain itu, Colab juga mendukung berbagai pustaka dan kerangka kerja populer seperti TensorFlow, PyTorch, dan scikit-learn, serta menyediakan penyimpanan data yang mudah diakses dan berbagi menggunakan Google Drive. Dengan kombinasi antara



kekuatan komputasi *cloud* dan kemudahan penggunaan, Google Colab menjadi pilihan yang populer di kalangan praktisi dan peneliti dalam bidang ilmu data dan kecerdasan buatan (Reina et al., 2019).

### 2.13. *Confusion Matrix*

*Confusion Matrix* adalah sebuah *matrix* yang dapat digunakan untuk melakukan evaluasi pada model klasifikasi dengan cara melakukan prediksi pada objek apakah benar atau salah. *Confusion Matrix* berisikan informasi mengenai klasifikasi aktual dan presisi yang dilakukan model. *Confusion Matrix* disajikan dalam bentuk tabel yang berisikan jumlah data uji yang benar dan salah.

Tabel 2. *Confusion Matrix*

<i>Correct</i>	<i>Classified as</i>	
	<i>Predicted "+"</i>	<i>Predicted "-"</i>
<i>Actual "+"</i>	<i>True Positives</i>	<i>False Negatives</i>
<i>Actual "-"</i>	<i>False Positives</i>	<i>True Negatives</i>

Keterangan :

*True Positives* (TP) = data positif yang terdeteksi benar.

*False Positives* (FP) = data negatif yang terdeteksi sebagai data positif.

*True Negatives* (TN) = data negatif yang terdeteksi benar.

*False Negatives* (FN) = data positif yang terdeteksi sebagai data negatif.

Tabel *confusion matrix* bertujuan untuk menghitung kinerja metode klasifikasi pada model dengan menghitung nilai *recall*, *precision*, *accuracy*, dan *F1-score*.

#### 2.13.1. *Precision*

*Precision* merupakan parameter yang mengukur unit yang bernilai *True Positif*.

*Precision* adalah nilai acuan bagi kita seberapa besar model dapat dipercaya dalam

melakukan prediksi (Grandini *et al.*, 2020). Rumus untuk mencari *precision* adalah sebagai berikut :

$$\frac{TP}{TP + FP}$$

Keterangan:

TP = data positif yang terdeteksi benar pada kelas.

FP = data negatif yang terdeteksi sebagai data positif.

### 2.13.2. Recall

*Recall* mengukur akurasi model untuk kelas positif. *Recall* digunakan untuk mengukur kemampuan dari sebuah model dalam menemukan seluruh unit positif pada *dataset* (Grandini *et al.*, 2020). Untuk menghitung *Recall* dapat digunakan rumus seperti di bawah.

$$\frac{TP}{TP + FN}$$

Keterangan:

TP = data positif yang terdeteksi benar pada kelas.

FP = data negatif yang terdeteksi sebagai data positif.

### 2.13.3. Accuracy

*Accuracy* adalah perhitungan yang digunakan untuk menilai kinerja suatu algoritma. Tingkat kedekatan antara nilai yang diprediksi oleh sistem dan nilai yang sebenarnya didefinisikan sebagai akurasi (Harun, 2020).

$$\frac{(TP + TN)}{(TP + FP + FN + TN)}$$

Keterangan :

TP = data positif yang terdeteksi benar pada.

FP = data negatif yang terdeteksi sebagai data positif.

FN = data positif yang terdeteksi sebagai data negatif.

TN = data negatif yang terdeteksi benar pada kelas.

#### **2.13.4. F1-Score**

F1-Score adalah perbandingan rata-rata presisi (*precision*) dan *recall*. Nilai terbaiknya adalah 1 dan nilai terburuk adalah 0 (Harun, 2020).

$$F1\ Score = 2 \times (Recall \times Precision) / (Recall + Precision)$$

Keterangan:

*Precision* = parameter untuk mengukur ketepatan dari suatu algoritma.

*Recall* = parameter yang digunakan untuk mengukur kelengkapan algoritma.

### III. METODOLOGI PENELITIAN

#### 3.1. Waktu dan Tempat Penelitian

Waktu dan tempat penelitian adalah sebagai berikut :

Tabel 3. Waktu Penelitian

Tahapan	2024																											
	Februari				Maret				April				Mei				Juni				Juli							
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4				
Pengajuan Judul	■	■	■	■																								
Proposal					■	■	■	■																				
Dataset									■	■																		
Pre-Processing											■	■	■	■														
Training Model															■	■	■	■										
Evaluation Model															■	■	■	■										
Develop Mobile Apps																							■	■				
Testing Mobile Apps																											■	■

Penelitian dilakukan pada semester genap 2023/2024 di Kota Bandar Lampung, Gedung Jurusan Ilmu Komputer Fakultas Matematika dan Pengetahuan Alam Universitas Lampung dan di Kediaman yang berlokasi di Jalan Menggala II Kelurahan Kampung Baru Kecamatan Labuhan Ratu Kota Bandar Lampung.

## 3.2. Perangkat Penelitian

Penelitian ini akan menggunakan dua jenis perangkat yaitu perangkat lunak dan juga perangkat keras. Berikut adalah perangkat yang akan digunakan selama penelitian.

### 3.2.1. Perangkat Lunak (*Software*)

- Sistem Operasi Windows 11 Home 64-Bit sebagai sistem operasi yang pada laptop yang digunakan.
- Python versi 3.10.12 sebagai *tools* untuk melakukan *training* dan *testing* pada model YOLOv8.
- Roboflow sebagai *tools* untuk melakukan anotasi pada data.
- Google Colab untuk melakukan proses *training*, *validation*, dan *testing*.
- Sistem operasi Android 13 yang digunakan sebagai sistem operasi untuk melakukan testing.
- Visual Studio Code digunakan sebagai *tools* untuk melakukan pengembangan aplikasi mobile menggunakan *framework* flutter.

### 3.2.2. Perangkat Keras (*Hardware*)

- Laptop ASUS TUF 505DD dengan RAM 16 GB, *Solid State Drive* 512 GB, *Processor* AMD Ryzen 5 3550H dan *Graphics Card* Nvidia GeForce GTX 1050 3GB GDDR5 VRAM sebagai alat pendukung dalam penelitian untuk melakukan pengolahan citra pada *dataset* kerusakan jalan Kota Bandar Lampung.
- *Smartphone* Samsung A32 Untuk melakukan pengambilan citra jalan rusak Kota Bandar Lampung serta sebagai media untuk melakukan testing aplikasi.
- *Stabilizer* DJI Osmo sebagai alat bantu saat melakukan pengambilan citra digital.

### 3.3. Bahan

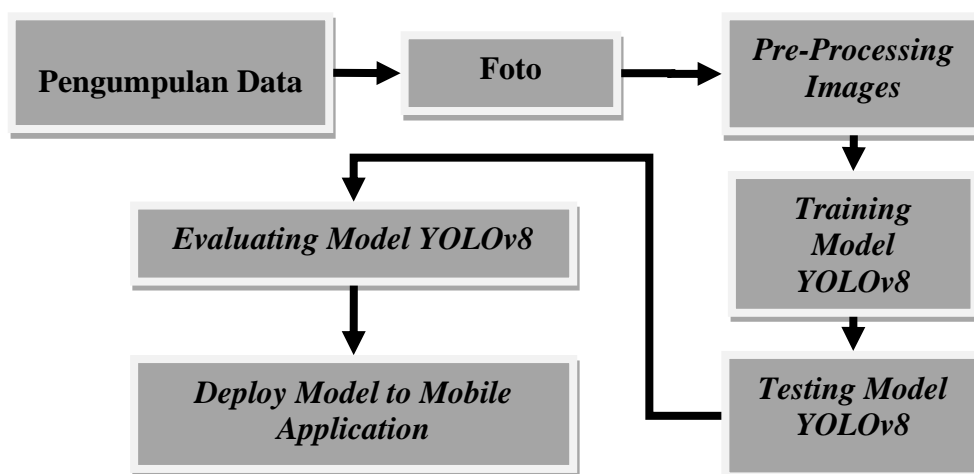
Bahan yang digunakan pada penelitian ini berupa *dataset* yang berisi 100 citra kerusakan jalan yang didapatkan dari gambar olahan pribadi dan dari <https://universe.roboflow.com/orangeki/patches2>, *dataset* berisikan 2 buah *class* yang berpedoman pada *International Roughness Index* (IRI). *Class* disesuaikan dengan jenis kerusakan yang paling banyak terjadi di Kota Bandar Lampung berdasarkan data yang didapat dari <https://data.pu.go.id/dataset/kondisi-permukaan-jalan-provinsi> yang di rilis oleh kementerian PUPR pada awal tahun 2024. Komposisi *dataset* dapat dilihat pada Tabel 4.

Tabel 4. Komposisi *Dataset*

No	Jenis Kerusakan Jalan	Instance
1	Retak	174
2	Berlubang	133
Total Data		307

### 3.4. Tahapan Penelitian

Di bawah ini adalah tahapan penelitian yang akan dilakukan :



Gambar 13. Tahapan Penelitian

### 3.4.1. Pengumpulan Data

Pengumpulan data dilakukan dengan melakukan pengambilan citra digital pada jalan rusak Kota Bandar Lampung serta melakukan pengambilan data dari <https://universe.roboflow.com/orangeki/patches2> yang bertujuan untuk mengumpulkan *dataset*. Citra digital akan digunakan sebagai *dataset* untuk *training*, *validation*, dan *testing*.

### 3.4.2. Pre-Processing Images

*Pre-Processing images* adalah proses untuk mengubah data mentah menjadi data yang lebih mudah dipahami. Tahap ini dilakukan menggunakan *tool* Roboflow. Tahapan ini terbagi menjadi dua yaitu *image annotation* dan *Splitting images datasets*.

#### 1. Images Annotation

*Images annotation* adalah tahapan untuk memberikan label pada setiap gambar yang akan digunakan sebagai *dataset* dengan tujuan menyimpan informasi citra. Proses ini dilakukan dengan membuat *bounding box* dan memberikan nama kelas pada setiap objek pada citra. Proses ini bertujuan untuk memberikan data yang dapat dikenali dengan baik oleh model YOLOv8.

#### 2. Splitting Images Datasets

*Splitting Images Datasets* adalah proses membagi *dataset* ke dalam beberapa bagian yaitu, *training* dan *validation*. Besaran pembagian data untuk masing-masing bagiannya yaitu data *training* sebesar 70% dari jumlah keseluruhan data yang ada yaitu sejumlah 70 gambar yang akan dilakukan *augmentasi* pada setiap citra sehingga menjadi 210 gambar dan *validation* sebesar 20% dari keseluruhan data yang ada yaitu sejumlah 20 gambar serta 10% untuk *testing* yaitu sejumlah 10 gambar. Tahap ini bertujuan agar model memiliki data yang cukup untuk melakukan *training* dengan efektif dan juga memiliki data yang dapat mendukung model dalam melakukan proses *validation* dengan optimal.

### 3.4.3. Training and Evaluate Model YOLOv8

#### 1. Training Model YOLOv8

*Training* data dilakukan dengan *dataset* sebanyak 100 citra kerusakan jalan, proses *training* dilakukan dengan resolusi 160 x 160, 320 x 320, 480 x 480, dan 640 x 640. Tools yang digunakan untuk melakukan *training* adalah Google Colab dengan *runtime* T4 GPU dan menggunakan *Hyperparameter*. *Hyperparameter* yang akan digunakan dapat dilihat pada tabel di bawah ini.

Tabel 5. *Hyperparameter*

No	Parameter	Nilai
1	<i>Epoch</i>	100
2	<i>Batch Size</i>	8
3	<i>Optimizer</i>	AdamW
4	<i>Learning Rate</i>	0,01
5	<i>Momentum</i>	0,9

*Epoch* sebesar 100 digunakan dengan tujuan untuk menguji model apakah model dapat menghasilkan akurasi yang lebih tinggi dengan jumlah pembelajaran yang lebih banyak, selain itu nilai *epoch* 100 adalah nilai yang disarankan oleh pengembang algoritma YOLOv8 untuk digunakan dalam pendeteksian objek. *Batch Size* sebesar 64 bertujuan untuk mengimbangi jumlah *dataset* yang besar agar waktu pembelajaran menjadi semakin cepat. *Optimizer* AdamW merupakan *Optimizer default* yang digunakan oleh arsitektur YOLOv8. Nilai yang digunakan pada *learning rate* dan *momentum* adalah nilai *default* pada arsitektur YOLOv8.

#### 2. Evaluation Model YOLOv8

*Evaluation* pada model dilakukan dengan cara melihat *Confusion Matrix*. *Confusion matrix* didapatkan saat proses *training* dan *validation* telah selesai dilakukan oleh model. Kemudian pengujian model akan dilakukan dengan menggunakan media *Image* yang berisi objek jalan rusak Kota Bandar Lampung



guna menguji keberhasilan dari model yang telah selesai melakukan *training*. Tolak ukur keberhasilan model dapat dilihat dari besar atau kecilnya nilai *precision*, *recall*, *accuracy*, dan *F1-Score*.

#### 3.4.4. *Deploy Model to Mobile Application*

Pada tahap ini dilakukan pembuatan aplikasi *mobile* yang dapat menjalankan model YOLOv8. Pada tahap ini terdapat 2 proses yang akan dilakukan yaitu *develop Mobile Application* dan *Testing Mobile Application*.

##### 1. *Developing Mobile Application*

Proses *developing* akan dilakukan menggunakan *Framework* Flutter, proses ini bertujuan untuk membangun aplikasi sederhana yang dapat menampilkan hasil deteksi dari model YOLOv8. *Package* yang digunakan dalam proses pembuatan aplikasi adalah *flutter\_version* sebagai *plugin*. Aplikasi yang dihasilkan harus dapat dijalankan tanpa koneksi internet.

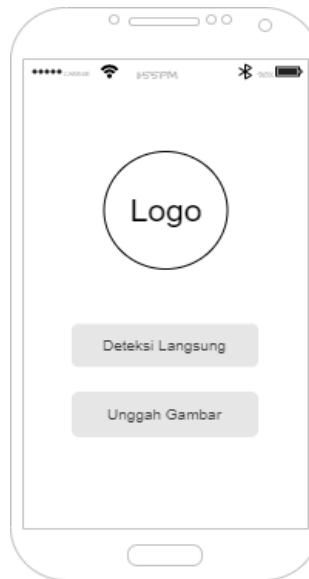
Proses ini dimulai dengan menentukan tujuan dari dibuatnya aplikasi yaitu untuk memudahkan pengamatan jalan rusak di Kota Bandar Lampung. Kebutuhan fitur aplikasi dapat dilihat pada tabel 6.

Tabel 6. Kebutuhan Fitur

No	Fitur	Deskripsi
1	Deteksi Kamera	Menampilkan kamera yang dapat menghasilkan <i>bounding box</i> pada kerusakan jalan
2	Identifikasi Gambar	Menambahkan file gambar dan menampilkan hasil identifikasi pada gambar

Proses dilanjutkan dengan melakukan *prototyping* aplikasi yang bertujuan mempermudah proses *developing*, prototipe yang dibuat akan merujuk kepada tabel kebutuhan fitur pada tabel 6 Prototipe tampilan aplikasi adalah sebagai berikut :

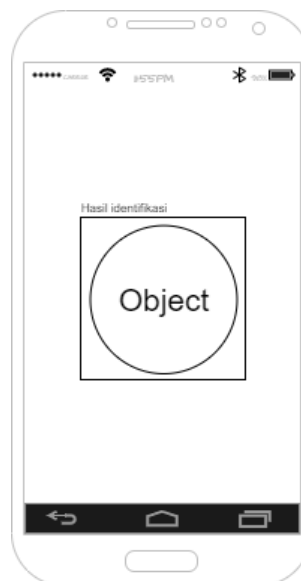
a. Prototipe Halaman *Home Screen*



Gambar 14. Prototipe *Home Screen*

Tampilan *Home Screen* adalah halaman utama dari aplikasi. Halaman ini daftar yang mengarahkan pengguna menuju menu pendeteksian *realtime* menggunakan kamera atau identifikasi melalui gambar. Tampilan *Home Screen* dapat dilihat pada gambar 15.

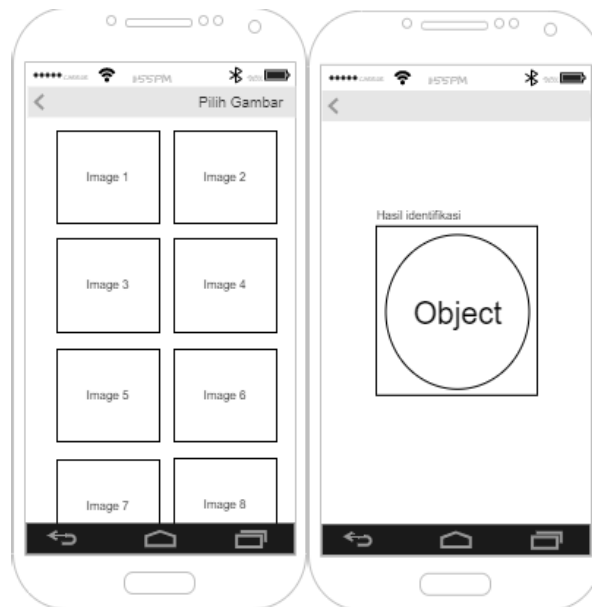
b. Prototipe Tampilan Halaman *Realtime Detection*



Gambar 15. Prototipe Halaman *Realtime Detection*

Gambar 22 menunjukkan halaman deteksi yang menggunakan kamera dan menampilkan *bounding box* deteksi secara *real-time* dengan menggunakan model YOLOv8 yang telah diekspor ke Tensorflow Lite.

c. Prototipe Tampilan Halaman *Image Detection*



Gambar 16. Prototipe Halaman *Image Detection*

Tampilan sisi kiri pada gambar 17 merupakan halaman *upload Image* sedangkan pada sisi kanan merupakan halaman yang menampilkan hasil pendeteksian dan *bounding box*.

2. *Testing Mobile Application*

Proses ini bertujuan untuk melakukan pengujian terhadap aplikasi yang sudah dibuat dan akan memfokuskan fungsionalitas dari *input* dan *output* dari aplikasi. Pada tahap ini aplikasi *mobile* yang telah dibuat menggunakan *framework* flutter akan dideploy ke perangkat *smartphone* dengan proses *debugging* menggunakan Visual Studio Code kemudian proses testing akan dilakukan dengan menggunakan data *test* pada *dataset* dan juga dilakukan pengetesan di lapangan secara langsung, selain itu untuk mengetahui apakah aplikasi berjalan baik atau tidak maka harus dilakukan *testing* pada setiap fungsi aplikasi. Skenario pengujian aplikasi dapat dilihat pada tabel 7 sampai

tabel 9.

Tabel 7. Skenario Pengujian *Home Screen*

<b>No</b>	<b>Test ID</b>	<b>Test Scenario</b>
1	A-01	Pengguna membuka menu utama dan memilih menu <i>image detection</i>
2	A-02	Pengguna membuka menu utama dan memilih menu <i>realtime detection</i>
3	A-03	Pengguna membuka menu utama dan tidak memilih menu

Tabel 8. Skenario Pengujian *Realtime Detection*

<b>No</b>	<b>Test ID</b>	<b>Test Scenario</b>
1	B-01	Pengguna melakukan deteksi langsung menggunakan kamera untuk mendeteksi objek berupa kerusakan jalan
2	B-02	Pengguna melakukan deteksi langsung menggunakan kamera untuk mendeteksi objek selain kerusakan jalan

Tabel 9. Skenario Pengujian *Image Detection*

<b>No</b>	<b>Test ID</b>	<b>Test Scenario</b>
1	C-01	Pengguna mengunggah gambar berupa gambar kerusakan jalan
2	C-02	Pengguna mengunggah gambar selain gambar kerusakan jalan

## V. KESIMPULAN DAN PEMBAHASAN

### 5.1. Kesimpulan

Kesimpulan yang diperoleh berdasar hasil penelitian ini adalah sebagai berikut :

1. Penelitian ini membuktikan bahwa pendeteksian jalan rusak di Kota Bandar Lampung dapat dilakukan dengan mengimplementasikan model YOLOv8.
2. Pada penelitian ini dihasilkan model YOLOv8 dengan nilai persentase mAP sebesar 87.1% *0.5 threshold*. Akurasi pada *testing* menggunakan data *test* pada *dataset* sebesar 90% dengan nilai *error* sebesar 10% dan akurasi pada saat *testing* di lapangan sebesar 86.7% dengan nilai *error* sebesar 13.3%.
3. Penelitian ini membuktikan bahwa model YOLOv8 dapat diimplementasikan pada aplikasi *mobile* untuk melakukan identifikasi dan klasifikasi kerusakan jalan di Kota Bandar Lampung.
4. Penelitian ini membuktikan bahwa *image size* yang digunakan dapat mempengaruhi hasil akurasi dari model yang dibuat, hal ini dapat dilihat pada gambar 23.

### 5.2. Saran

Saran untuk penelitian yang selanjutnya adalah sebagai berikut :

1. Melakukan pengambilan *dataset* secara pribadi agar *dataset* yang di dapat memiliki ukuran *pixel* yang seragam.
2. Melakukan improvisasi terhadap algoritma YOLOv8 untuk mendapatkan nilai akurasi yang lebih baik.
3. Membuat *dataset* yang lebih variatif sehingga dapat memperkecil kemungkinan

kesalahan deteksi pada objek.

4. Melakukan pengembangan aplikasi *mobile* dengan menggunakan *framework* yang berbeda seperti React Native, Ionic, Xamarin atau Corona SDK.

## DAFTAR PUSTAKA

- Adityah, D. A. (2021). *Deteksi dan Klasifikasi Keretakan Jalan Menggunakan Metode You Only Look Once*. repositori.usu.ac.id. <https://repositori.usu.ac.id/handle/123456789/47311>
- Afriani, L. (2014). Pengertian Infrastruktur. *Graha Ilmu*, 87, 1–2.
- Afriyana, L., Salmah, E., Sriningsih, S., Harsono, I., & Kunci, K. (2023). Analisis Dampak Pembangunan Infrastruktur Terhadap Pertumbuhan Ekonomi Inklusif Pada Kabupaten / Kota di Provinsi Nusa Tenggara Barat Tahun 2016-2021 Info Artikel Keywords : Pertumbuhan ekonomi di Indonesia dipengaruhi oleh banyak faktor selain atensi Pem. *Jurnal Ekonomi Pembangunan*, 5(1), 1–12.
- Bertarina, Mahendra, O., Lestari, F., & Safitri, D. (2022). Analisis Pengaruh Hambatan Samping (Studi Kasus: Jalan Raya Za Pagar Alam di Bawah Flyover Kedaton Kota Bandar Lampung). *Jurnal Teknik Sipil ITP*, 9(1), 5. <https://doi.org/10.21063/jts.2022.v901.05>
- Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). *YOLOv8: Optimal Speed and Accuracy of Object Detection*. <http://arxiv.org/abs/2004.10934>
- Brilliant, S., Adji, B., Kusumawardani, S., & Hidayah, I. (2018). Rekomendasi Berdasarkan Nilai Pretest Mahasiswa Menggunakan Metode Collaborative Filtering dan Bayesian Ranking. *Edu Komputika*, 5(1), 13–22.
- Eriksson, M. (2023). *Road damage detection with Yolov8 on Swedish roads*. *Civilingenjörprogrammet i informationsteknologi*. <https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1803747&dswid=-8465>
- Feurer, M., Klein, A., Jost, K. E., Springenberg, T., Blum, M., & Hutter, F. (2019).

- Auto-sklearn: efficient and robust automated machine learning. In: Automated machine learning: methods, systems, challenges.* 113–134. <http://automl.org>
- Grandini, M., Bagli, E., & Visani, G. (2020). *Metrics for Multi-Class Classification: an Overview.* 1–17. <http://arxiv.org/abs/2008.05756>
- Harun, A. (2020). Implementasi Deep Learning Menggunakan Metode You Only Look Once untuk Mendeteksi Rokok. *Jurnal Ekonomi Volume 18, Nomor 1 Maret 201*, 2(1), 41–49.
- Hidayat. (2020). Bab II Tinjauan Pustaka Faktor Penyebab Kerusakan Jalan. *BAB II Tinjauan Pustaka Faktor Penyebab Kerusakan Jalan*, 13, 4–15. [http://eprints.itenas.ac.id/1056/5/05\\_Bab\\_2\\_222016129.pdf](http://eprints.itenas.ac.id/1056/5/05_Bab_2_222016129.pdf)
- Hiremath, R., Malshikare, K., Mahajan, M., & Kulkarni, R. V. (2021). *A Smart App for Pothole Detection Using Yolo Model BT - ICT Analysis and Applications* (S. Fong, N. Dey, & A. Joshi (eds.); pp. 155–164). Springer Singapore.
- Indonesia.go.id. (2024). *Jalan makin Mulus, Arus Barang makin Lancar.* <https://indonesia.go.id/kategori/editorial/8272/jalan-makin-mulus-arus-barang-makin-lancar?lang=1>
- Jensen, S. H. N., Doest, M. E. B., Aanæs, H., & Del Bue, A. (2021). A Benchmark and Evaluation of Non-Rigid Structure from Motion. *International Journal of Computer Vision*, 129(4), 882–899. <https://doi.org/10.1007/s11263-020-01406-y>
- Jiang, Y., Yan, H., Zhang, Y., Wu, K., Liu, R., & Lin, C. (2023). RDD-YOLOv5: Road Defect Detection Algorithm with Self-Attention Based on Unmanned Aerial Vehicle Inspection. *Sensors*, 23(19), 1–28. <https://doi.org/10.3390/s23198241>
- Jumadi, J., Yudianti, Y., & Sartika, D. (2021). Pengolahan Citra Digital Untuk Identifikasi Objek Menggunakan Metode Hierarchical Agglomerative Clustering. *JST (Jurnal Sains Dan Teknologi)*, 10(2), 148–156. <https://doi.org/10.23887/jstundiksha.v10i2.33636>
- Khare, O. M., Gandhi, S., Rahalkar, A. M., & Mane, S. (2023). *YOLOv8-Based*



*Visual Detection of Road Hazards: Potholes, Sewer Covers, and Manholes.*  
<http://arxiv.org/abs/2311.00073>

- Lecun, Y., Bottou, L., Bengio, Y., Haffner, P., Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). *Gradient-based learning applied to document recognition* To cite this version : HAL Id : hal-03926082 *Gradient-Based Learning Applied to Document Recognition*. 86(11), 2278–2324.
- Loshchilov, I., & Hutter, F. (2019). Decoupled weight decay regularization. *7th International Conference on Learning Representations, ICLR 2019*.
- Mubarok, H. (2019). Identifikasi Ekspresi Wajah Berbasis Citra Menggunakan Algoritma Convolutional Neural Network (CNN). *Universitas Islam Negeri Maulana Malik Ibrahim Malang*, 3(1), 10–12.
- Mukhtar, E., Sudjani, & Supriatna, N. (2022). Desain Road Barrier untuk Persimpangan Jalan (Studi Kasus: Jalan Layang Jakarta, Bandung) *Amar. JPTB: Jurnal Pendidikan Teknik Bangunan*, 2(1), 31–40.
- Nagaraj, K., Prabakaran, B., & Ramkumar, M. O. (2022). Application Development for a Project using Flutter. *2022 3rd International Conference on Smart Electronics and Communication (ICOSEC)*, 947–951.  
<https://doi.org/10.1109/ICOSEC54921.2022.9951938>
- Pamungkas, A. (2024). *Pengolahan Citra Digital dan Aplikasinya*.  
<https://pemrogramanmatlab.com/2024/02/06/pengolahan-citra-digital-dan-aplikasinya/>
- Pasaribu, S. D., Putri, D. I. H., & Suranegara, G. M. (2023). *Implementasi Algoritma YOLOv8 Dan Deep Learning Untuk Rancang Bangun Sistem Deteksi Objek Kendaraan Pada Jam Sibuk Di Jalan Pasir Kaliki (Paskal Timur) Tjokroaminoto Bandung*.
- Pramestya, R. H. (2018). Deteksi dan Klasifikasi Kerusakan Jalan Aspal Menggunakan Metode YOLO Berbasis Citra Digital. *Institut Teknologi Sepuluh Nopember*, 91. [http://repository.its.ac.id/59044/1/06111650010019-Master\\_Thesis.pdf](http://repository.its.ac.id/59044/1/06111650010019-Master_Thesis.pdf)

- Pratama, T. A. (2021). Penerapan Arsitektur Mobilenet Dan Algoritma Soft-NMS Pada Faster R-CNN Untuk Deteksi Pengguna Jalan. *Universitas Islam Negeri Malik Ibrahim*, 16–22.
- Putra, N. S. (2023). *Implementasi Algoritma Convolutional Neural Network Untuk Identifikasi Jenis Kelamin Dan Ras Pada Citra Wajah*.
- Raihan, R. M., & Yulianto, S. (2023). Penerapan Pemrograman Python Dalam Menentukan Waktu Overhaul Kondensor Turbin Uap. *Jurnal Konversi Energi Dan Manufaktur*, 8(1), 49–57. <https://doi.org/10.21009/jkem.8.1.6>
- Rawansyah, Subhi, D. H., & Alim, M. S. (2020). Sistem Pendukung Keputusan Penentuan Prioritas Perbaikan Jalan Rusak Dengan Metode Multifactor Evaluation Process ( MFEP ) ( Studi Kasus Kabupaten Bojonegoro ). *Seminar Informatika Aplikatif Polinema (Siap)*, 124–129.
- Reina, P. N., Agustin, F. E. M., & Hakiem, N. (2019). Penerapan Metode Convolutional Neural Network Pada Pendeteksi Gambar Notasi Balok. In *Rabit : Jurnal Teknologi dan Sistem Informasi Univrab* (Vol. 1, Issue 1).
- Roboflow. (2023). *Getting Started with Roboflow*. <https://blog.roboflow.com/getting-started-with-roboflow/>
- Rochmawati, N., Hidayati, H. B., Yamasari, Y., Tjahyaningtijas, H. P. A., Yustanti, W., & Prihanto, A. (2023). Analisa Learning rate dan Batch size Pada Klasifikasi Covid Menggunakan Deep learning dengan Optimizer Adam. *Jurnal Pendidikan Dan Teknologi Indonesia*, 3(6), 265–271. <https://doi.org/10.52436/1.jpti.306>
- Sasmito, B., Setiadji, B. H., & Isnanto, R. (2023). Deteksi Kerusakan Jalan Menggunakan Pengolahan Citra Deep Learning di Kota Semarang. *Teknik*, 44(1), 7–14. <https://doi.org/10.14710/teknik.v44i1.51908>
- Sharma, D. K., Pradhan, R., & Agrawal, R. (2022). Comparative Study on Real-Time Vehicle Classification. *International Conference on System Modeling & Advancement in Research Trends (SMART)*. <https://doi.org/10.1109/SMART55829.2022.10046885>

- Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 1–14.
- Umi, T., Setyawan, A., & Suprpto, M. (2016). Penggunaan Metode International Roughness Index (IRI), Surface Distress Index (SDI) Dan Pavement Condition Index (PCI) untuk Penilaian Kondisi Jalan Di Kabupaten Wonogiri. *Prosiding Semnastek*, 1–9.  
<https://jurnal.umj.ac.id/index.php/semnastek/article/view/685>
- Wang, X., Gao, H., Jia, Z., & Li, Z. (2023). BL-YOLOv8: An Improved Road Defect Detection Model Based on YOLOv8. *Sensors (Basel, Switzerland)*, 23(20). <https://doi.org/10.3390/s23208361>
- Wasil, M., Harianto, & Fathurrahman. (2022). Pengaruh Epoch pada Akurasi menggunakan Convolutional Neural Network untuk Klasifikasi fashion dan Furniture. *Infotek: Jurnal Informatika Dan Teknologi*, 5(1), 53–61.  
<https://doi.org/10.29408/jit.v5i1.4393>
- Widiyanto, E. P. (2023). *AUTOMATED PAVEMENT DEFECT DETECTION USING YOLOv8 OBJECT DETECTION*. 1–13.  
<https://doi.org/10.58674/phpji.v16i1.388>