

**PENGEMBANGAN YOLOv4 DENGAN FITUR EKSTRAKTOR  
MobileNetV3 UNTUK DETEKSI DAN KLASIFIKASI PLAT  
KENDARAAN**

**(Skripsi)**

**Oleh**

**AURA HUSNAINI PUTRI ZIDANI  
NPM 2017051045**



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

## **ABSTRAK**

### **PENGEMBANGAN YOLOv4 DENGAN FITUR EKSTRAKTOR MobileNetV3 UNTUK DETEKSI DAN KLASIFIKASI PLAT KENDARAAN**

**Oleh**

**AURA HUSNAINI PUTRI ZAIDANI**

*Computer vision* banyak dimanfaatkan untuk pengembangan teknologi seperti deteksi objek. Dalam deteksi objek, terdapat berbagai algoritma yang dapat digunakan salah satunya YOLO. Metode ini menggunakan *deep learning* untuk melakukan deteksi secara *real-time*. Pada penelitian ini akan dilakukan pengembangan YOLOv4 dengan fitur ekstraktor MobileNetV3 untuk melakukan deteksi dan klasifikasi plat kendaraan. Hasil penelitian ini menunjukkan bahwa pengembangan YOLOv4-MobileNetV3 memiliki tingkat performa yang lebih tinggi dibandingkan original YOLOv4, yang menggunakan CSPDarknet53 sebagai fitur ekstraktor. Evaluasi perbandingan performa antara kedua model adalah dari segi performa akurasi dan waktu komputasi. YOLOv4-MobileNetV3 memiliki rata-rata akurasi sebesar 97.54% sedangkan YOLOv4-CSPDarknet53 memiliki rata-rata akurasi sebesar 96.93%. Dalam hal waktu komputasi, YOLOv4-MobileNetV3 membutuhkan waktu yang lebih sedikit yaitu rata-rata sekitar 0.133 *seconds* sedangkan YOLOv4-CSPDarknet53 membutuhkan rata-rata waktu sekitar 0.418 *seconds* untuk melakukan deteksi warna plat kendaraan.

Kata Kunci: *Deep Learning*, Fitur Ekstraktor, YOLOv4, MobileNetV3, Deteksi, Klasifikasi, Plat Kendaraan.

## **ABSTRACT**

### ***IMPROVED YOLOv4 WITH FEATURE EXTRACTION MobileNetV3 FOR DETECTION AND CLASSIFICATION VEHICLE PLATE***

***By***

**AURA HUSNAINI PUTRI ZAIDANI**

*Computer vision is widely utilized in the development of technologies such as object detection. In object detection, various algorithms can be used, one of which is YOLO. This method uses deep learning to detect vehicle plates on the highway in real time. In this research, the YOLOv4 model was improved with the MobileNetV3 feature extractor for vehicle license plate detection and classification. The results of this study show that the development of YOLOv4-MobileNetV3 has a higher performance compared to the original YOLOv4, which uses CSPDarknet53 as a feature extractor. The performance comparison evaluation between the two models focuses on accuracy and computational time. YOLOv4-MobileNetV3 achieved an average accuracy of 97.54%, whereas YOLOv4-CSPDarknet53 achieved an average accuracy of 96.93%. In terms of computation time, YOLOv4-MobileNetV3 required less time, averaging around 0.133 seconds, compared to YOLOv4-CSPDarknet53, which averaged around 0.418 seconds for vehicle license plate color detection.*

***Keywords: Deep Learning, Feature Extraction, YOLOv4, MobileNetV3, Detection, Classification, Vehicle Plate.***

**PENGEMBANGAN YOLOv4 DENGAN FITUR EKSTRAKTOR  
MobileNetV3 UNTUK DETEKSI DAN KLASIFIKASI PLAT  
KENDARAAN**

**Oleh**

**AURA HUSNAINI PUTRI Z Aidani**

**Skripsi**

**Sebagai Salah Satu Syarat untuk Memperoleh Gelar  
SARJANA ILMU KOMPUTER**

**Pada**

**Jurusan Ilmu Komputer  
Fakultas Matematika dan Ilmu Pengetahuan Alam**



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

**2024**

Judul Skripsi : **PENGEMBANGAN YOLOv4 DENGAN FITUR EKSTRAKTOR MobileNetV3 UNTUK DETEKSI DAN KLASIFIKASI PLAT KENDARAAN**

Nama Mahasiswa : **Aura Husnaini Putri Zaidani**

Nomor Pokok Mahasiswa : 2017051045

Program Studi : **S1-Ilmu Komputer**

Fakultas : **Matematika dan Ilmu Pengetahuan Alam**



**1. Komisi Pembimbing**

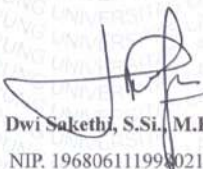
  
**Dr. rer. nat. Akmal Junaidi, M.Sc**

**NIP. 197101291997021001**

  
**Ridho Sholehurrohman, M. Mat**

**NIP. 232111970128101**

**2. Ketua Jurusan Ilmu Komputer**



**Dwi Sakethi, S.Si., M.Kom**

**NIP. 196806111994021001**

**MENGESAHKAN**

**1. Tim Penguji**

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



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



**Penguji  
Bukan Pembimbing : 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: 11 Juli 2024**

## PERNYATAAN

Saya bertanda tangan di bawah ini:

Nama : Aura Husnaini Putri Zaidani

NPM : 2017051045

Dengan ini menyatakan bahwa skripsi saya yang berjudul **“Pengembangan YOLOv4 dengan Fitur Ekstraktor MobileNetV3 Untuk Deteksi dan Klasifikasi Plat Kendaraan”** merupakan hasil karya saya sendiri dan bukan karya orang lain. Seluruh tulisan yang tertuang di skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila di kemudian hari terbukti skripsi saya merupakan hasil penjiplakan atau dibuat orang lain, maka saya bersedia menerima sanksi berupa pencabutan gelar yang telah saya terima.

Bandar Lampung, 11 Juli 2024



Aura Husnaini Putri Zaidani  
NPM. 2017051045

## RIWAYAT HIDUP



Penulis dilahirkan pada tanggal 27 Februari 2003 di Liwa-Balik Bukit, Kabupaten Lampung Barat, Provinsi Lampung sebagai anak pertama dari tiga bersaudara. Ayah bernama Drs. Azwari (Alm) dan Ibu Endang Wahyuni, S.IP., MM. Penulis menyelesaikan Pendidikan Sekolah Dasar di SDN 1 Martapura, Ogan Komering Ulu Timur pada tahun 2014, menyelesaikan Sekolah Menengah Pertama di SMPN 1 Martapura pada tahun 2017, kemudian menyelesaikan sekolah di jenjang Sekolah Menengah Atas di SMAN 3 Martapura pada tahun 2020.

Pada tahun 2020, penulis terdaftar sebagai mahasiswa Jurusan Ilmu Komputer FMIPA Universitas Lampung melalui jalur SBMPTN. Selama menjadi mahasiswa, penulis melakukan beberapa kegiatan antara lain.

1. Menjadi anggota Adapter Himpunan Mahasiswa Jurusan Ilmu Komputer pada periode 2020/2021.
2. Menjadi anggota Staf Ahli Keuangan Himpunan Mahasiswa Jurusan Ilmu Komputer pada periode 2021.
3. Menjadi anggota Biro Kesekretariatan Himpunan Mahasiswa Jurusan Ilmu Komputer pada periode 2022.
4. Menjadi Asisten Dosen Jurusan Ilmu Komputer pada tahun 2022 hingga 2023.



5. Menjadi Sekretaris Koordinator Divisi Dana Usaha pada acara Pekan Raya Jurusan Ilmu Komputer 2021.
6. Menjadi Sekretaris Pelaksana pada acara Pekan Raya Jurusan Ilmu Komputer tahun 2022.
7. Mengikuti *Course UI/UX Designer Pemula* pada Program Kredensial Mikro Mahasiswa (KMMI) tahun 2021.
8. Melaksanakan Kerja Praktik di PT Jasa Raharja Putera Cabang Bandar Lampung pada periode I tahun 2023.
9. Melaksanakan Kuliah Kerja Nyata (KKN) di desa Hanura, Kecamatan Teluk Pandan Kabupaten Pesawaran, Provinsi Lampung pada tahun 2023 dengan program kerja pengembangan *website* pencatatan PBB Desa Hanura.

## MOTTO

1. *“Tuhanmu tiada meninggalkan kamu dan tiada (pula) benci kepadamu.”*  
(QS. Ad-Dhuha: 3)
2. *“Sesungguhnya beserta kesulitan ada kemudahan.”*  
(QS. Al-Insyirah: 6)
3. *“Ingatlah, sesungguhnya pertolongan Allah itu dekat.”*  
(QS. Al-Baqarah: 214)
4. *When you leave everything in Allah’s hand, you’ll see Allah’s hand in everything.*  
(Anonim)
5. *You can’t keep doing the same thing and expect different results.*  
(Anonim)
6. *Trust what was meant to be yours, will be yours. Unrush yourself.*  
(Anonim)

## **PERSEMBAHAN**

*Alhamdulillahirobbilalamin*

Puji syukur kehadiran Allah Subhanahu Wa Ta'ala atas segala rahmat dan karunia-Nya sehingga skripsi ini dapat diselesaikan dengan sebaik-baiknya. Shalawat beriring salam selalu tercurahkan kepada junjungan Nabi Agung Muhammad Shallallahu 'Alaihi Wasallam.

Aku persembahkan karya ini kepada:

### **Kedua Orang Tuaku Tercinta**

Yang selalu mendukung, memberikan cinta dan kasih sayang yang tak terhingga, serta do'a yang selalu menyertaiku. Kuucapkan terima kasih sebesar-besarnya atas pengorbanan dan perjuangan dalam mendidik dan membesarkanku yang tak akan dapat terbalaskan. Kuucapkan juga terima kasih kepada adik-adikku atas dukungan dan do'a yang senantiasa diberikan kepadaku.

### **Seluruh Keluarga Besar Ilmu Komputer 2020**

Yang senantiasa memberikan semangat dan dukungan

### **Almamater Tercinta, Universitas Lampung dan Jurusan Ilmu Komputer**

Tempat bernaung mengemban ilmu untuk menjadi bekal hidup

## SANWACANA

Puji Syukur kehadiran Allah Subhanahu Wa Ta'ala atas limpahan rahmat dan karunia-Nya. Shalawat serta salam senantiasa tercurahkan kepada junjungan Nabi Agung Muhammad Shallallahu 'Alaihi Wasallam., sehingga penulis dapat menyelesaikan skripsi dengan judul **“Pengembangan YOLOv4 Dengan Fitur Ekstraktor MobileNetV3 Untuk Deteksi dan Klasifikasi Plat Kendaraan”**. Penulis mengucapkan terima kasih yang paling tulus kepada semua pihak yang telah membantu dalam melaksanakan penelitian dan penyusunan laporan skripsi ini, antara lain:

1. Kedua orang tua yang sangat penulis sayangi, Ayah dan Ibu yang senantiasa mengajarkan kebaikan, memberikan perhatian, cinta dan kasih sayang, do'a terbaik, serta kepercayaan atas setiap keputusan yang diambil oleh penulis hingga detik ini.
2. Saudari penulis, Nurrahmi Putri Zaidani dan Fitrah Maryam Putri Zaidani yang selalu memberi dukungan, doa, dan semangat dalam setiap langkah penyusunan skripsi hingga saat ini.
3. Bapak Dr. rer. nat. Akmal Junaidi, M.Sc selaku pembimbing utama yang telah memberikan arahan dan bimbingan dalam penelitian ini.
4. Bapak Ridho Sholehurrohman, M. Mat selaku pembimbing kedua dalam penelitian ini yang selalu memberikan arahan, bimbingan, dan bantuan ketika mengalami kesulitan dalam menyelesaikan penelitian ini.
5. Prof. Admi Syarif, Ph.D sebagai pembahas yang telah memberikan masukan serta saran yang bermanfaat dalam perbaikan skripsi ini.
6. Bapak Dr. Eng. Heri Satria, S.Si., M.Si. selaku dekan FMIPA Universitas Lampung.

7. Bapak Dwi Sakethi, S.Si., M.Kom selaku Ketua Jurusan Ilmu Komputer FMIPA Universitas Lampung.
8. Ibu Anie Rose Irawati, S.T., M.Cs selaku Sekretaris Jurusan Ilmu Komputer FMIPA Universitas Lampung.
9. Bapak Didik Kurniawan, S.Si., M.T selaku dosen pembimbing akademik.
10. Seluruh Dosen, Staf, dan Karyawan Jurusan Ilmu Komputer yang telah memberikan ilmu, pelajaran, dan bantuan terbaik selama penulis menempuh pendidikan di Jurusan Ilmu Komputer Universitas Lampung.
11. Teman-teman seperjuangan semasa kuliah Melan Caniadi, Yulia Dwi Putri, Ages Mahesa, Dita Faradila, dan Pynka Aryani Angelia Haryanto yang selalu mendukung, menemani, dan berbagi cerita indah selama masa perkuliahan.
12. Sahabat-sahabat penulis Monalisa Febriyanti, Sintia Rani Prastita, Tasya Aulia Maharani, Diandra Vania Putri, dan Rara Fitra Oktora yang selalu memberikan dukungan dan semangat kepada penulis.
13. Teman-teman Himakom yang sudah mengajarkan banyak hal dalam berorganisasi dan memberikan banyak pengalaman.
14. Keluarga Ilmu Komputer 2020 yang telah memberikan pengalaman berharga selama menjalankan studi di Jurusan Ilmu Komputer Universitas Lampung.
15. Seluruh pihak yang telah membantu penulis secara langsung maupun tidak langsung, atas dukungannya dalam menyelesaikan skripsi.

Penulis menyadari bahwa penyusunan skripsi ini masih jauh dari kata sempurna. Namun penulis sangat mengharapkan skripsi ini dapat bermanfaat bagi para civitas akademik Universitas Lampung pada umumnya dan mahasiswa Ilmu Komputer pada khususnya.

Bandar Lampung, 11 Juli 2024



Aura Husnaini Putri Zaidani  
NPM. 2017051045

## DAFTAR ISI

	Halaman
<b>DAFTAR ISI</b> .....	<b>xiv</b>
<b>DAFTAR TABEL</b> .....	<b>xvii</b>
<b>DAFTAR GAMBAR</b> .....	<b>xviii</b>
<b>DAFTAR KODE PROGRAM</b> .....	<b>xx</b>
<b>I. PENDAHULUAN</b> .....	<b>21</b>
1.1 Latar Belakang .....	21
1.2 Rumusan Masalah .....	23
1.3 Batasan Masalah.....	24
1.4 Tujuan Penelitian.....	24
1.5 Manfaat Penelitian.....	24
<b>II. TINJAUAN PUSTAKA</b> .....	<b>25</b>
2.1 Penelitian Terdahulu .....	25
2.2 Peraturan Kepolisian tentang TNKB.....	27
2.3 <i>Computer vision</i> .....	27
2.4 Pengolahan Citra Digital .....	28
2.4.1 Jenis Citra Digital .....	29
2.5 <i>Deep Learning</i> .....	30
2.5.1 <i>Convolutional Neural Network</i> .....	30
2.5.2 <i>Artificial Neural Network</i> .....	31
2.5.3 <i>Convolutional Layer</i> .....	32
2.5.4 <i>Pooling Layer</i> .....	33
2.5.5 Fungsi Aktivasi .....	34
2.5.6 <i>Fully Connected Layer</i> .....	34

2.5.7	Fitur Ekstraktor .....	35
2.5.8	CSPDarkNet53.....	35
2.5.9	MobileNetV3 .....	36
2.5.10	YOLOv4 .....	38
2.6	<i>Confusion Matrix</i> .....	40
2.6.1	<i>Accuracy</i> .....	42
2.6.2	<i>Recall</i> .....	42
2.6.3	<i>Precision</i> .....	42
2.6.4	<i>F1 Score</i> .....	42
<b>III. METODOLOGI PENELITIAN.....</b>		<b>43</b>
3.1	Waktu dan Tempat Penelitian .....	43
3.2	Alat dan Bahan .....	44
3.3	Alur Kerja Penelitian.....	45
3.3.1	Pengambilan Data .....	45
3.3.2	<i>Preprocessing</i> .....	45
3.3.3	Analisis YOLOv4-CSPDarknet53 .....	46
3.3.4	Analisis MobilenetV3 .....	46
3.3.5	Perancangan YOLOv4-MobileNetV3 .....	47
3.3.6	<i>Training Model</i> .....	47
3.3.7	Analisis Hasil .....	48
3.3.8	Evaluasi Model .....	48
<b>IV. PEMBAHASAN.....</b>		<b>49</b>
4.1	Deskripsi Arsitektur YOLOv4-CSPDarknet53 .....	49
4.1.1	CSPDarknet53 .....	50
4.1.2	<i>Convolutional Block</i> .....	51
4.1.3	<i>Residual Block</i> .....	54
4.1.4	<i>Spatial Pyramid Pooling (SPP)</i> .....	55
4.1.5	<i>Path Aggregation Network(PAN)</i> .....	56
4.2	Analisis CSPDarknet53 .....	57
4.3	Arsitektur MobileNetV3.....	57
4.4	Perancangan YOLOv4-MobileNetV3 .....	59
4.4.1	Penyesuaian MobileNetV3 Terhadap YOLOv4 .....	59

4.5	Implementasi YOLOv4-MobileNetV3.....	62
4.5.1	Pengambilan Data .....	62
4.5.2	<i>Preprocessing</i> .....	63
4.5.3	Implementasi <i>Training</i> Data .....	72
4.5.4	<i>Training Configuration</i> .....	73
4.6	Hasil <i>Training</i> Model .....	73
4.6.1	Hasil Training YOLOv4-CSPDarknet53 .....	74
4.6.2	Hasil Training YOLOv4-MobileNetV3 .....	74
4.6.3	Perbandingan Tahap <i>Training</i> .....	75
4.7	Analisis Hasil .....	75
4.7.1	Hasil Testing YOLOv4-CSPDarknet53 .....	76
4.7.2	Hasil Testing YOLOv4-MobileNetv3.....	77
4.8	Evaluasi Model.....	78
4.8.1	Evaluasi YOLOv4-CSPDarknet53 .....	78
4.8.2	Evaluasi YOLOv4-MobileNetV3 .....	80
4.8.3	Perbandingan Performa Model .....	82
<b>V.</b>	<b>PENUTUP.....</b>	<b>85</b>
5.1	Simpulan.....	85
5.2	Saran .....	86
	<b>DAFTAR PUSTAKA.....</b>	<b>87</b>



## DAFTAR TABEL

Tabel	Halaman
1. Penelitian Terdahulu.....	25
2. Confusion Matrix .....	41
3. Timeline Penelitian .....	43
4. Alat Pendukung (Hardware).....	44
5. Arsitektur CSPDarknet53 .....	50
6. Arsitektur MobileNetV3.....	58
7. Hasil Augmentasi .....	66
8. Jumlah Objek Hasil Anotasi Tiap Class .....	72
9. Konfigurasi Hyperparameter.....	73
10. Training YOLOv4-CSPDarknet53 .....	74
11. Training YOLOv4-MobileNetV2 .....	74
12. Confusion Matrix YOLOv4-CSPDarknet53 .....	79
13. Hasil Testing YOLOv4-CSPDarknet53 .....	79
14. Evaluasi Deteksi dan Klasifikasi YOLOv4-CSPDarknet53 .....	79
15. Confusion Matrix YOLOv4-MobileNetV3.....	80
16. Hasil Testing YOLOv4-MobileNetV3 .....	81
17. Evaluasi Deteksi dan Klasifikasi YOLOv4-MobileNetV3 .....	81
18. Perbandingan running-time Model .....	83

## DAFTAR GAMBAR

Gambar	Halaman
1. Penyusun nilai citra.....	28
2. Jenis-jenis Citra Digital.....	29
3. Artificial Intelligence, Machine Learning, dan Deep Learning. ....	30
4. Arsitektur CNN .....	31
5. Artificial Neural Network .....	32
6. Proses Konvolusi dengan 2 Filter Untuk Menghasilkan Feature Map .....	33
7. Average pooling dan Max pooling .....	34
8. Struktur CSPDarknet53.....	36
9. Standard Convolution, Depthwise Convolution, dan Pointwise Convolution..	37
10. Struktur MobileNetV3 .....	38
11. Arsitektur YOLOv4.....	39
12. YOLOv4 Object Detector Architecture.....	39
13. Perbandingan YOLOv4 dengan model lain .....	40
14. Alur Penelitian.....	45
15. Alur YOLOv4-CSPDarknet53 .....	49
16. CSPDenseblock.....	51
17. Convolutional Block .....	52
18. Residual Block .....	54
19. Modifikasi SPP Pada YOLOv4 .....	55
20. (a) PAN (b) Modifikasi PAN.....	56
21. Rancangan YOLOv4-MobileNetV3 .....	59
22. Skema Depthwise Separable .....	60
23. Struktur Bottleneck MobileNetV3 .....	60
24. Arsitektur Pengembangan YOLOv4-MobileNetV3.....	61
25. Ilustrasi Pengambilan Data .....	63
26. Citra Hasil Ekstrak Video.....	64

27. Perbandingan Dataset Sebelum dan Sesudah Augmentasi.....	66
28. Citra Hasil Cropping .....	67
29. Citra hasil resize .....	67
30. Tampilan File Hasil Anotasi .....	69
31. Tampilan File classes.txt .....	69
32. Pembuatan Kotak Pembatas .....	70
33. Ilustrasi Koordinat Kotak Pembatas.....	70
34. Koordinat Kotak Pembatas .....	71
35. Grafik Perbandingan Running-Time Training .....	75
36. Hasil Testing YOLOv4-CSPDarknet53 Gambar 1 .....	76
37. Hasil Testing YOLOv4-CSPDarknet53 Gambar 2 .....	77
38. Hasil Testing YOLOv4-MobileNetV3 Gambar 1 .....	77
39. Hasil Testing YOLOv4-MobileNetV3 Gambar 2 .....	78
40. Grafik Perbandingan accuracy, precision, recall, dan F1 score YOLOv4- CSPDarknet53.....	80
41. Grafik Perbandingan accuracy, precision, recall, dan F1 score YOLOv4- MobileNetV3 .....	82
42. Grafik Perbandingan Akurasi.....	82

## DAFTAR KODE PROGRAM

Kode Program	Halaman
1. SPP pada YOLOv4.....	55
2. Ekstrak Video .....	64
3. Augmentasi Data. ....	65
4. Cropping gambar.....	67
5. Resize gambar .....	68

# I. PENDAHULUAN

## 1.1 Latar Belakang

Perkembangan teknologi, terutama dalam ranah deteksi objek, telah mengalami kemajuan pesat dalam beberapa tahun terakhir. *Computer Vision* diperkenalkan pada tahun 1960-an, dan sejak saat itu telah banyak penelitian yang difokuskan pada pengembangan algoritma untuk mengenali dan mendeteksi objek dalam gambar. *Computer vision* adalah suatu sistem yang mengajarkan komputer untuk mengenali dunia sekitar. Fokus utamanya adalah untuk menciptakan mesin yang memiliki kemampuan untuk "melihat" dunia dengan cara serupa seperti manusia, dan kemudian menggunakan informasi visual tersebut sebagai dasar untuk mengambil keputusan (Indra, Herman, & Budi, 2020). *Computer vision* menjadi salah satu solusi untuk membangun sistem dalam mendukung pembangunan *smart city*, seperti yang direncanakan oleh Korps Lalu Lintas Polri. Teknologi ini digunakan untuk mendeteksi objek yang dalam hal ini adalah plat kendaraan bermotor. Pendeteksian objek (*Object detection*) digunakan untuk mengenali dan mendeteksi objek pada sebuah gambar berdasarkan dari warna dan bentuk dari *dataset* yang dikumpulkan (Lin, Deng, Albers, & Siebert, 2020). Salah satu metode yang digunakan untuk deteksi objek adalah metode *Deep Learning*. Beberapa algoritma deteksi seperti *Fast-RCNN*, *Faster-RCNN*, dan *You Only Look Once (YOLO)* telah menjadi pilihan utama dalam pengembangan sistem deteksi yang mengalami perkembangan signifikan dan menghasilkan hasil yang akurat dengan beban komputasi yang relatif rendah dalam berbagai konteks (Aprilino & Al Amin, 2022). Dalam penelitian ini, YOLO dipilih sebagai metode deteksi yang akan digunakan karena memiliki kombinasi kecepatan dan akurasi yang optimal dalam tugas deteksi objek.

YOLO pertama kali diperkenalkan oleh Redmon, Divvala, Girshick, dan Farhadi (2016) yang menjelaskan bahwa YOLO dapat mendeteksi objek dengan cepat dan akurasi yang tinggi. YOLO menggunakan jaringan *Convolutional Neural Network* (CNN) untuk deteksi dan klasifikasi objek menggunakan kotak pembatas (*bounding box*). Algoritma YOLOv4 meningkatkan FPS dan presisi rata-rata (AP) sebesar 12% dan 10% dibandingkan pendahulunya, YOLOv3. Dengan peningkatan ini, YOLOv4 menunjukkan kinerja AP50 sebesar 65.7% dalam pelatihan menggunakan *dataset* MS COCO, dan mampu berjalan dengan kecepatan *real-time* sekitar 65 FPS pada sistem Tesla V100 (Mulyanto, Jatmiko, Mursanto, Prasetyawan, & Borman, 2021).

Dalam deteksi objek, jaringan saraf tiruan (Neural Network) digunakan sebagai backbone untuk mengekstraksi fitur-fitur penting dari gambar input sebelum digunakan untuk mendeteksi objek. Pada *Computer vision*, backbone untuk fitur ekstraksi akan dilatih pada banyak tugas untuk menunjukkan efektivitasnya (Elharrouss, Akbari, Almaadeed, & Al-Maadeed, 2022). Pemilihan fitur ekstraktor penting untuk meningkatkan tingkat akurasi deteksi dan waktu komputasi yang dibutuhkan. YOLOv4 mengeksplorasi beberapa backbone dan metode augmentasi data seperti CSPResNext50, CSPDarknet53, dan EfficientNet-B3. Setelah melalui banyak pengujian dan didapatkan hasil eksperimen, CSPDarknet53 dipilih menjadi backbone dari YOLOv4.

Pada penelitian Yuan & Xu (2021) berjudul "*Lightweight Vehicle Detection Algorithm Based on Improved YOLOv4*" dikembangkan YOLOv4 dengan mengganti fitur ekstraktor aslinya yaitu CSPDarknet53 menjadi MobileNetV3. Hasil penelitiannya menunjukkan bahwa pengembangan ini berhasil mengurangi parameter hingga 78% dan meningkatkan akurasi sebesar 0.53% dibandingkan dengan YOLOv4 asli. Pada penelitian Huang, He, & Chen (2021) berjudul "*M-YOLO: A Nighttime Vehicle Detection Method Combining MobileNetV2 and YOLOv3*" yang dikembangkan berdasarkan YOLOv3 dengan menggunakan jaringan MobileNetV2 dapat mengurangi jumlah parameter dan ukuran model. Dalam penelitian Jamil,

Fadhil, Hamzah, & Ramli (2021) berjudul “*Comparing MobileNet-SSD and YOLOv3 Learning Architecture for Real-time Driver’s Fatigue Detection*” menjelaskan bahwa MobileNet-SSD memiliki kecepatan dan akurasi yang lebih baik dibandingkan dengan YOLOv3. Meskipun MobileNet-SSD memperoleh rata-rata presisi yang sedikit lebih rendah jika dibandingkan dengan YOLOv3, namun MobileNet-SSD memiliki kecepatan deteksi hampir empat kali lipat melampaui YOLOv3.

Berdasarkan uraian di atas, dapat dipertimbangkan bahwa MobileNet memiliki keunggulan dalam kinerja model. Pada penelitian ini akan dikembangkan model YOLOv4 dengan fitur ekstraktor MobileNetV3 untuk deteksi dan klasifikasi. YOLOv4-MobileNetV3 kemudian akan dibandingkan dengan YOLOv4-CSPDarknet53 dengan menguji metode untuk deteksi dan klasifikasi gambar plat kendaraan di jalan.

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang di atas, dapat dirumuskan suatu permasalahan dalam penelitian ini yaitu sebagai berikut:

- a. Bagaimana merancang arsitektur dan algoritma YOLOv4 dengan fitur ekstraktor MobileNetV3?
- b. Bagaimana implementasi YOLOv4 dengan fitur ekstraktor MobileNetV3 untuk deteksi dan klasifikasi plat kendaraan?
- c. Bagaimana perbandingan performa akurasi dan waktu komputasi deteksi dan klasifikasi plat kendaraan dengan menggunakan YOLOv4-MobileNetV3 dan YOLOv4- CSPDarknet53?

### 1.3 Batasan Masalah

Batasan masalah dalam penelitian ini yaitu sebagai berikut:

- a. Data yang digunakan adalah gambar lalu lintas kendaraan di jalan raya yang diambil menggunakan kamera *smartphone*.
- b. Pengambilan gambar diambil secara langsung di Jl. Raden Intan, Kota Bandar Lampung dalam kondisi cukup cerah.
- c. Plat nomor yang digunakan adalah plat nomor yang berlaku di Indonesia yaitu plat yang berwarna merah, putih, hitam, dan kuning. Selain itu, terdapat tambahan satu kelas yaitu non plat yang mencakup objek kendaraan tanpa TNKB (Tanda Nomor Kendaraan Bermotor) dan objek lain yang terdeteksi sebagai plat kendaraan oleh model.

### 1.4 Tujuan Penelitian

Adapun tujuan dari penelitian ini yaitu sebagai berikut:

- a. Merancang arsitektur YOLOv4 dengan arsitektur MobileNetV3 untuk deteksi dan deteksi dan klasifikasi plat nomor kendaraan.
- b. Mengimplementasikan YOLOv4 dengan fitur ekstraktor MobileNetV3 untuk deteksi dan klasifikasi plat kendaraan.
- c. Membandingkan performa akurasi dan waktu komputasi dari YOLOv4-MobileNetV3 dan YOLOv4-CSPDarknet53 dalam deteksi dan klasifikasi plat kendaraan.

### 1.5 Manfaat Penelitian

- a. Penelitian ini diharapkan dapat dijadikan sebagai rujukan penelitian lainnya mengenai deteksi dan klasifikasi plat kendaraan.
- b. Penelitian ini dapat memberikan kontribusi dan manfaat untuk mendukung peningkatan sistem e-tilang, pengembangan *Smart City* dan *Traffic Management System* (TMS).



## II. TINJAUAN PUSTAKA

### 2.1 Penelitian Terdahulu

Tabel 1. Penelitian Terdahulu

Peneliti	Judul	Data	Hasil
(Liu, Wang, Zhang, Liu, & Zhao, 2022)	<i>Real-Time Defect Detection Of Metal Surface Based on Improved YOLOv4</i>	Data permukaan logam yang cacat sebanyak 1500 gambar digunakan dalam penelitian ini.	Penelitian ini menghasilkan akurasi sebesar 91.13% dan kecepatan deteksi 26.39 FPS.
(Liao, et al., 2021)	<i>YOLOv4-MN3 for PCB Surface Defect Detection</i>	Data yang digunakan yaitu 2008 gambar PCB dengan cacat permukaan.	Penelitian ini memiliki akurasi 98.64% dengan kecepatan deteksi mencapai 56.98 FPS.
(Huang, He, & Chen, 2021)	<i>M-YOLO: A Nighttime Vehicle Detection Method Combining Mobilenet v2 and YOLO v3</i>	<i>Dataset</i> yang digunakan sekitar 5.576 gambar lalu lintas di malam hari	Hasil dari penelitian ini yaitu rata-rata akurasi dari M-YOLO mencapai 94.96% dan kecepatan deteksi mencapai 10 FPS, tiga kali lebih cepat dari YOLOv3.

Penelitian oleh Liu, Wang, Zhang, Liu, & Zhao (2022) berjudul “*Real-Time Defect Detection Of Metal Surface Based on Improved YOLOv4*” menggunakan *dataset* yang memiliki 5 kelas yang merupakan tipe permukaan logam yang cacat, dimana tiap kelas memiliki 300 gambar sehingga total keseluruhan gambar yang digunakan adalah 1500. Data tersebut dibagi menjadi *training*, *test*, dan *verification set* dengan rasio 8:1:1. Penelitian ini mengganti CSPDarknet53 pada YOLOv4 menjadi MobileNetV3. Pengembangan model ini menghasilkan peningkatan mAP dari 83.19% menjadi 91.13% dan peningkatan kecepatan dari 21.87 FPS menjadi 26.39 FPS dibandingkan YOLOv4 asli dalam deteksi cacat permukaan logam. Penelitian yang dilakukan oleh Liao, et al. (2021) judul “*YOLOv4-MN3 for PCB Surface Defect Detection*” menggunakan gambar PCB dengan permukaan cacat/rusak yang diambil dari pabrik produksi PCB di China. Jumlah gambar yang diperoleh adalah 2008, kemudian dilakukan proses augmentasi sehingga diperoleh 19029 gambar dengan resolusi 416×416 di enam kategori cacat. Dataset kemudian dibagi menjadi 90% untuk *training* dan 10% untuk testing secara random. Pada penelitian ini digunakan MobileNetV3 untuk menggantikan CSPDarknet53 sebagai fitur ekstraktor dari YOLOv4. Dari hasil penelitian ini didapatkan akurasi tertinggi yaitu 98.64% dengan kecepatan deteksi 56.98 FPS. Selanjutnya penelitian oleh Huang, He, & Chen (2021) berjudul “*M-YOLO: A Nighttime Vehicle Detection Method Combining MobileNetV2 and YOLO v3*” menggunakan *dataset* sebanyak 5,576 gambar lalu lintas di malam hari dengan resolusi 640×360. Dalam *training* model, 90% gambar dipilih secara *random* untuk dipilih sebagai set pelatihan dan 10% sisanya digunakan untuk set tes. Pada penelitian ini digunakan M-YOLO yang merupakan YOLOv3 dengan MobileNetV2 sebagai fitur ekstraktor. Dengan menggunakan *depth separable convolution*, MobileNet dapat mengurangi parameter dan ukuran dari model. Hasil dari penelitian ini menunjukkan akurasi rata-rata dari M-YOLO dapat mencapai 94.96% dengan AP 18.45% lebih tinggi dibandingkan YOLOv3.

## 2.2 Peraturan Kepolisian tentang TNKB

Pada awal Juni tahun 2022, Korps Lalu Lintas Polri mulai menerapkan kebijakan pergantian TNKB secara bertahap dari hitam menjadi putih. Kebijakan perubahan warna TNKB tercantum di dalam Peraturan Kepolisian Nomor 7 Tahun 2021 tentang Registrasi dan Identifikasi Kendaraan Bermotor yang ditandatangani oleh Kapolri Jendral Listyo Sigit Prabowo. Adapun ketentuan warna baru dari TNKB yang tercantum pada Pasal 45 Peraturan Kepolisian Nomor 7 Tahun 2021 Ayat (1) yaitu plat dasar putih tulisan hitam untuk kendaraan perseorangan, badan hukum, perwakilan negara asing, dan badan internasional; plat dasar kuning tulisan hitam untuk kendaraan umum; plat dasar merah tulisan putih untuk kendaraan instansi pemerintah; plat dasar hijau tulisan hitam untuk kendaraan di kawasan perdagangan bebas. Selain empat TNKB tersebut, terdapat tanda khusus untuk TNKB Ranmor listrik seperti yang telah ditetapkan dengan Keputusan Kakorlantas Polri yaitu plat dasar putih dengan tulisan hitam disertai penambahan warna biru di bagian bawahnya.

## 2.3 *Computer vision*

*Computer vision* merupakan salah satu cabang dari kecerdasan buatan yang memungkinkan komputer untuk memahami dan menginterpretasikan informasi visual dari dunia nyata. *Computer vision* memungkinkan komputer untuk meniru kinerja mata manusia dalam hal merekam, menganalisis, dan menginterpretasikan citra dan video untuk mendapatkan informasi dan pemahaman dari objek tertentu untuk kemudian diterapkan dalam perangkat mesin atau komputer (Arnita, Marpaung, Aulia, Suryani, & Nabila, 2022).

Dalam penerapannya terdapat tiga kategori *vision* yaitu *Low-Level Vision*, *Intermediate-Level Vision*, serta *3-D Vision* dan gerakan. *Low-Level Vision* mencakup operasi dan tugas yang mendasar dalam pemrosesan gambar seperti bagaimana komputer memahami gambar sebagai objek digital. *Intermediate-Level Vision* membantu dalam menggambarkan konteks dan

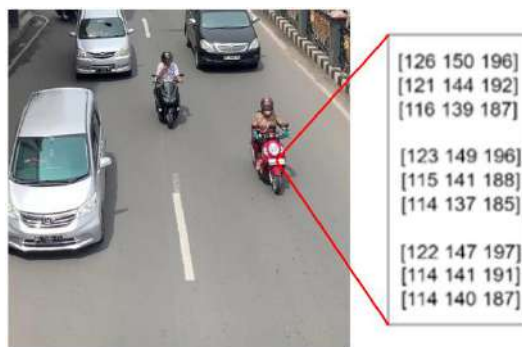
hubungan antara elemen-elemen visual dalam gambar seperti menganalisis pola, garis, dan tepi dari objek. *3-D Vision* dan gerakan berfokus pada pemahaman objek dalam tiga dimensi termasuk pemahaman ruang dan struktur objek. Komputer mampu mengidentifikasi objek berbentuk tiga dimensi dengan menganalisis bentuk dan gambar yang timbul akibat pencahayaan pada permukaan objek.

## 2.4 Pengolahan Citra Digital

Pengolahan citra merupakan proses yang bertujuan meningkatkan kualitas gambar atau mengungkap informasi yang terkandung dalamnya dengan menonjolkan aspek-aspek tertentu dari gambar (Fadjeri, Saputra, Adri Ariyanto, & Kurniatin, 2022). Ada beberapa langkah yang diperlukan dalam proses pengolahan citra:

- 1) Mengimpor gambar menggunakan alat akuisisi gambar,
- 2) Menganalisis dan memanipulasi citra,
- 3) Hasil dari proses ini dapat berupa citra yang telah dimodifikasi atau data hasil analisis citra.

Citra digital terbentuk melalui metode yang dikenal dengan nama kuantisasi. Kuantisasi adalah proses mengubah intensitas piksel dari sinyal analog menjadi bilangan bulat dengan nilai yang berhingga (Arnita, Marpaung, Aulia, Suryani, & Nabila, 2022).



Gambar 1. Penyusun nilai citra

Pengolahan citra adalah proses perbaikan kualitas citra agar lebih mudah diidentifikasi atau dipresentasikan oleh manusia atau mesin (komputer). Ketika citra mengandung derau (*noise* atau bintik-bintik putih), kabur, atau memiliki warna yang kurang intens, pengolahan citra digunakan untuk memperbaikinya sehingga citra tersebut dapat dengan lebih baik diinterpretasikan baik oleh manusia maupun mesin (komputer) (Ikhsan, Utami, & Wibowo, 2020).

### 2.4.1 Jenis Citra Digital

#### a. Citra Digital Berwarna (RGB)

Citra berwarna atau RGB terdiri dari 3 komponen yaitu merah (Red atau R), hijau (Green atau G), dan biru (Blue atau B). Setiap komponen memiliki rentang nilai 0-255.

#### b. Citra Digital Biner

Citra biner merupakan citra yang setiap piksel hanya memiliki satu dari dua nilai intensitas yang mungkin, yaitu 0 atau 1. Dalam citra ini, nilai 1 dan 0 digunakan untuk merepresentasikan warna hitam dan putih.

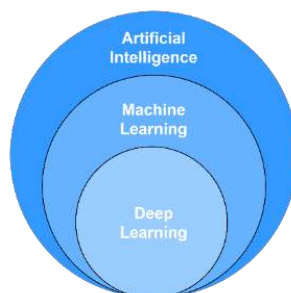
#### c. Citra Digital Keabuan (Grayscale)

Citra keabuan (*grayscale*) adalah citra dengan intensitas warna keabuan yang dinyatakan dalam rentang 0-255, dimana 0 menyatakan hitam dan 255 menyatakan putih dan nilai di antaranya adalah derajat keabuan.



Gambar 2. Jenis-jenis Citra Digital

## 2.5 Deep Learning



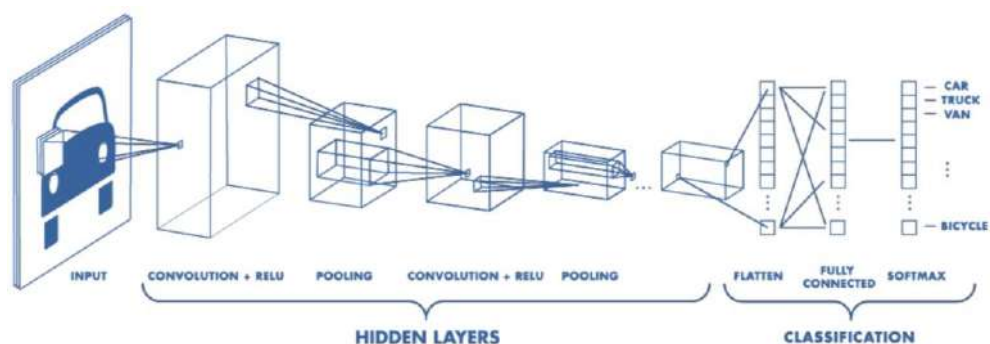
Gambar 3. *Artificial Intelligence, Machine Learning, dan Deep Learning.*

*Deep Learning* adalah subdivisi dari *Machine Learning* yang melibatkan algoritma pemodelan tingkat tinggi pada data dengan menggunakan rangkaian fungsi transformasi non-linier yang tersusun dalam lapisan-lapisan yang dalam (Cholissodin & Soebroto, 2020). *Deep Learning* efektif digunakan dalam berbagai jenis pembelajaran, termasuk *supervised learning*, *unsupervised learning*, *semi-supervised learning*, dan juga *reinforcement learning*, di berbagai konteks aplikasi seperti pengenalan gambar, suara, klasifikasi teks, dan lainnya (Cholissodin & Soebroto, 2020). Pada dasarnya, dalam *Deep Learning*, model-model dibentuk menggunakan Jaringan Saraf Tiruan (*Neural Network*). Apabila suatu jaringan memiliki lebih dari tiga lapisan, maka jaringan tersebut diklasifikasikan sebagai Jaringan Dalam (*Deep Network*).

### 2.5.1 *Convolutional Neural Network*

*Convolutional Neural Network* (CNN) adalah bentuk lain dari *Multilayer Perceptron* yang terinspirasi oleh cara kerja otak manusia. CNN terdiri dari lapisan dengan neuron dalam bentuk 3D yang menunjukkan lebar, tinggi, dan kedalaman. Lebar dan tinggi menunjukkan ukuran lapisan, sementara kedalaman berkaitan dengan jumlah lapisan. Umumnya, layer CNN dibagi menjadi dua yaitu Layer ekstraksi fitur gambar dan Layer klasifikasi. Layer ekstraksi fitur gambar terdiri dari beberapa layer dimana setiap layer terdiri dari neuron yang terhubung dengan daerah lokal pada layer sebelumnya. Terdapat dua jenis layer utama di dalamnya, yaitu layer konvolusi dan layer

*pooling*. Fungsi aktivasi diterapkan pada setiap layer, dan posisinya secara bergantian antara jenis pertama dan jenis kedua. Layer klasifikasi, terdiri dari beberapa layer dimana setiap layer tersusun atas neuron yang saling terkoneksi penuh (*fully connected*). Layer ini mengonversi vektor hasil ekstraksi fitur gambar menjadi skoring kelas untuk klasifikasi, mirip dengan *Multi Neural Networks* dengan tambahan beberapa *hidden layer* (Kusuma & Mauizah, 2023).



Gambar 4. Arsitektur CNN

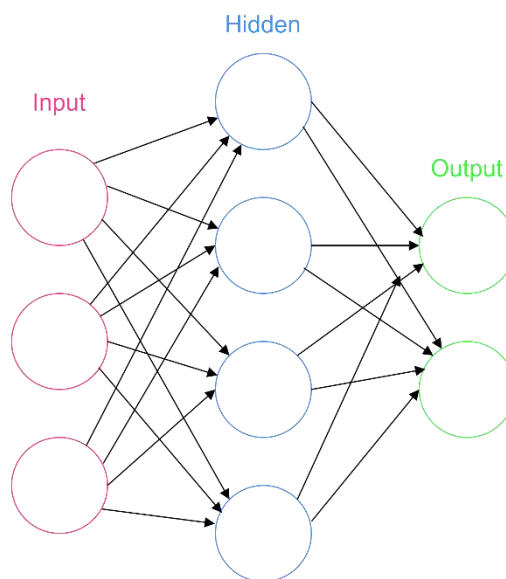
(MathWorks, 2017).

Layer CNN terdiri dari lapisan *input*, lapisan *ouput*, dan beberapa *hidden layers*. *Hidden layers* terdiri dari *convolutional layers*, *ReLU layer*, *fully connected layers*, dan *loss layer*. Arsitektur CNN menerima *input* berupa *raw image* yang tersusun dari arsitektur tiga dimensi yaitu lebar (*widht*), tinggi (*height*), dan dalam (*depth*).

### 2.5.2 Artificial Neural Network

*Artificial Neural Network* (ANN) merupakan sistem komputasi yang mengabstraksi jaringan saraf manusia dari sudut pandang pemrosesan informasi dan menyusun berbagai jaringan berdasarkan koneksi-koneksi yang berbeda (Wu & Feng, 2017). Dalam bidang teknik dan akademisi, istilah yang umum digunakan adalah jaringan saraf atau *neural network*. Jaringan saraf ini merupakan suatu model komputasi yang terdiri dari sejumlah besar *node* (*neuron*) yang saling terhubung. Setiap *node* merepresentasikan suatu fungsi keluaran khusus yang disebut sebagai fungsi aktivasi. Hubungan antara setiap

dua *node* mewakili bobot yang disebut sebagai *weight*, yang setara dengan memori dari jaringan saraf tiruan. *Output* dari jaringan akan bervariasi tergantung pada bagaimana jaringan terhubung, nilai bobot, dan fungsi insentif. ANN dapat digunakan untuk mengidentifikasi pola-pola dalam data, dan aplikasinya yang cocok termasuk analisis data citra (Pramana, Khrisne, & Sastra, 2021).

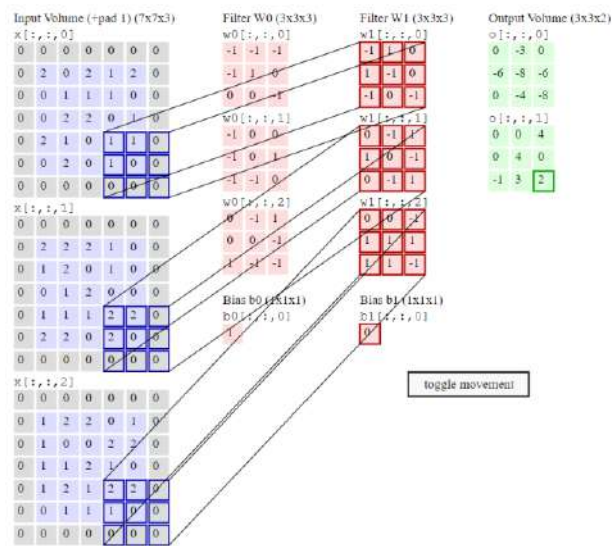


Gambar 5. *Artificial Neural Network*  
(Wu & Feng, 2017)

### 2.5.3 *Convolutional Layer*

*Convolutional Layer* adalah komponen sentral dalam jaringan CNN dan bertanggung jawab atas sebagian besar perhitungan dalam CNN. *Convolutional Layer* adalah lapisan yang melakukan pemindaian pada data *input* dan menggunakan filter untuk menjalankan operasi konvolusi, menghasilkan fitur map sebagai *output* dari data yang awalnya berbentuk citra. *Input* citra berukuran  $n \times n$  akan dipecah menjadi tiga *channel* yaitu *layer Red*, *layer Green*, dan *layer Blue* lalu akan membentuk *array* multidimensi berukuran  $n \times n \times 3$ . *Convolutional Layer* meningkatkan kompleksitas model secara signifikan dengan mengoptimalkan *output*. Berikut ini ilustrasi dari proses konvolusi ditunjukkan pada Gambar 6.

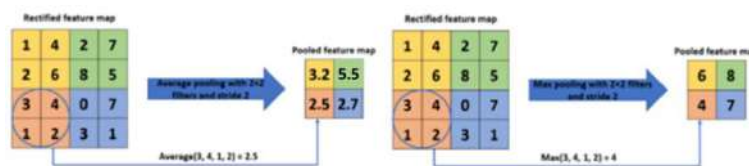




Gambar 6. Proses Konvolusi dengan 2 Filter Untuk Menghasilkan *Feature Map* (Karpathy, 2018).

#### 2.5.4 Pooling Layer

*Pooling layer* merupakan bagian penting dalam *Convolutional Neural Network* (CNN) dimana layer ini melakukan *downsampling* pada fitur map yang berasal dari *layer* sebelumnya dan menghasilkan fitur map baru dengan resolusi yang lebih padat. *Pooling layer* memiliki dua tujuan utama yaitu mengurangi jumlah parameter atau bobot sehingga mengurangi biaya komputasi dan mengendalikan *overfitting* dari jaringan. Metode *pooling* diharapkan hanya mengekstrak informasi yang berguna dan membuang detail yang tidak relevan (Gholamalinezhad & Khosravi, 2020). Dalam *Convolutional Neural Network* (CNN), terdapat dua jenis *pooling layer* yang umum digunakan, yaitu *max pooling* dan *average pooling*. *Max pooling* memilih elemen maksimum dari wilayah fitur map yang ditutupi oleh *filter*, sedangkan *average pooling* menghitung rata-rata elemen yang ada di wilayah fitur map yang ditutupi oleh filter. Berikut ini adalah contoh dari *average pooling* dan *max pooling*.



Gambar 7. Average pooling dan Max pooling  
(Gholamalinezhad & Khosravi, 2020).

### 2.5.5 Fungsi Aktivasi

Fungsi aktivasi adalah suatu fungsi non-linier yang memungkinkan jaringan menangani permasalahan kompleks yang memerlukan analisis mendalam. Setiap fungsi aktivasi beroperasi pada suatu nilai dengan operasi matematika tertentu. Dalam arsitektur CNN, fungsi aktivasi diterapkan pada akhir perhitungan keluaran fitur map setelah operasi konvolusi atau *pooling* dan menghasilkan pola fitur. Terdapat beberapa fungsi aktivasi yang dapat digunakan dalam penelitian seperti *Rectified Linear Unit* (ReLU), Leaky ReLU (LReLU), *Parametric ReLU*, *sigmoid*, dan *tanh* (Zufar & Setiyono, 2016).

### 2.5.6 Fully Connected Layer

*Fully connected layer* adalah lapisan akhir yang bertugas melakukan transformasi fitur map yang berbentuk matriks menjadi vektor. *Layer* ini terdiri dari tiga bagian yaitu *input layer*, *hidden layer* dan *output layer* (Sholehurrohman, 2021). Pada layer ini, jaringan saraf tiruan di setiap neuron terhubung ke semua neuron dalam layer sebelumnya. Untuk menghubungkan semua neuron di lapisan *Fully-Connected*, setiap aktivasi dari lapisan sebelumnya harus diubah menjadi bentuk data satu dimensi terlebih dahulu. Perbedaan dari lapisan konvolusi biasa dengan lapisan *fully-connected* terletak pada koneksi antara neuron-neuronnya. Pada lapisan konvolusi, neuron hanya terhubung dengan area tertentu pada *input*. Sedangkan lapisan *fully-connected* memiliki koneksi yang mencakup seluruh *input*. Meskipun demikian, keduanya tetap menggunakan operasi perkalian

titik (*dot product*) sehingga peran fungsionalnya tidak berbeda secara signifikan (Fauzy, 2019).

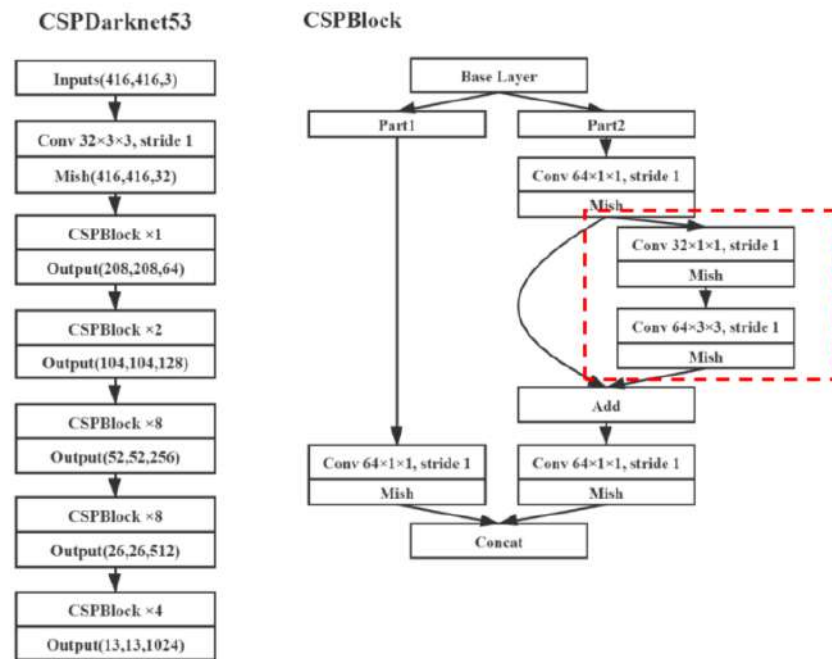
### 2.5.7 Fitur Ekstraktor

Fitur ekstraktor adalah suatu metode dalam pemrosesan citra atau gambar yang bertujuan menghasilkan fitur map yang berisi informasi penting, sehingga mempermudah pengelolaan dan pengolahan dalam tugas deteksi dan klasifikasi objek. Proses ekstraksi fitur menjadi bermanfaat ketika diperlukan pengurangan sumber daya yang digunakan untuk pemrosesan, tanpa kehilangan data yang signifikan atau relevan. Fitur ekstraktor juga mampu mengurangi volume data yang berlebihan yang harus dianalisis, serta mengurangi beban kerja mesin dalam membangun kombinasi variabel (fitur) untuk pelatihan yang efisien dan efektif.

### 2.5.8 CSPDarkNet53

Algoritma YOLOv4 meningkatkan fitur ekstraktor menjadi CSPDarknet53 untuk menggantikan fitur ekstraktor Darknet53 berdasarkan konsep CSPNet (Fu, Chen, & Lv, 2022). Perbedaan CSPDarknet53 dari versi Darknet53 adalah penggunaan konsep “*Cross-Part-Partial Connection*” atau CSPNet. Konsep ini memungkinkan jaringan untuk membagi pengaturan fitur yang signifikan antara beberapa bagian, yang dapat meningkatkan akurasi dalam tugas deteksi objek. Selain itu, CSPDarknet53 dirancang untuk menjaga kecepatan operasi jaringan sehingga dapat digunakan dalam aplikasi *real-time*.

CSPNet dengan DarkNet53 yang disebut model CSPDarknet53 memiliki presisi yang lebih tinggi dalam deteksi objek dibandingkan ResNet. Model ini dapat membagi pengaturan fitur yang signifikan sambil mempertahankan kecepatan operasi jaringan. Struktur dari CSPDarknet53 ditunjukkan pada Gambar 8 berikut.

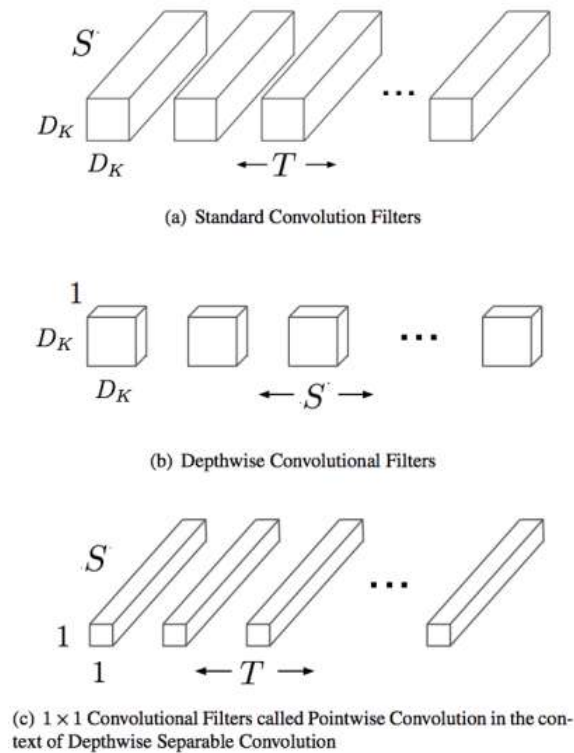


Gambar 8. Struktur CSPDarknet53

(Xu, et al., 2021).

### 2.5.9 MobileNetV3

*MobileNet* adalah salah satu bentuk arsitektur *Convolutional Neural Network* (CNN) yang dapat menangani perhitungan data yang besar. Perbedaan utama antara *MobileNet* dan arsitektur CNN tradisional terletak pada cara lapisan konvolusi digunakan, khususnya dalam mengatur ketebalan filter sesuai dengan ketebalan citra *input*. *MobileNet* menjadi salah satu arsitektur CNN yang digunakan untuk mengatasi komputasi yang kompleks. Peneliti merancang arsitektur CNN untuk dapat digunakan di ponsel, sesuai dengan namanya yaitu *Mobile*. *MobileNet* memisahkan proses konvolusi menjadi dua tahap, yaitu *depthwise convolution* dan *pointwise convolution* seperti pada Gambar 9 berikut.

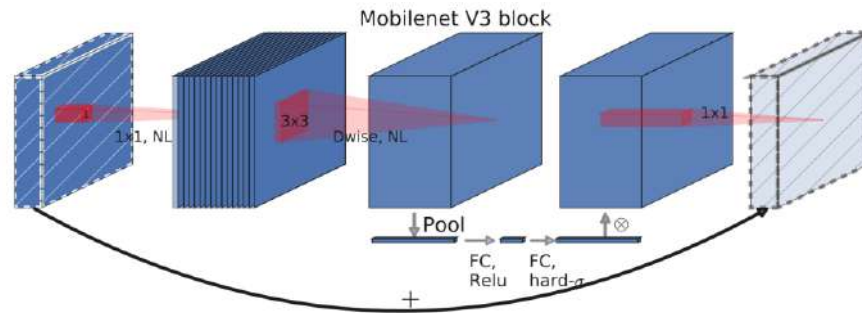


Gambar 9. *Standard Convolution*, *Depthwise Convolution*, dan *Pointwise Convolution*

(Howard, et al., 2017)

*Standard convolution* (a) dibagi menjadi dua lapisan yaitu *depthwise convolution* (b) dan *pointwise convolution* (c) untuk membuat *depthwise separable filter*.

*MobileNet* merilis versi ketiga pada November 2019. *MobileNetV3* mengombinasikan *depthwise separable convolution*, *inverted residuals*, *linear bottlenecks*, *SE module*, dan *H-swish function* yang meningkatkan akurasi seri dari *MobileNet*. Adapun struktur dasar dari *MobileNetV3* ditunjukkan pada Gambar 10.



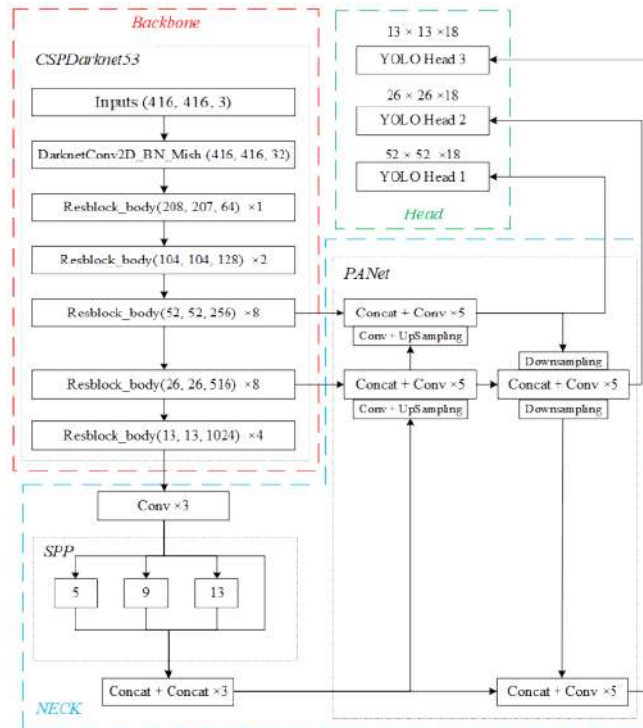
Gambar 10. Struktur MobileNetV3

(Howard, et al., 2019)

MobileNetV3 memiliki dua model yaitu MobileNetV3-Large dan MobileNetV3-Small yang masing-masing penggunaannya didasarkan kebutuhan dan ketersediaan sumber daya. Perbedaan keduanya yaitu terletak pada beberapa layer pada MobileNetV3-Large yang tidak digunakan pada MobileNetV3-Small yang berpengaruh pada kecepatan latensi dan akurasi. MobileNetV3-Large memiliki tingkat akurasi lebih baik dibandingkan MobileNetV3-Small, tetapi memiliki latensi yang lebih besar dibandingkan MobileNetV3-Small (Howard, et al., 2019).

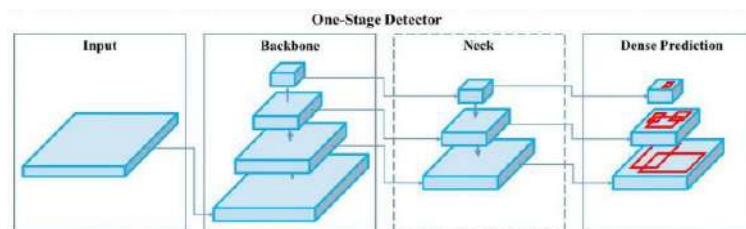
#### 2.5.10 YOLOv4

YOLOv4, yang merupakan singkatan dari "*You Only Look Once version 4*" adalah salah satu model deteksi objek yang sangat canggih dalam bidang visi komputer dan *deep learning*. YOLO diciptakan oleh Joseph Redmon di tahun 2016. YOLOv4 merupakan evolusi dari versi-versi sebelumnya (YOLO, YOLOv2, dan YOLOv3) dan telah mengatasi beberapa keterbatasan yang ada, seperti akurasi deteksi yang lebih baik dan kecepatan inferensi yang lebih tinggi. Model ini menggunakan *Convolution layers* 3x3 dan 1x1 dengan kinerja yang baik. Adapun arsitektur dari YOLOv4 ditunjukkan pada Gambar11 berikut.



Gambar 11. Arsitektur YOLOv4  
(Bubbling, 2020).

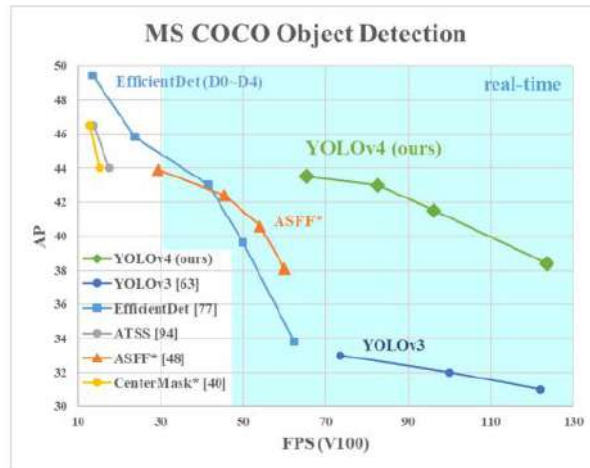
Arsitektur YOLOv4 terdiri dari tiga komponen utama yaitu *backbone*, *neck*, dan *predicting head*. *Backbone* mengekstrak fitur data dari *input* gambar. *Neck* mengumpulkan dan menggabungkan fitur data *multiscale* untuk menghasilkan tiga fitur peta dengan skala berbeda. Dari fitur peta yang dihasilkan tersebut, *predicting head* akan mendeteksi objek. YOLOv4 menggunakan CSPDarknet53 sebagai *backbone*, SPP dan PANet sebagai *neck*, dan YOLO sebagai *predicting head*.



Gambar 12. YOLOv4 *Object Detector Architecture*  
(Bochkovskiy, Wang, & Liao, 2020).

Dalam deteksi objek, presisi yang tinggi bukanlah satu-satunya hal yang dibutuhkan. Untuk itu, diperlukan model yang dapat berjalan dengan mudah

di perangkat dengan *resource* rendah. YOLOv4 memiliki kecepatan optimal (FPS) dan akurasi (presisi rata-rata) dalam deteksi objek. Hal ini menjadi peningkatan besar dari model deteksi objek sebelumnya yang hanya memiliki performa tinggi atau kecepatan inferensi tinggi. Berikut ini merupakan perbandingan YOLOv4 dengan detektor objek canggih lainnya.



Gambar 13. Perbandingan YOLOv4 dengan model lain (Bochkovskiy, Wang, & Liao, 2020).

Dari Gambar 13, YOLOv4 memiliki kecepatan dua kali lipat dibandingkan dengan *EfficientNet* dengan kinerja yang sebanding. YOLOv4 juga meningkatkan AP dan FPS dari YOLOv3 masing-masing sebesar 10% dan 12% (Bochkovskiy, Wang, & Liao, 2020).

## 2.6 Confusion Matrix

Evaluasi untuk mengukur kinerja model yang telah dibuat pada penelitian sangat diperlukan. Salah satu teknik yang digunakan dalam *supervised learning* adalah *confusion matrix*. *Confusion matrix* atau *error matrix* merupakan salah satu teknik untuk membandingkan data prediksi dari hasil pemodelan dengan data aktual atau data sebenarnya. *Confusion matrix* menyatakan klasifikasi jumlah data benar dan salah (Normawati & Prayogi, 2021).



Tabel 2. *Confusion Matrix*

		<i>Actual</i>					
		<i>Classes</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>
<i>Predicted</i>	<i>A</i>		TP <sub>A</sub>	FP <sub>A</sub>	FP <sub>A</sub>	FP <sub>A</sub>	FP <sub>A</sub>
				FN <sub>B</sub>	FN <sub>C</sub>	FN <sub>D</sub>	FN <sub>E</sub>
	<i>B</i>		FP <sub>B</sub>	TP <sub>B</sub>	FP <sub>B</sub>	FP <sub>B</sub>	FP <sub>B</sub>
			FN <sub>A</sub>		FN <sub>C</sub>	FN <sub>D</sub>	FN <sub>E</sub>
	<i>C</i>		FP <sub>C</sub>	FP <sub>C</sub>	TP <sub>C</sub>	FP <sub>C</sub>	FP <sub>C</sub>
			FN <sub>A</sub>	FN <sub>B</sub>		FN <sub>D</sub>	FN <sub>E</sub>
	<i>D</i>		FP <sub>D</sub>	FP <sub>D</sub>	FP <sub>D</sub>	TP <sub>D</sub>	FP <sub>D</sub>
			FN <sub>A</sub>	FN <sub>B</sub>	FN <sub>C</sub>		FN <sub>E</sub>
	<i>E</i>		FP <sub>E</sub>	FP <sub>E</sub>	FP <sub>E</sub>	FP <sub>E</sub>	TP <sub>E</sub>
			FN <sub>A</sub>	FN <sub>B</sub>	FN <sub>C</sub>	FN <sub>D</sub>	

Tabel 2 menunjukkan nilai dari TP, FP, FN, dan TN untuk masing-masing kelas. Representasi hasil proses klasifikasi pada *confusion matrix* yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN) (Narkhede, 2018).

a. *True Positive* (TP)

*True Positive* (TP) menunjukkan data nilai prediksi positif dan nilai sebenarnya positif.

b. *True Negative* (TN)

*True Negative* (TN) menunjukkan data nilai prediksi negatif dan nilai sebenarnya negatif.

c. *False Positive* (FP)

*False Positive* (FP) menunjukkan data nilai prediksi positif dan nilai sebenarnya negatif.

d. *False Negative* (FN)

*False Negative* (FN) menunjukkan data nilai prediksi negatif dan nilai sebenarnya positif.

Melalui perhitungan *Confusion matrix* dapat dihasilkan nilai *accuracy*, *precision*, *recall*, dan *F1 Score* (Normawati & Prayogi, 2021). Pada penelitian ini digunakan keempat metrik tersebut untuk evaluasi model.

### 2.6.1 Accuracy

*Accuracy* menunjukkan tingkat keakuratan suatu model untuk melakukan klasifikasi. *Accuracy* adalah perbandingan antara prediksi benar (positif dan negatif) dengan seluruh data. Jadi, *accuracy* mengukur sejauh mana nilai prediksi mendekati nilai aktual yang sebenarnya. *Accuracy* dihitung dengan persamaan (1) berikut.

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \dots\dots\dots(1)$$

### 2.6.2 Recall

*Recall* menunjukkan tingkat keberhasilan model dari nilai positif yang ada. *Recall* dihitung dengan persamaan (2) berikut.

$$recall = \frac{TP}{TP+FN} \dots\dots\dots(2)$$

### 2.6.3 Precision

*Precision* menunjukkan seberapa akurat antara data aktual dengan hasil prediksi model. *Precision* dihitung dengan persamaan (3) berikut.

$$precision = \frac{TP}{TP+FP} \dots\dots\dots(3)$$

### 2.6.4 F1 Score

F1 Score menunjukkan nilai perbandingan rata-rata dari *precision* dan *recall*. Nilai *F1 Score* dapat dihitung dengan persamaan (4) berikut.

$$F1\ Score = 2 \times \frac{recall \times precision}{recall + precision} \dots\dots\dots(4)$$



### 3.2 Alat dan Bahan

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

Perangkat keras yang digunakan dalam penelitian ini ditunjukkan pada Tabel 4 berikut.

Tabel 4. Alat Pendukung (*Hardware*)

No	Perangkat	Spesifikasi
1	Laptop	Processor 11th Gen Intel® Core™ i7-11800H @2.30GHz, RAM 16GB, SSD 512GB, Grapichs NVIDIA® GeForce RTX™ 3060 Laptop GPU (6 GB GDDR6 dedicated), Wireless Intel® Wi-Fi CERTIFIED 6™ AX201 (2×2) dan Bluetooth® 5 combo ( <i>Supporting Gigabit data rate</i> ).
2	Smartphone	Chip Apple A15 Bionic, Pengambilan gambar pada kecepatan 60 fps dengan format MOV.

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

Perangkat lunak yang digunakan dalam penelitian ini yaitu:

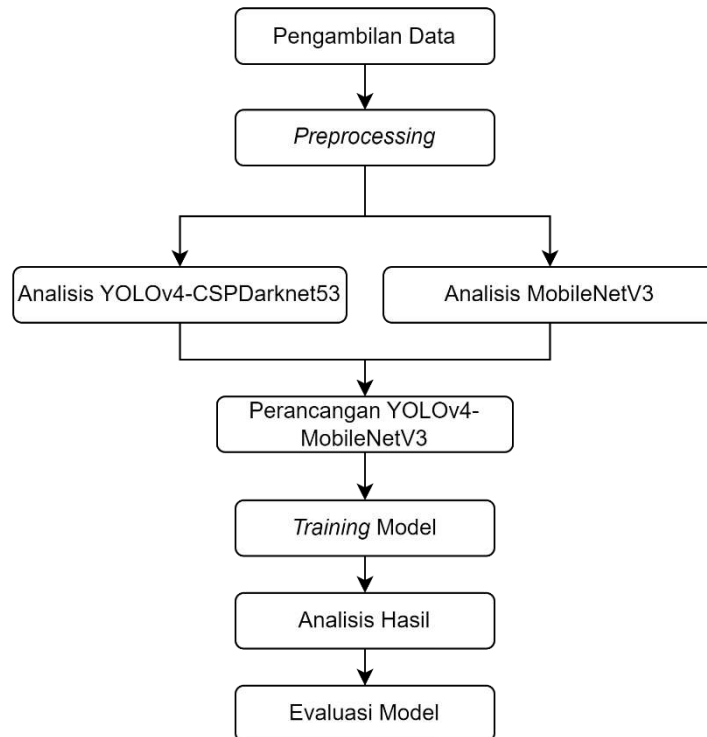
- Sistem Operasi : *Windows 11 Home Single 64-bit*
- Bahasa Pemrograman : *Python*
- *Text Editor* : *Google Colaboratory*
- Anotasi Objek : *LabelImg*
- *Web Browser* : *Google Chrome*
- Penyimpanan Data : *Google Drive*

#### c. Bahan

Bahan yang digunakan dalam penelitian ini adalah gambar kendaraan dengan berbagai jenis plat yang berwarna merah, putih, hitam, kuning, dan non plat di Jl. Raden Intan, Kota Bandar Lampung pada 30 November 2023 hingga 7 Desember 2023.

### 3.3 Alur Kerja Penelitian

Adapun alur kerja penelitian ini dapat dilihat pada Gambar 14 berikut.



Gambar 14. Alur Penelitian

#### 3.3.1 Pengambilan Data

Penelitian ini menggunakan data yang diperoleh dari pengambilan gambar secara langsung di Jl. Raden Intan, Kota Bandar Lampung, Lampung. Pengambilan data dilakukan dengan cara mengambil gambar masing-masing objek dengan kamera belakang pada cuaca cerah. Data citra yang digunakan memiliki ekstensi JPG atau JPEG (*Joint Photographic Experts Groups*).

#### 3.3.2 Preprocessing

Tahap *preprocessing* yang dilakukan pada penelitian ini adalah *extract*, *crop*, *resize*, dan anotasi objek. *Preprocessing* bertujuan untuk mempersiapkan data sehingga diperoleh citra terbaik sebelum diproses ke tahap selanjutnya. Data yang berupa video akan diekstrak terlebih dahulu menjadi gambar, kemudian

di-*crop* untuk membatasi area yang akan dianotasi. Proses *resize* bertujuan untuk memastikan bahwa seluruh citra memiliki rasio piksel yang sama sehingga tidak membebankan proses pelatihan model. Sedangkan proses anotasi objek merupakan proses pembuatan label pada gambar dengan memberikan *bounding box* beserta nama kelas pada objek yang akan dideteksi.

Proses anotasi objek atau pelabelan dilakukan untuk membantu proses pelatihan model agar lebih akurat dalam mendeteksi objek pada citra. Proses pelabelan dibagi menjadi lima kelas, yaitu plat nomor kendaraan berwarna dasar hitam, putih, kuning, merah, dan non plat. Untuk *class* non plat didefinisikan sebagai kendaraan yang tidak menggunakan TNKB. Selain itu, ketika ada objek lain yang terdeteksi selain plat kendaraan, seperti misalnya stiker pada kendaraan, akan masuk ke dalam kategori non plat.

### 3.3.3 Analisis YOLOv4-CSPDarknet53

Penelitian ini menggunakan YOLOv4 untuk dikembangkan dengan fitur ekstraktor MobileNetV3. Sebelum merancang YOLOv4-MobileNetV3, akan dilakukan analisis YOLOv4 dengan fitur ekstraktor asli yaitu CSPDarknet53. Analisis ini meliputi layer dalam CSPDarknet53 yang memproses citra sehingga menghasilkan fitur map untuk melakukan deteksi dan klasifikasi plat kendaraan.

### 3.3.4 Analisis MobilenetV3

MobileNetV3 merupakan fitur ekstraktor yang memiliki *Bottleneck*. Di dalam *bottleneck* tersebut terdapat *depthwise separable convolution* yaitu *depthwise convolution* dan *pointwise convolution*. Proses konvolusi pada *bottleneck* diawali dengan konvolusi  $1 \times 1$  pada *input* citra lalu dilanjutkan dengan konvolusi  $3 \times 3$ . Layer-layer tersebut akan memproses citra untuk menghasilkan fitur map. Pada penelitian ini akan dijelaskan proses penyesuaian MobileNetV3 dengan model YOLOv4.

### 3.3.4.1 Pointwise Convolution

Pada proses *layer pointwise convolution* terdapat *batch norm* dan ReLU. *Layer pointwise convolution* menerima *input* dalam bentuk fitur map yang dihasilkan dari *layer depthwise convolution*. Selanjutnya, pada *layer* ini terjadi proses konvolusi antara fitur map dari *layer* sebelumnya dengan ukuran  $256 \times 256$  menggunakan filter  $1 \times 1$ . Sebanyak 128 filter digunakan dalam operasi ini, dan *output* yang dihasilkan berupa fitur map dengan ukuran  $128 \times 128$ .

### 3.3.4.2 Depthwise Convolution

Pada proses *layer depthwise convolution* terdapat *batch norm* dan ReLU. *Layer* ini menerima *input* dalam bentuk fitur map dari *layer bottleneck*. Di dalamnya, terdapat proses konvolusi antara fitur map hasil dari *layer* sebelumnya yang berukuran  $512 \times 512$  dengan filter berukuran  $3 \times 3$ . Ukuran fitur map pada *input* YOLOv4 yaitu  $416 \times 416$ , maka pada *depthwise convolution* terjadi proses *average pooling* pada fitur map yang semula berukuran  $512 \times 512$  menjadi  $416 \times 416$  sesuai *input* YOLOv4.

### 3.3.5 Perancangan YOLOv4-MobileNetV3

Pada penelitian ini digunakan algoritma YOLOv4 dengan fitur ekstraktor berupa CSPDarknet53 dan MobileNetV3. Algoritma YOLOv4 akan dikembangkan dengan mengganti fitur ekstraktor aslinya yaitu CSPDarknet53 menjadi MobileNetV3 sebagai pendeteksi objek untuk kemudian akan diklasifikasikan dengan mengenali jenis dari objek tersebut.

### 3.3.6 Training Model

Tahap *training* dilakukan dengan melatih model dengan data yang telah melalui proses *preprocessing*. Dari tahap ini akan dihasilkan model dengan *weight* terbaik dan akan digunakan untuk tahap *testing*.

### 3.3.7 Analisis Hasil

Pada tahap ini akan dilakukan analisis dari hasil *testing* menggunakan model yang telah melalui tahap *training*. Hal ini bertujuan untuk mengetahui seberapa baik model melakukan deteksi dan klasifikasi plat kendaraan pada *test set* dari *dataset* penelitian.

### 3.3.8 Evaluasi Model

Pada tahap ini dilakukan evaluasi model yang mencakup penilaian terhadap performa model berdasarkan hasil analisis yang telah dilakukan pada tahap sebelumnya. Metrik yang digunakan untuk evaluasi performa model adalah metrik *accuracy*, *precision*, *recall*, dan *F1 score*. Pada tahap ini juga dilakukan perbandingan performa akurasi dan waktu komputasi antara YOLOv4-CSPDarknet53 dan YOLOv4-MobileNetV3.



## V. PENUTUP

### 5.1 Simpulan

Berdasarkan penelitian yang telah dilakukan mengenai pengembangan YOLOv4 dengan fitur ekstraktor MobileNetV3 untuk deteksi dan klasifikasi plat kendaraan didapatkan beberapa kesimpulan sebagai berikut.

1. Penelitian ini mengembangkan YOLOv4 dengan fitur ekstraktor MobileNetV3, dengan melakukan penyesuaian arsitektur pada *backbone*. Model dapat melakukan deteksi dan klasifikasi plat kendaraan berdasarkan warna, dimana model mampu membuat *bounding box* pada setiap objek dalam gambar.
2. Performa model YOLOv4-CSPDarknet53 dalam melakukan deteksi dan klasifikasi plat kendaraan menghasilkan rata-rata akurasi sekitar 96.93% sedangkan YOLOv4-MobileNetV3 memiliki rata-rata akurasi sekitar 97.54%. Hal ini menunjukkan bahwa YOLOv4-CSPDarknet53 memiliki rata-rata akurasi 0.61% lebih baik dibandingkan dengan YOLOv4-MobileNetV3.
3. Performa model YOLOv4-CSPDarknet53 dalam melakukan deteksi dan klasifikasi plat kendaraan membutuhkan waktu rata-rata 0.418 s, sedangkan YOLOv4-MobileNetV3 membutuhkan waktu rata-rata 0.133 s. YOLOv4-MobileNetV3 cenderung membutuhkan waktu yang lebih konsisten dalam melakukan deteksi baik dalam kondisi lengang maupun padat. Dengan demikian, waktu komputasi YOLOv4-MobileNetV3 menunjukkan waktu performa yang lebih baik dibandingkan YOLOv4-CSPDarknet53.

## 5.2 Saran

Saran dari penelitian ini antara lain.

1. Menambah variasi *dataset* dengan kualitas gambar yang lebih baik dan jumlah data yang seimbang pada setiap *class*.
2. Pada penelitian selanjutnya dapat memodifikasi model sehingga dapat membaca karakter yang terdapat dalam plat kendaraan.
3. Menggunakan metode lain dalam *deep learning* untuk melakukan deteksi dan klasifikasi plat kendaraan.

## DAFTAR PUSTAKA

- Aprilino, A., & Al Amin, I. H. (2022). Implementasi Algoritma YOLO dan Tesseract OCR Pada Sistem Deteksi Plat Nomor Otomatis. *Jurnal TEKNOINFO*, 16(1), 54-59.
- Arnita, Marpaung, F., Aulia, F., Suryani, N., & Nabila, R. C. (2022). *Computer Vision dan Pengolahan Citra Digital*. Surabaya: Pustaka Aksara.
- Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv preprint arXiv:2004.10934*.
- Bubbling. (2020). *Object Detection YOLOv4 (tensorflow2)*. Retrieved from [https://blog.csdn.net/weixin\\_44791964/article/details/106533581](https://blog.csdn.net/weixin_44791964/article/details/106533581)
- Cholissodin, I., & Soebroto, A. A. (2020). *AI, Machine Learning & Deep Learning*. Malang.
- Elharrouss, O., Akbari, Y., Almaadeed, N., & Al-Maadeed, S. (2022). Backbones-Review: Feature Extraction Networks for Deep Learning and Deep Reinforcement Learning Approaches. *arXiv preprint arXiv:*, 2206.08016.
- Fadjeri, A., Saputra, B. A., Adri Ariyanto, D. K., & Kurniatin, L. (2022). Karakteristik Morfologi Tanaman Selada Menggunakan Pengolahan Citra Digital. *Jurnal Ilmiah Sinus (JIS)*, Vol, 20(2).
- Fauzy, A. R. (2019). Implementasi Metode Region Convolutional Neural Network Dalam Mendiagnosa Anomali Pneumonia Pada Foto Thorax.
- Fu, J., Chen, X., & Lv, Z. (2022). Rail Fastener Status Detection Based on MobileNet-YOLOv4. *Electronics*, 11(22), 3677.

- Gholamalinezhad, H., & Khosravi, H. (2020). Pooling Methods in Deep Neural Networks, a Review. *arXiv: 2009.07485*.
- Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., . . . Adam, H. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv:1704.04861v1*.
- Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., . . . Adam, H. (2019). Searching for MobileNetV3. *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1314-1324.
- Huang, S., He, Y., & Chen, X.-a. (2021). M-YOLO: A Nighttime Vehicle Detection Method Combining Mobilenet v2 and YOLO v3. *IOP Publishing*, Vol. 1883, No. 1, p. 012094.
- Ikhsan, D., Utami, E., & Wibowo, F. W. (2020). Metode Klasifikasi Mutu Greenbean Kopi Arabika Lanang Dan Biasa Menggunakan K-Nearest Neighbor Berdasarkan Bentuk. *Jurnal Ilmiah Sinus (JIS)*, 18(2), 1-8.
- Indra, D., Herman, & Budi, F. S. (2020). Implementasi Sistem Penghitung Kendaraan Otomatis Berbasis Computer Vision. *Komputika: Jurnal Sistem Komputer*, 12(1), 53-62.
- Jamil, N., Fadhil, M. H., Hamzah, R., & Ramli, M. I. (2021). Comparing MobileNet-SSD and YOLO v3 Learning Architecture for Real-time Driver's Fatigue Detection. *International Journal of Academic Research in Business and Social Sciences*, 11(12), 2538–2549.
- Karpathy, A. (2018). *CS231n Convolutional Neural Network for Visual*. Stanford University. Retrieved from <https://cs231n.github.io> pada 23 November 2023.
- Kusuma, D. H., & Mauizah. (2023). Deteksi Lampu Lalu Lintas Menggunakan YOLO untuk Autonomous Car. *JCENIM*, 1(1), 21-27.
- Liao, X., Lv, S., Li, D., Luo, Y., Zhu, Z., & Jiang, C. (2021). YOLOv4-MN3 for PCB Surface Defect Detection. *applied sciences*, 11(24), 11701.

- Lin, H., Deng, J. D., Albers, D., & Siebert, F. W. (2020). Helmet Use Detection of Tracked Motorcycles Using CNN-Based Multi-Task Learning. *IEEE Access*, 8, 162073-162084.
- Liu, Y., Wang, Q., Zhang, H., Liu, Y., & Zhao, K. (2022). Real-Time Defect Detection Of Metal Surface Based on Improved YOLOv4. *International Journal of Innovative Computing, Information and Control*, 18(4), 1329-1338.
- MathWorks. (2017). *Introduction to Deep Learning: What Are Convolutional Neural*. Retrieved from Mathworks pada 12 Oktober 2023: [www.mathworks.com](http://www.mathworks.com)
- Mulyanto, A., Jatmiko, W., Mursanto, P., Prasetyawan, P., & Borman, R. I. (2021). A New Indonesian Traffic Obstacle Dataset and Performance Evaluation of YOLOv4 for ADAS. *Journal of ICT Research and Applications*, 14(3), 286-298.
- Narkhede, S. (2018). *Understanding Confusion Matrix*. Retrieved from Towards Data Science 27 Nov 2023: <https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62>
- Normawati, D., & Prayogi, S. A. (2021). Implementasi Naïve Bayes Classifier Dan Confusion Matrix Pada Analisis Sentimen Berbasis Teks Pada Twitter. *Jurnal Sains Komputer & Informatika (J-SAKTI)*, 5(2), 697-711.
- Pramana, C. G., Khrisne, D. C., & Sastra, N. P. (2021). Rancang Bangun Object Detection Pada Robot Soccer Menggunakan Metode Single Shot Multibox Detector (SSD MobileNetV2). *Jurnal SPEKTRUM*, Vol, 8(2).
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. *Proceedings of the IEEE conference on computer vision and pattern recognition*.
- Sholehurrohman, R. (2021). Pengembangan YOLOv3 Dengan Fitur Ekstraktor MobileNetV2 Untuk Deteksi Dan Klasifikasi Kendaraan Bergerak. *Skripsi*.
- Wu, Y.-c., & Feng, J.-w. (2017). Development and Application of Artificial Neural Network. *Wireless Pers Commun*, 102, 1645-1656.

- Xu, P., Li , Q., Zhang , B., Wu, F., Zhao, K., Du, X., . . . Zhong, R. (2021). On-Board Real-Time Ship Detection in HISEA-1 SAR Images Based on CFAR and Lightweight Deep Learning. *Remote Sensing*, 13(10), 1995.
- Yuan, D., & Xu, Y. (2021). Lightweight Vehicle Detection Algorithm Based on Improved YOLOv4. *Engineering Letters*, 29(4).
- Zufar, M., & Setiyono, B. (2016). Convolutional Neural Networks untuk Pengenalan Wajah Secara Real-Time. *Jurnal Sains dan Seni ITS*, 5(2), 128862.