

**PENERAPAN ARSITEKTUR MOBILENET DAN ALGORITMA  
SOFTNMS PADA FASTER R-CNN UNTUK DETEKSI  
PENGGUNA JALAN**

**SKRIPSI**

**Oleh :  
TEGAR ADITIA PRATAMA  
NIM. 16650061**



**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM  
MALANG  
2021**

**PENERAPAN ARSITEKTUR MOBILENET DAN ALGORITMA  
SOFTNMS PADA FASTER R-CNN UNTUK DETEKSI  
PENGGUNA JALAN**

**SKRIPSI**

**Diajukan kepada:  
Universitas Islam Negeri Maulana Malik Ibrahim  
Untuk memenuhi Salah Satu Persyaratan dalam  
Memperoleh Gelar Sarjana Komputer (S.Kom)**

**Oleh:  
TEGAR ADITIA PRATAMA  
NIM. 16650061**

**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM  
MALANG  
2021**

**LEMBAR PERSETUJUAN**

**PENERAPAN ARSITEKTUR MOBILENET DAN ALGORITMA  
SOFTNMS PADA FASTER R-CNN UNTUK DETEKSI  
PENGGUNA JALAN**

**SKRIPSI**

**Oleh:**  
**TEGAR ADITIA PRATAMA**  
**NIM. 16650061**

Telah Diperiksa dan Disetujui untuk Diuji  
Tanggal : 4 Juni 2021

Pembimbing I



**Irwan Budi Santoso, M.Kom**  
**NIP. 19770103 201101 1 004**

Pembimbing II



**Fatchurrohman, M.Kom**  
**NIP. 19700731 200501 1 002**

Mengetahui,  
Ketua Jurusan Teknik Informatika  
Fakultas Sains dan Teknologi  
Universitas Islam Negeri Maulana Malik Ibrahim Malang



**Dr. Cahyo Crys dian**  
**NIP. 19740424 200901 1 008**

## LEMBAR PENGESAHAN

### PENERAPAN ARSITEKTUR MOBILENET DAN ALGORITMA SOFTNMS PADA FASTER R-CNN UNTUK DETEKSI PENGGUNA JALAN

#### SKRIPSI

Oleh:  
**TEGAR ADITIA PRATAMA**  
NIM. 16650061

Telah Dipertahankan di Depan Dewan Penguji  
Dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan untuk Memperoleh  
Gelar Sarjana Komputer (S.Kom)  
Pada Tanggal 10 Juni 2021

#### Susunan Dewan Penguji

- |                       |  |   |
|-----------------------|--|---|
| 1. Penguji Utama      | : <u>M. Ainul Yaqin, M.Kom</u><br>NIP. 19761013 200604 1 004                 | (  ) |
| 2. Ketua Penguji      | : <u>Khadijah Fahmi Hayati Holle, M.Kom</u><br>NIDT. 19900626 20160801 2 077 | (  ) |
| 3. Sekretaris Penguji | : <u>Irwan Budi Santoso, M.Kom</u><br>NIP. 19770103 201101 1 004             | (  ) |
| 4. Anggota Penguji    | : <u>Fatchurrohman, M.Kom</u><br>NIP. 19700731 200501 1 002                  | (  ) |

Mengetahui,  
Ketua-Jurusan Teknik Informatika  
Fakultas Sains dan Teknologi  
Universitas Islam Negeri Maulana Malik Ibrahim Malang

  
  
Dr. Cahyo Crysdian  
NIP. 19740424 200901 1 008

## PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini :

Nama : Tegar Aditia Pratama  
NIM : 16650061  
Fakultas : Sains dan Teknologi  
Jurusan : Teknik Informatika  
Judul Skripsi : Penerapan Arsitektur MobileNet dan Algoritma  
SoftNMS pada Faster R-CNN untuk Deteksi  
Pengguna Jalan

Menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambilan data, tulisan, atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka. Apabila di kemudian hari terbukti atau dapat dibuktikan skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan saya tersebut.

Malang, 4 Juni 2021

Yang membuat pernyataan.



Tegar Aditia Pratama  
NIM. 16650061

## HALAMAN MOTTO

“Urip rasah digawe susah.  
Seng penting ora maksiat.  
Yowis ngono wae.”

-Gus Baha

## HALAMAN PERSEMBAHAN

الْحَمْدُ لِلَّهِ رَبِّ الْعَالَمِينَ

Segala puji syukur kehadirat Allah SWT, berkat hidayah dan inayahnya karya tulis ini dapat terselesaikan. Shalawat serta salam semoga tetap tercurahkan kepada Nabi Muhammad SAW.

Karya tulis ini penulis persembahkan kepada Ayah, Ibu, Adik, Kakek, dan Nenek di Ponorogo, yang telah memberikan motivasi dan dukungan yang tidak terduga dari penulis menempuh kuliah sampai karya tulis ini terselesaikan, dan sampai seterusnya.

Untuk kedua pembimbing penulis, Bapak Irwan Budi Santoso, M.Kom dan Bapak Fatchurrohman, M.Kom yang telah memberikan motivasi, saran, dan kritik terhadap penulis sampai terselesaikannya karya tulis ini.

Untuk kedua penguji skripsi penulis, Bapak M. Ainul Yaqin, M.Kom dan Ibu Khadijah Fahmi Hayati Holle, M.Kom yang juga telah memberikan saran dan kritik untuk menjaga kualitas dari karya tulis ini.

Untuk segenap dosen dan staf akademik Teknik Informatika UIN Malang, yang telah membantu dan memberikan ilmu kepada penulis selama masa perkuliahan.

Untuk kawan-kawan mahasiswa Teknik Informatika UIN Malang, khususnya keluarga Andromeda. Terima kasih telah menjadi teman yang baik untuk penulis, dan membantu penulis selama perkuliahan.

Untuk teman-teman kontrakan muslimin yang telah menemani penulis selama 3 tahun lebih.

Untuk semua pihak yang pernah terlibat dalam kehidupan penulis, terima kasih telah membantu penulis dalam menjalani kehidupan dan membentuk penulis menjadi pribadi yang seperti sekarang.

## KATA PENGANTAR

*Assalamu 'alaikum Warohmatullaahi Wabarokaatuh.*

Segala puji bagi Allah SWT Tuhan yang Maha Esa, karena dengan rahmat dan karunia-Nya penulis dapat menyelesaikan skripsi yang berjudul Penerapan Arsitektur MobileNet dan Algoritma SoftNMS pada Faster R-CNN untuk Deteksi Pengguna dengan baik dan lancar. Shalawat serta salam selalu dipanjatkan kepada kepada junjungan kita nabiullah Muhammad SAW yang telah membawa umatnya keluar dari zaman jahilia menuju zaman yang terang benderang yaitu Islam rahmatan lil alamin.

Selama proses penyelesaian skripsi ini, penulis mendapatkan banyak dukungan, bantuan, serta bimbingan berbagai pihak. Oleh sebab itu dengan rasa syukur dan kelapangan hati penulis mengucapkan terima kasih dan doa yang tulus kepada:

1. Bapak Dr. Cahyo Crysdiyan, selaku Ketua Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.
2. Bapak Irwan Budi Santoso, M.Kom selaku dosen pembimbing I yang telah membimbing, memberikan arahan serta masukan sehingga penulis dapat menyelesaikan skripsi ini.
3. Bapak Fatchurrohman, M.Kom selaku dosen pembimbing II yang telah membimbing penulis menyelesaikan skripsi.
4. Bapak M. Ainul Yaqin, M.Kom dan Ibu Khadijah Fahmi Hayati Holle, M.T selaku dosen penguji telah memberikan masukan serta saran yang membangun sehingga seluruh proses seminar proposal hingga sidang skripsi berjalan dengan lancar.
5. Seluruh staf dan dosen jurusan Teknik Informatika secara langsung maupun tidak langsung terlibat dalam proses pengerjaan skripsi.
6. Bapak dan Ibu beserta keluarga yang telah memberi dukungan sehingga penulis diberi kemudahan dalam menyelesaikan skripsi ini.

7. Seluruh teman komunitas jurusan Teknik Informatika, teman se-angkatan, dan teman-teman kontrakan baik secara langsung maupun tidak langsung terlibat dalam proses pengerjaan skripsi.

Penulis menyadari bahwa masih banyak kekurangan dari laporan ini. Oleh karena itu, penulis memohon maaf atas segala kekurangan yang terjadi selama proses penyusunan skripsi ini. Semoga tugas akhir ini dapat memberikan kontribusi yang bermanfaat bagi penulis dan pembaca khususnya.

*Wassalamu'alaikum Warohmatullaahi Wabarokaatuh.*

Malang, 4 Juni 2021

Penulis

## DAFTAR ISI

HALAMAN JUDUL.....	i
LEMBAR PERSETUJUAN.....	ii
LEMBAR PENGESAHAN .....	iii
PERNYATAAN KEASLIAN TULISAN.....	iv
HALAMAN MOTTO .....	v
HALAMAN PERSEMBAHAN.....	vi
KATA PENGANTAR .....	vii
DAFTAR ISI.....	ix
DAFTAR GAMBAR .....	xii
DAFTAR TABEL.....	xiv
ABSTRAK.....	xvii
ABSTRACT.....	xviii
المخلص.....	xix
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang .....	1
1.2. Rumusan Masalah .....	4
1.3. Tujuan Penelitian .....	4
1.4. Manfaat Penelitian .....	5
1.5. Batasan Masalah.....	5
1.6. Sistematika Penulisan.....	5

BAB II STUDI PUSTAKA.....	7
2.1. Object Detection.....	7
2.2. Deep Learning.....	9
2.3. Artificial Neural Network (ANN).....	11
2.4. Convolutional Neural Network (CNN).....	12
2.4.1. Convolutional Layers.....	13
2.4.2. Pooling Layers.....	14
2.4.3. Fully Connected Layers.....	15
2.4.4. Softmax Function.....	15
2.5. Region based Convolutional Neural Network (R-CNN).....	16
2.6. Penelitian Terkait.....	19
BAB III METODOLOGI PENELITIAN.....	23
3.1. Prosedur Penelitian.....	23
3.2. Pengumpulan Data.....	24
3.3. Analisis Kebutuhan Sistem.....	28
3.3.1. Perangkat Lunak ( <i>Software</i> ).....	28
3.3.2. Perangkat Keras ( <i>Hardware</i> ).....	28
3.4. Desain Sistem.....	28
3.4.1. Input.....	29
3.4.2. Proses.....	32
3.4.3. Output.....	42

3.5. Pengujian Sistem.....	42
BAB IV UJI COBA DAN PEMBAHASAN .....	43
4.1. Skenario Uji Coba.....	43
4.2. Hasil Uji Coba.....	44
4.2.1. Akurasi .....	44
4.2.2. Kecepatan.....	61
4.3. Pembahasan.....	62
BAB V PENUTUP.....	69
5.1. Kesimpulan .....	69
5.2. Saran.....	70
DAFTAR PUSTAKA .....	71

## DAFTAR GAMBAR

<b>Gambar 2.1</b> Contoh deteksi objek (Girshick et al., 2014) .....	8
<b>Gambar 2.2</b> AI on <i>Nested Russian Doll</i> (Eda, 2020).....	9
<b>Gambar 2.3</b> pembelajaran tradisional vs <i>deep learning</i> (Andrew, 2015).....	10
<b>Gambar 2.4</b> Arsitektur ANN (Arden Dertat, 2019).....	12
<b>Gambar 2.5</b> Contoh arsitektur CNN (Arden Dertat, 2019) .....	13
<b>Gambar 2.6</b> 6x6 <i>input feature map</i> dan 3x3 <i>filter</i> .....	14
<b>Gambar 2.7</b> Convolution process of 6x6 input dengan 3x3 filter .....	14
<b>Gambar 2.8</b> Maxpooling layer 2x2 dan stride 2 .....	15
<b>Gambar 2.9</b> Contoh Arsitektur R-CNN (Girshick et al., 2014).....	16
<b>Gambar 2.10</b> Region Proposal.....	17
<b>Gambar 2.11</b> Ilustrasi <i>Selective Search</i> (Uijlings et al., 2013).....	17
<b>Gambar 3.1</b> Alur Penelitian .....	23
<b>Gambar 3.2</b> Contoh citra pengguna jalan .....	24
<b>Gambar 3.3</b> Citra mobil .....	24
<b>Gambar 3.4</b> Citra sepeda motor .....	24
<b>Gambar 3.5</b> Citra manusia .....	25
<b>Gambar 3.6</b> Menjalankan aplikasi ffmpeg pada terminal.....	26
<b>Gambar 3.7</b> Hasil proses mengubah video ke frame .....	26
<b>Gambar 3.8</b> Proses labeling menggunakan LabelImg.....	27
<b>Gambar 3.9</b> Desain Sistem .....	29
<b>Gambar 3.10</b> <i>Bounding Box</i> dari file anotasi.....	31
<b>Gambar 3.11</b> Skema Faster R-CNN .....	32
<b>Gambar 3.12</b> Arsitektur MobileNet.....	34

<b>Gambar 3.13</b> <i>Anchor</i> pada <i>sliding window</i> .....	35
<b>Gambar 3.14</b> <i>Flowchart</i> SoftNMS (Bodla et al., 2017) .....	36
<b>Gambar 3.15</b> Arsitektur klasifikasi.....	39
<b>Gambar 3.16</b> Tampilan citra prediksi .....	40
<b>Gambar 4.4</b> Grafik Pengaruh Nilai Epoch terhadap Akurasi .....	63
<b>Gambar 4.5</b> Grafik Pengaruh Learning Rate terhadap Akurasi.....	64

## DAFTAR TABEL

Tabel 2.1 Penelitian terkait perbaikan metode Faster R-CNN.....	22
Tabel 3.1 Perbandingan MobileNet dengan VGG16 (Howard et al., 2017).....	33
Tabel 3.2 Confusion Matrix proses testing .....	41
Tabel 4.1 Variasi learning rate dan epoch pengujian akurasi .....	43
Tabel 4.2 Tabel pengukuran kecepatan.....	44
Tabel 4.3 Hasil uji coba model 1 .....	45
Tabel 4.4 Confussion Matrix Model 1 (epoch 50, learning rate 0,1 ).....	45
Tabel 4.5 Hasil uji coba model 2 .....	45
Tabel 4.6 Confussion Matrix Model 2 (epoch 50, learning rate 0,01 ).....	46
Tabel 4.7 Hasil uji coba model 3 .....	46
Tabel 4.8 Confussion Matrix Model 3 (epoch 50, learning rate 0,001).....	47
Tabel 4.9 Hasil uji coba model 4 .....	47
Tabel 4.10 Confussion Matrix Model 4 (epoch 50, learning rate 0,0001).....	47
Tabel 4.11 Hasil uji coba model 5 .....	48
Tabel 4.12 Confussion Matrix Model 5 (epoch 100, learning rate 0,1).....	48
Tabel 4.13 Hasil uji coba model 6 .....	49
Tabel 4.14 Confussion Matrix Model 6 (epoch 100, learning rate 0,01).....	49
Tabel 4.15 Hasil uji coba model 7 .....	49
Tabel 4.16 Confussion Matrix Model 7 (epoch 100, learning rate 0,001).....	50
Tabel 4.17 Hasil uji coba model 8 .....	50
Tabel 4.18 Hasil uji coba model 9 .....	51
Tabel 4.19 Confussion Matrix Model 9 (epoch 150, learning rate 0,1).....	51
Tabel 4.20 Hasil uji coba model 10 .....	52

Tabel 4.21 Confussion Matrix Model 10 (epoch 150, learning rate 0,01).....	52
Tabel 4.22 Hasil uji coba model 11 .....	53
Tabel 4.23 Confussion Matrix Model 11 (epoch 150, learning rate 0,001).....	53
Tabel 4.24 Hasil uji coba model 12 .....	53
Tabel 4.25 Confussion Matrix Model 12 (epoch 150, learning rate 0,0001).....	54
Tabel 4.26 Hasil uji coba model 13 .....	54
Tabel 4.27 Confussion Matrix Model 13 (epoch 200, learning rate 0,1).....	55
Tabel 4.28 Hasil uji coba model 14 .....	55
Tabel 4.29 Confussion Matrix Model 14 (epoch 200, learning rate 0,01).....	55
Tabel 4.30 Hasil uji coba model 15 .....	56
Tabel 4.31 Confussion Matrix Model 15 (epoch 200, learning rate 0,001 ).....	56
Tabel 4.32 Hasil uji coba model 16 .....	57
Tabel 4.33 <i>Confussion Matrix</i> Model 16 ( <i>epoch 200, learning rate 0,0001</i> ).....	57
Tabel 4.34 Hasil uji coba model 17 .....	57
Tabel 4.35 Confussion Matrix Model 17 (epoch 250, learning rate 0,1 ).....	58
Tabel 4.36 Hasil uji coba model 18 .....	58
Tabel 4.37 Confussion Matrix Model 18 (epoch 250, learning rate 0,01 ).....	59
Tabel 4.38 Hasil uji coba model 19 .....	59
Tabel 4.39 Confussion Matrix Model 19 (epoch 250, learning rate 0,001 ).....	59
Tabel 4.40 Hasil uji coba model 20 .....	60
Tabel 4.41 <i>Confussion Matrix</i> Model 20 ( <i>epoch 250, learning rate 0,0001</i> ).....	60
Tabel 4.42 Pengujian ke-1 (ukuran citra 1920x1080 pixel).....	61
Tabel 4.43 Pengujian ke-2 (ukuran citra 1152x648 pixel).....	61
Tabel 4.44 Pengujian ke-3 (ukuran citra 576x324 pixel).....	62

Tabel 4.45 Perbandingan Nilai Akurasi.....	62
Tabel 4.46 Perbandingan kecepatan deteksi masing-masing ukuran citra.....	65

## ABSTRAK

Pratama, Tegar Aditia. 2021. Penerapan Arsitektur MobileNet dan Algoritma SoftNMS pada Faster R-CNN untuk Deteksi Pengguna Jalan. Jurusan Teknik Informatika, Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Pembimbing: (I) Irwan Budi Santoso, M.Kom, (II) Fatchurrohman, M.Kom.

---

Kata Kunci : Deteksi Pengguna Jalan, Deteksi Objek, SoftNMS, Faster R-CNN.

Deteksi kendaraan atau pengguna jalan memiliki peran penting dalam tugas *surveillance* pada *Intelligent Transportation System (ITS)*. Kecepatan dan akurasi diperhatikan untuk mengembangkan sistem lalu lintas cerdas tersebut. Dengan berkembang pesatnya *Convolutional Neural Network (CNN)*, metode deteksi pengguna jalan berbasis *vision* telah mencapai peningkatan yang signifikan dibandingkan dengan metode tradisional. Pada tahun 2014, metode *Region-based Convolutional Neural Network (R-CNN)* muncul sebagai solusi dari masalah yang terjadi pada CNN. Akan tetapi, metode tersebut membutuhkan biaya komputasi tinggi yang menyebabkan proses lama dalam deteksi. Kemudian pada tahun 2015, perbaikan pada metode R-CNN tersebut muncul metode Fast R-CNN. Namun, metode Fast R-CNN belum cukup cepat dalam melakukan deteksi secara *realtime*. Dari masalah tersebut, pada tahun 2015 dilakukan perubahan pada algoritma pengumpulan *region proposal* pada Fast R-CNN yang semula *selective search* menjadi *Region Proposal Network (RPN)* dan mendapatkan kecepatan deteksi dan akurasi yang lebih baik dari pendahulunya. Metode tersebut adalah Faster R-CNN. Masing-masing kecepatan R-CNN, Fast R-CNN, Faster R-CNN berturut-turut yaitu 50 detik, 2 detik, dan 0,2 detik. Dengan catatan, hardware yang digunakan pada penelitian tersebut menggunakan computer *high-end*. Pada pengukuran akurasi, keluarga R-CNN mendapat akurasi rata-rata diatas 70%. Dengan kehandalan yang dimiliki Faster R-CNN, penulis membangun sistem deteksi pengguna jalan. Namun, spesifikasi *hardware* yang lebih rendah yang dimiliki, penulis mengusulkan menggunakan arsitektur yang lebih ringan, yaitu menggunakan MobileNet pada *base network* dan menggunakan SoftNMS pada algoritma pengurangan duplikasi *region proposal*. Dari eksperimen yang telah dilakukan menggunakan dataset PascalVOC dan dataset yang diperoleh penulis, tingkat akurasi yang dihasilkan menggunakan metode Faster R-CNN yang diperbarui sebesar 96,84%. Sedangkan kecepatan deteksi yang diperoleh sebesar 0,16 citra/detik.

## ABSTRACT

Pratama, Tegar Aditia. 2021. Application of MobileNet Architecture and SoftNMS Algorithm on Faster R-CNN for Road User Detection. Thesis. Informatics Engineering Department of Science and Technology Faculty Islamic State University Maulana Malik Ibrahim Malang.  
Supervisor: (I) Irwan Budi Santoso, M.Kom, (II) Fatchurrohman, M.Kom.

---

Keyword : Road User Detection, Object Detection, SoftNMS, Faster R-CNN.

Detection of vehicles or road users has an important role in the surveillance task of the Intelligent Transportation System (ITS). Speed and accuracy are considered to develop such intelligent traffic systems. With the rapid development of Convolutional Neural Network (CNN), vision-based road user detection methods have achieved significant improvements compared to traditional methods. In 2014, the Region-based Convolutional Neural Network (R-CNN) method emerged as a solution to the problems that occurred in CNN. However, this method requires high computational costs which causes a long process of detection. Then in 2015, improvements to the R-CNN method appeared the Fast R-CNN method. However, the Fast R-CNN method is not fast enough to detect in real-time. From this problem, in 2015 changes were made to the regional proposal collection algorithm on Fast R-CNN from selective search to Region Proposal Network (RPN) and got better detection speed and accuracy than its predecessor. The method is Faster R-CNN. The respective speeds of R-CNN, Fast R-CNN, Faster R-CNN are 50 seconds, 2 seconds, and 0.2 seconds, respectively. On a note, the hardware used in this study uses a high-end computer. In measuring accuracy, the R-CNN family gets an average accuracy of above 70%. With the reliability possessed by Faster R-CNN, the authors built a road user detection system. However, due to the lower hardware specifications, the authors propose using a lighter architecture, namely using MobileNet on the base network and using SoftNMS on the proposed region duplication reduction algorithm. From the experiments that have been carried out using the PascalVOC dataset and the dataset obtained by the author, the level of accuracy generated using the updated Faster R-CNN method is 96.84%. While the detection speed obtained is 0.16 images/second.

## المخلص

فراثاما، تيغار أديتيا. ٢٠٢١. تطبيق هندسة MobileNet وخوارزمية SoftNMS على FasterR-CNN لاكتشاف مستخدم الطريق. أطروحة. قسم الهندسة المعلوماتية، كلية العلوم والتكنولوجيا، جامعة إسلام نجري مولانا مالك إبراهيم مالانج. المشرف: (١) إرون بودي سنتوسو. الماجستير. (٢) فتورحمن. الماجستير.

الكلمات الرئيسية: اكتشاف مستخدم الطريق، اكتشاف الأشياء، SoftNMS، أسرع R-CNN

يلعب اكتشاف المركبات أو مستخدم الطريق دورًا مهمًا في مهمة المراقبة لنظام النقل الذكي (ITS). تعتبر السرعة والدقة لتطوير أنظمة المرور الذكية هذه. مع التطور السريع للشبكة العصبية التلافيفية (CNN)، حققت طرق اكتشاف مستخدم الطريق القائمة على الرؤية تحسينات كبيرة مقارنة بالطرق التقليدية. في عام 2014، ظهرت طريقة الشبكة العصبية التلافيفية (R-CNN) القائمة على المنطقة كحل للمشاكل التي حدثت في شبكة CNN. ومع ذلك، تتطلب هذه الطريقة تكاليف حسابية عالية مما يؤدي إلى عملية اكتشاف طويلة. تم في عام 2015، ظهرت تحسينات على طريقة R-CNN بطريقة Fast R-CNN. ومع ذلك، فإن طريقة Fast R-CNN ليست سريعة بما يكفي للكشف عنها في الوقت الفعلي. من هذه المشكلة، تم إجراء تغييرات في عام 2015 على خوارزمية جمع الاقتراحات الإقليمية على Fast R-CNN من البحث الانتقائي إلى شبكة اقتراح المنطقة (RPN) وحصلت على سرعة ودقة اكتشاف أفضل من سابقتها. الطريقة هي أسرع R-CNN. السرعات الخاصة بكل من R-CNN و Fast R-CNN و Faster R-CNN هي 50 ثانية و 2 ثانية و 0.2 ثانية على التوالي. ملاحظة: الأجهزة المستخدمة في هذه الدراسة تستخدم جهاز كمبيوتر متطور. في قياس الدقة، تحصل عائلة R-CNN على متوسط دقة أعلى من 70٪. مع الموثوقية التي تمتلكها Faster R-CNN، بنى المؤلفون نظامًا لاكتشاف مستخدم الطريق. ومع ذلك، نظرًا لانخفاض مواصفات الأجهزة، يقترح المؤلفون استخدام بنية أخف، أي استخدام MobileNet على الشبكة الأساسية واستخدام SoftNMS في خوارزمية تقليل تكرار المنطقة المقترحة. من التجارب التي تم إجراؤها باستخدام مجموعة بيانات PascalVOC ومجموعة البيانات التي حصل عليها المؤلف، بلغ مستوى الدقة المتولد باستخدام طريقة Faster R-CNN المحدثة 96.84٪. بينما تبلغ سرعة الكشف التي تم الحصول عليها 0.16 صورة / ثانية.

# **BAB I**

## **PENDAHULUAN**

### **1.1. Latar Belakang**

Pengguna jalan di Indonesia terus mengalami kenaikan setiap tahunnya. Istilah pengguna jalan menurut UU Nomor 22 tahun 2009 tentang Lalu Lintas dan Angkutan Umum Pasal 1 ayat 27, Pengguna Jalan adalah orang yang menggunakan jalan untuk berlalu lintas. Peningkatan pengguna jalan tersebut dibuktikan dengan adanya kenaikan jumlah kendaraan bermotor. Menurut Badan Pusat Statistik pertumbuhan perTahunnya mencapai 6,13% dari tahun 2015 sampai tahun 2019. Jenis kendaraan bermotor dikelompokkan menjadi 4 jenis, yaitu: mobil penumpang, bis, mobil barang, dan sepeda motor. Tercatat pada tahun 2015, jumlah kendaraan bermotor berjumlah 105 juta. Kemudian pada tahun 2019, naik menjadi 133 juta.

Tetapi peningkatan pengguna jalan tersebut tidak diimbangi dengan pembangunan jalan yang memadai, sehingga timbul masalah seperti kemacetan dan tidak jarang berakibat pada kecelakaan. Untuk menangani masalah tersebut dapat menggunakan cara konvensional melalui pelebaran jalan dan menambah panjang jalan. Namun cara konvensional tersebut memerlukan rencana yang matang dan membutuhkan biaya yang tinggi. Sebagai alternatif dari biaya infrastruktur pelebaran jalan yang mahal, maka diciptakanlah Intelligent Transportation System (ITS). ITS merupakan sistem yang bertujuan untuk memberikan layanan inovatif yang berkaitan dengan berbagai moda transportasi, manajemen lalu lintas, dan membuat penggunaan transportasi yang lebih aman serta terkoordinasi.

Salah satu tugas dari ITS adalah melakukan pengawasan terhadap pengguna jalan di jalan raya. Dengan melakukan pengawasan tersebut akan memudahkan dalam manajemen arus lalu lintas sehingga kemacetan dapat dikurangi. Sebagai tugas penting di ITS, deteksi pengguna jalan membantu dalam hal klasifikasi jenis kendaraan, penghitungan kendaraan, identifikasi kecelakaan di lalu lintas, pengukuran kecepatan kendaraan, dan prediksi kepadatan arus lalu lintas. Mendeteksi pengguna jalan diperlukan ilmu object detection pada bidang computer vision yang membantu mendeteksi dan mengenali objek pada citra. Terdapat dua metode pada object detection, yang pertama adalah menggunakan metode deteksi yang tradisional dan yang kedua menggunakan metode CNN. Deteksi objek yang menggunakan metode tradisional diantaranya yaitu menggunakan deskriptor seperti *Scale Invariant Feature Transform* (SIFT) dan *Histogram of Oriented Gradient* (HOG) dan juga menggunakan *Support Vector Machine* (SVM) sebagai classifiernya. Pada metode tradisional, terdapat halangan yang dapat membuat kinerja deteksi menjadi buruk seperti efek bayangan, skenario yang kompleks, dan kendaraan yang terhalang. Dengan lahirnya *Convolutional Neural Network* (CNN) pada tahun 2012, masalah yang terjadi pada metode tradisional tersebut dapat diselesaikan dan mencapai kesuksesan untuk mendeteksi objek serta deteksi kendaraan (Huang et al., 2017).

Namun, pada saat menerapkan CNN ke dalam sistem deteksi kendaraan, ini akan menjadi sebuah tantangan dikarenakan sistem tersebut haruslah secara realtime. Banyak metode terbaru didasarkan pada modifikasi detektor objek berbasis CNN yang populer untuk meningkatkan kinerja hasil deteksi. Metode yang

menggunakan CNN untuk mendeteksi kendaraan ataupun pengguna jalan yang populer digunakan saat ini adalah Faster R-CNN dan YOLO.

Pada 2015, Ren et al. mengusulkan suatu perbaikan *detector* dari pendahulunya di keluarga *Region-based Convolutional Neural Network* (RCNN) yaitu Faster R-CNN. Faster R-CNN menggunakan Region Proposal Network (RPN) sebagai metode untuk menentukan region proposal, sedangkan sebelumnya pada RCNN dan Fast RCNN masih menggunakan selective search yang lamban dalam menemukan region proposal. Dari RCNN hingga Faster R-CNN, Faster R-CNN merupakan metode yang mendekati realtime untuk mendeteksi kendaraan ataupun objek. Kelebihan dari Faster R-CNN yaitu dapat mendeteksi dan mengenali objek yang lebih kecil dengan bagus. Tetapi Faster R-CNN masih perlu ditingkatkan kecepatannya sehingga bisa mendeteksi secara realtime.

YOLO merupakan detektor objek yang diupayakan untuk membuat sistem mendeteksi secara realtime. Karena YOLO tidak melakukan langkah proses mencari region proposal dan hanya memprediksi melalui sejumlah bounding box yang terbatas, YOLO mampu melakukan deteksi dengan sangat cepat. Namun, YOLO juga memiliki kelemahan diantaranya akurasi yang dihasilkan lebih rendah dari Faster R-CNN dan sulit untuk mengenali objek yang kecil dibandingkan dengan Faster R-CNN.

Dengan melihat kelebihan dalam akurasi Faster R-CNN, peneliti ingin mengembangkan sistem deteksi pengguna jalan menggunakan metode tersebut. Namun, dengan kemampuan hardware yang dimiliki oleh peneliti lebih rendah dibandingkan dengan hardware yang digunakan pada penelitian Faster R-CNN tersebut. Maka, penulis melakukan perubahan pada arsitektur Faster R-CNN

dengan menggunakan MobileNet karena arsitektur tersebut lebih ringan daripada VGG16 pada Faster R-CNN aslinya. Diharapkan hasil dari penelitian ini dapat memberikan manfaat khususnya dalam pengembangan sistem lalu lintas cerdas demi terciptanya ketertiban dan keamanan dalam berkendara di jalan raya. Keamanan berkendara dalam UU Nomor 22 Tahun 2009 merupakan aturan pemerintah yang harus ditaati oleh seluruh rakyat Indonesia. Sebagaimana yang telah disebutkan dalam Al-Qur'an pada surah An-Nisa ayat 59, kita diwajibkan untuk menaati pemerintah. Ayat tersebut berbunyi:

يَا أَيُّهَا الَّذِينَ آمَنُوا أَطِيعُوا اللَّهَ وَأَطِيعُوا الرَّسُولَ وَأُولَى الْأَمْرِ مِنْكُمْ

Artinya : “*Hai orang-orang yang beriman, ta’atilah Allah dan ta’atilah Rasul (Nya), dan ulil amri di antara kamu.*” (Q.S. An-Nisa’ : 59)

Pada surah tersebut, menurut Syaikh Wahbah Az Zuhaili dalam Tafsir Al Munir, *ulil amri* adalah pemimpin dan para ulama. Sedangkan menurut Ibnu Katsir, *ulil amri* itu bersifat umum baik pemerintah maupun ulama

## 1.2. Rumusan Masalah

Dari latar belakang diatas maka rumusan masalah dalam penelitian ini sebagai berikut:

1. Seberapa akurat metode Faster R-CNN yang diperbarui dalam mendeteksi pengguna jalan dengan studi kasus di Kota Malang?
2. Seberapa cepat metode Faster R-CNN yang diperbarui dalam mendeteksi pengguna jalan dengan studi kasus di Kota Malang?

## 1.3. Tujuan Penelitian

Tujuan yang ingin dicapai dalam penelitian ini yaitu sebagai berikut:

1. Mengetahui tingkat akurasi metode Faster R-CNN yang diperbarui dalam mendeteksi pengguna jalan dengan studi kasus di Kota Malang.
2. Mengetahui tingkat kecepatan metode Faster R-CNN yang diperbarui dalam mendeteksi pengguna jalan dengan studi kasus di Kota Malang.

#### **1.4. Manfaat Penelitian**

Manfaat dari penelitian ini adalah dapat dibuatnya sistem deteksi pengguna jalan yang memiliki kecepatan deteksi dan akurasi yang baik untuk keperluan *surveillance* pada *Intelligent Transportation System (ITS)*.

#### **1.5. Batasan Masalah**

Batasan yang ada pada penelitian ini adalah sebagai berikut:

- a. Data primer yang digunakan pada penelitian ini didapat dari PascalVOC2007.
- b. Data sekunder yang digunakan pada penelitian ini adalah data berupa video dan foto pengguna jalan di Kota Malang.
- c. Titik lokasi yang diteliti di kota Malang yaitu: Klojen dan Blimbing.
- d. Pengambilan data dilakukan pada siang hari saat kondisi cerah.
- e. Pengguna jalan yang diteliti ada 3 yaitu mobil, sepeda motor, dan manusia.

#### **1.6. Sistematika Penulisan**

Penelitian ini terdiri dari beberapa bab pembahasan yang disusun dalam laporan sebagai berikut:

- a. **Bab I Pendahuluan:** pada bab ini berisi tentang latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, dan batasan penelitian.
- b. **Bab II Studi Pustaka:** bab ini berisi tentang *literature review*, teori tentang metode Faster R-CNN, dan tentang penelitian terkait.

- c. **Bab III Metode Penelitian:** bab ini berisi perancangan dan pengumpulan data, desain sistem, implementasi Faster R-CNN yang diperbaiki dalam sistem yang akan dibangun.
- d. **Bab IV Uji Coba dan Pembahasan:** bab ini berisi skenario pengujian dan hasil uji coba dari metod Faster R-CNN yang diperbaiki dalam sistem yang dibangun.
- e. **Bab V Penutup:** bab ini berisi kesimpulan dan saran dari penelitian yang telah dilakukan.

## **BAB II**

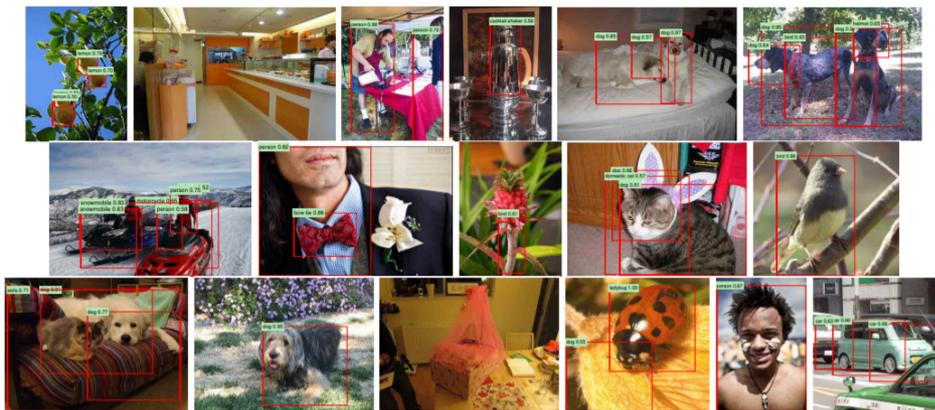
### **STUDI PUSTAKA**

*Image classification* (klasifikasi citra) dan *object detection* (deteksi objek) adalah istilah yang terkait dengan *computer vision* dan *image processing*. Klasifikasi citra adalah mengklasifikasikan citra ke dalam kelas tertentu, sedangkan deteksi objek adalah tentang mengenali, menemukan dan memberi label objek di semua kelas yang diprediksi terdapat objek (Jalied & Voronkov, 2016). Deteksi objek telah digunakan dalam banyak aplikasi, termasuk: robotika, keamanan, transportasi, dan elektronik konsumen. *Deep learning* mulai mendominasi *computer vision* sejak 2012, ketika Alex Krizhevsky memenangkan ImageNet Large Scale Visual Recognition Competition (ILSVRC) dengan merilis AlexNet. Bab ini menjabarkan konsep-konsep penting untuk penelitian ini dan arsitektur klasifikasi citra yang berbeda, dan teknik deteksi objek.

#### **2.1. Object Detection**

Dalam bidang *computer vision*, *classification* adalah topik penelitian paling terkenal yang bertujuan untuk mengklasifikasikan citra ke dalam salah satu dari banyak kategori berbeda. Berbeda dengan klasifikasi, *localization* menemukan objek tunggal di dalam citra. Berhubungan dengan masalah *localization* dan *classification* kita berakhir dengan kebutuhan untuk mendeteksi dan mengklasifikasikan beberapa objek pada saat yang bersamaan. Deteksi objek adalah bagaimana menemukan dan mengklasifikasikan sejumlah variabel objek pada citra, yang ditunjukkan pada Gambar 2.1.

Deteksi objek biasanya dianggap sebagai tugas paling mendasar dalam *computer vision*, yang menghadapi banyak tantangan. Yang pertama adalah jumlah variabel objek. Saat melatih model pembelajaran mesin, kita perlu merepresentasikan data ke dalam vektor yang berukuran tetap. Karena jumlah objek dalam citra tidak diketahui sebelumnya, kita tidak akan tahu jumlah output yang benar. Secara historis, jumlah variabel output telah ditangani menggunakan pendekatan berbasis *sliding window*, menghasilkan fitur ukuran tetap dari jendela itu untuk semua posisi yang berbeda. Setelah mendapatkan semua prediksi, ada yang dibuang dan ada yang digabungkan untuk mendapatkan hasil akhir. Tantangan berat lainnya adalah berbagai skala benda yang bisa dibayangkan. Saat melakukan klasifikasi sederhana, hanya objek yang menutupi sebagian besar citra yang perlu diklasifikasikan. Namun, untuk mendeteksi objek tertentu sekecil dan banyak piksel (atau persentase kecil dari gambar asli), metode *sliding window* sangat tidak efisien. Berbagai kesulitan lain adalah perubahan iluminasi, variasi sudut pandang, dan masalah deformasi.

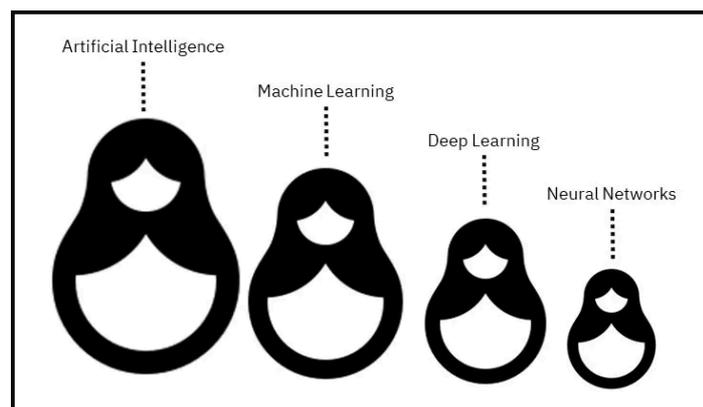


**Gambar 2.1** Contoh deteksi objek (Girshick et al., 2014)

Dua pendekatan deteksi objek klasik yang paling populer adalah metode Haar yang digunakan Viola-Jones dan model bagian yang dapat dideformasi (P. Felzenszwalb et al., 2008). Metode Viola-Jones bekerja dengan menghasilkan pengklasifikasi biner sederhana yang berbeda menggunakan fitur Haar. Penggolong ini dinilai dengan multi scale sliding window dalam kaskade dan jatuh lebih awal jika klasifikasi negatif. Metode DPM menggunakan fitur *Histogram of Oriented Gradients* (HOG) dan SVM untuk klasifikasi. Pada penelitian terbaru, detektor *deep learning* menggabungkan pembelajaran fitur dan pelatihan classifier bersama untuk mencapai kinerja yang memuaskan dengan kecepatan tinggi dan kecepatan pemrosesan yang tinggi.

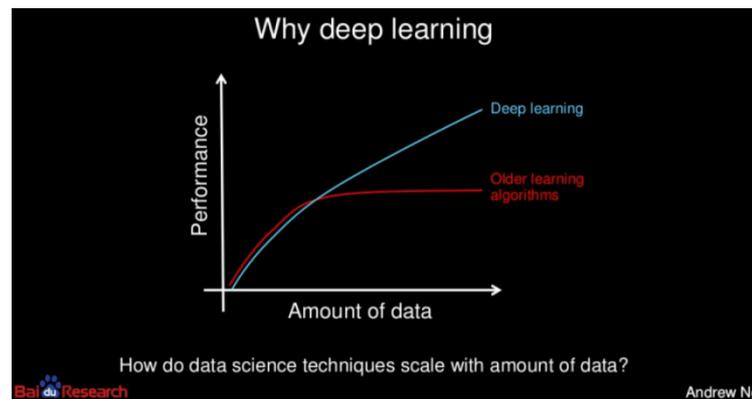
## 2.2. Deep Learning

Deep Learning merupakan sebuah subbidang *Machine Learning*, terkait dengan pelatihan model komputasi yang terdiri dari jaringan saraf tiruan berlapis-lapis. JST berlapis-lapis dikenal sebagai *Deep Neural Network* (DNN). Untuk lebih mudah memahami letak dari bidang *deep learning* dapat dilihat pada Gambar 2.2.



**Gambar 2.2** AI on *Nested Russian Doll* (Eda, 2020)

Metode *deep learning* didasarkan pada representasi pembelajaran (fitur) dari data, seperti teks, gambar, atau video, daripada menerapkan algoritma spesifik tugas. *Learning* dapat *unsupervised* atau *supervised*. Namun, sebagian besar sistem praktis menerapkan *supervised learning* untuk memanfaatkan manfaat *deep learning* (Krizhevsky et al., 2012). *Supervised learning*, pada dasarnya, berarti belajar dari data berlabel.



**Gambar 2.3** pembelajaran tradisional vs *deep learning* (Andrew, 2015)

Pada Gambar 2.3 menunjukkan kinerja metode *deep learning* meningkat dengan peningkatan jumlah data pelatihan yang bertentangan dengan metode pembelajaran tradisional yang jenuh dalam kinerja; karakteristik ini membuat metode pembelajaran yang mendalam menjadi terukur.

Aspek penting lain dari *deep learning* adalah pembelajaran fitur hierarkis. Pembelajaran fitur adalah ekstraksi fitur secara otomatis dari data mentah. Karena jaringan saraf yang dalam terdiri dari beberapa lapisan, itu melibatkan representasi pembelajaran pada berbagai tingkat kompleksitas dan berbagai tingkat abstraksi. Seperti disebutkan sebelumnya, lapisan awal mempelajari fitur tingkat rendah dan kemudian mengirimkannya ke lapisan selanjutnya. Lapisan-lapisan berikut kemudian mengumpulkan fitur tingkat tinggi berdasarkan fitur tingkat bawah yang

telah dipelajari sebelumnya. Fitur pembelajaran pada tingkat abstraksi dan kompleksitas yang berbeda memfasilitasi model *deep learning* untuk mempelajari fungsi kompleks tanpa tergantung pada fitur buatan manusia (Bengio, 2009).

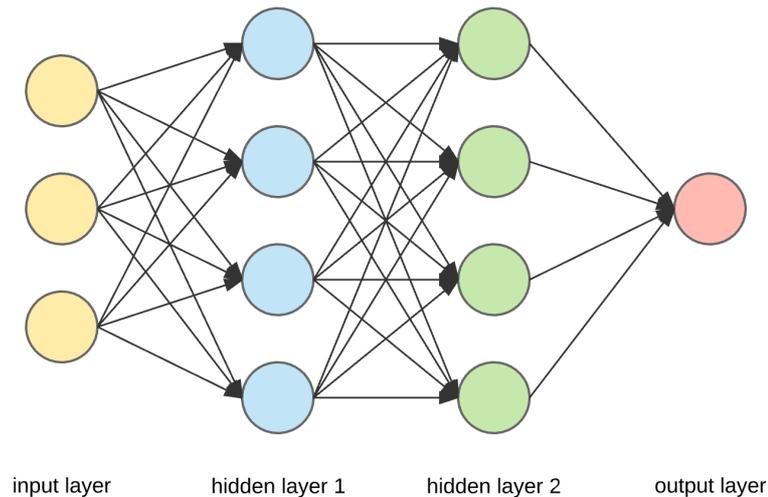
Aspek pembelajaran skalabilitas dan fitur hirarkis dari jaringan saraf dalam menjadikannya pilihan yang layak untuk tugas-tugas seperti klasifikasi citra, deteksi objek, dan pengenalan suara. Terdapat 3 tipe dari *deep learning*, yaitu: Artificial Neural Networks (ANN), Convolution Neural Networks (CNN), Recurrent Neural Networks (RNN).

### **2.3. Artificial Neural Network (ANN)**

Jaringan Syaraf Tiruan, pada dasarnya diilhami oleh struktur saraf otak kita, terdiri dari jaringan simpul yang terhubung yang dikenal sebagai neuron. Neuron adalah unit pemrosesan sentral dari jaringan saraf, dan mereka terhubung seperti neuron dan sinapsis dalam otak biologis. Koneksi saraf tertimbang membantu dalam mentransmisikan sinyal (fitur) dari satu neuron ke yang lain. Jaringan saraf tiruan belajar dengan contoh daripada memprogramnya dengan aturan spesifik tugas. Misalnya, dalam pengenalan pola visual, mereka mungkin belajar dengan contoh pelatihan digit tulisan tangan dari 0 hingga 9 dan menggunakan pengetahuan ini untuk mengidentifikasi digit dalam sampel lain yang tidak terlihat (Nielsen, 2015).

Jaringan saraf tiruan (JST) sebagian besar terdiri dari tiga jenis lapisan, lapisan input, lapisan output, dan satu atau lebih lapisan tersembunyi. Pada Gambar 2.2 yang ditunjukkan di atas, ada dua lapisan tersembunyi. Lapisan tersembunyi pertama mempelajari fitur-fitur sederhana dan dasar dan meneruskannya ke lapisan tersembunyi kedua. Berdasarkan input dari lapisan sebelumnya, lapisan kedua

mempelajari fitur yang lebih kompleks dan pada tingkat yang lebih abstrak daripada yang pertama. Demikian pula, lapisan yang lebih jauh dalam jaringan dapat mempelajari fitur yang lebih rumit dan sebagainya. Dengan cara ini, jaringan multi-layered dapat digunakan secara efektif untuk menyelesaikan masalah canggih seperti klasifikasi citra, deteksi objek, dan lainnya.



**Gambar 2.4** Arsitektur ANN (Arden Dertat, 2019)

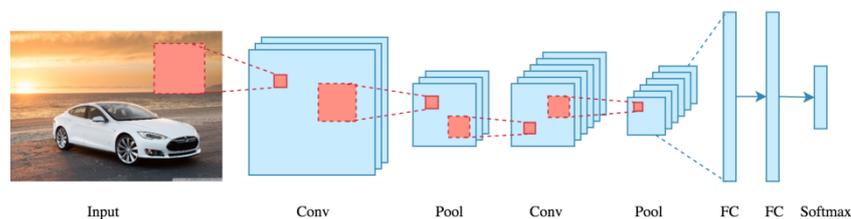
#### 2.4. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) adalah kelas arsitektur yang paling umum digunakan untuk *deep learning*. CNN bekerja dengan cara yang sama sebagai jaringan saraf tiruan kecuali ia memiliki serangkaian lapisan konvolusional pada awalnya. CNN banyak digunakan untuk tugas-tugas pengenalan visual seperti klasifikasi citra, deteksi objek, dan pengenalan suara. CNN telah menjadi mana-mana di kalangan peneliti dan komunitas visi komputer sejak AlexNet memamerkan kinerja luar biasa di ImageNet Challenge pada 2012 (Krizhevsky et al., 2012).

Pada Gambar 2.4 menunjukkan arsitektur dasar CNN. Komponen utama arsitektur CNN adalah sebagai berikut:

- Convolutional layers
- Pooling layers
- Fully connected layers
- Softmax function

Setiap citra input dilewatkan melalui beberapa lapisan konvolusional dan penyatuan, diikuti oleh beberapa lapisan yang sepenuhnya terhubung. Selain itu, fungsi softmax diterapkan di akhir untuk melakukan klasifikasi multiclass.



**Gambar 2.5** Contoh arsitektur CNN (Arden Dertat, 2019)

#### 2.4.1. Convolutional Layers

Lapisan konvolusi adalah lapisan pertama dan utama dari CNN. Lapisan konvolusi terdiri dari neuron yang bertindak sebagai *filter*. *Filter* digeser berurutan di atas seluruh citra untuk membuat peta fitur. Proses geser ini dikenal sebagai konvolusi (Athiwaratkun & Kang, 2015). Gambar 2.5 dan 2.6 menggambarkan proses konvolusi pada peta fitur input 6x6 menggunakan *filter* 3x3 (kernel). Setelah menggeser filter di seluruh input, hasilnya adalah peta fitur keluaran ukuran 4x4.

Setiap lapisan berfungsi sebagai filter yang berbeda dan mempelajari peta fitur yang berbeda dari gambar input atau peta fitur input. Misalnya, diberi gambar sebagai input, jika kita memiliki tiga lapisan konvolusi dalam satu blok. Output dari setiap lapisan adalah peta fitur untuk tiga karakteristik berbeda seperti warna, tepi, dan bentuk. Juga, output dari setiap blok konvolusi memiliki tiga

dimensi, tinggi, lebar, dan kedalaman. Tinggi dan lebar sesuai dengan tinggi dan lebar peta fitur, dan kedalaman adalah jumlah lapisan konvolusi.

3x1	2x0	5x0	1	8	6
1x0	4x1	6x1	4	7	3
8x1	9x1	1x1	5	3	7
2	4	5	2	9	2
3	1	6	1	5	8
9	3	7	3	2	5

(a) Convolution process

31	27	25	26
22	21	30	27
27	24	24	30
28	24	23	25

(b) Output feature map

**Gambar 2.6** 6x6 input feature map dan 3x3 filter

3	2	5	1	8	6
1	4	6	4	7	3
8	9	1	5	3	7
2	4	5	2	9	2
3	1	6	1	5	8
9	3	7	3	2	5

(a) Input feature map

1	0	0
0	1	1
1	1	1

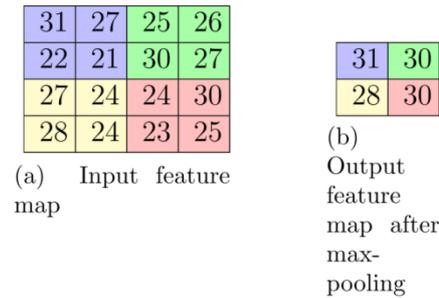
(b) Filter

**Gambar 2.7** Convolution process of 6x6 input dengan 3x3 filter

#### 2.4.2. Pooling Layers

Lapisan penyatuan biasanya mengikuti lapisan konvolusi. Lapisan pooling mengurangi dimensi masing-masing peta fitur (Nielsen, 2015). Lapisan mengambil setiap peta fitur independen, mengambil sampelnya dan membuat peta fitur terkompresi. Meskipun, output dari lapisan penyatuan masih tiga dimensi pada lapisan konvolusional. Kedalaman tetap sama, hanya tinggi dan lebar yang kental.

Ada berbagai jenis pooling seperti max-pooling, sum-pooling, dan average pooling. Max-pooling adalah yang paling populer di antara mereka. Max-pooling berfungsi dengan menggeser jendela di atas setiap peta fitur dan memilih nilai maksimal di jendela itu.



**Gambar 2.8** Maxpooling layer 2x2 dan stride 2

Gambar 2.7 di atas menunjukkan contoh max-pooling di mana ukuran jendela adalah 2x2 dan langkahnya adalah 2. Panjang langkahnya adalah dengan berapa banyak piksel yang kita pindahkan filter atau jendela di setiap langkah. Keuntungan utama dari pengumpulan adalah membantu mengurangi jumlah parameter yang pada dasarnya mengurangi waktu pelatihan.

### 2.4.3. Fully Connected Layers

Beberapa layer yang terhubung sepenuhnya ditambahkan setelah lapisan konvolusi dan pooling. Jaringan yang sepenuhnya terhubung adalah jaringan di mana semua neuron terhubung ke setiap neuron di lapisan tetangga. Seperti yang disebutkan di atas, output dari lapisan konvolusi + penyatuan adalah tiga dimensi, tetapi lapisan yang terhubung sepenuhnya mengkonsumsi vektor satu dimensi sebagai input. Jadi, kami meratakan output akhir dan kemudian mengirimkannya sebagai input dari lapisan yang terhubung sepenuhnya. Lapisan FC kemudian menggabungkan peta fitur ini untuk menyusun model

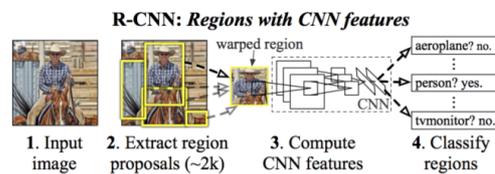
### 2.4.4. Softmax Function

Fungsi softmax adalah salah satu fungsi kerugian yang dapat ditambahkan sebagai lapisan terakhir CNN. Fungsi softmax berguna untuk deteksi multiclass; ini memberi kita kemungkinan untuk memprediksi satu kelas dari semua kelas lainnya. Jika kita menggunakan fungsi softmax untuk deteksi multiclass, output dari fungsi

ini adalah vektor probabilitas satu dimensi dari semua kelas. Jadi kelas dengan probabilitas tertinggi adalah kelas target.

## 2.5. Region based Convolutional Neural Network (R-CNN)

Memanfaatkan keberhasilan dari *Convolutional Neural Network* untuk merancang jaringan untuk deteksi objek yang disebut sebagai *Region-based Convolutional Neural Network* (R-CNN). Metode R-CNN tersebut merupakan perkembangan dari metode CNN. R-CNN menggabungkan antara region proposal dengan CNN (Girshick et al., 2014).



**Gambar 2.9** Contoh Arsitektur R-CNN (Girshick et al., 2014)

### 2.5.1. Region Proposals

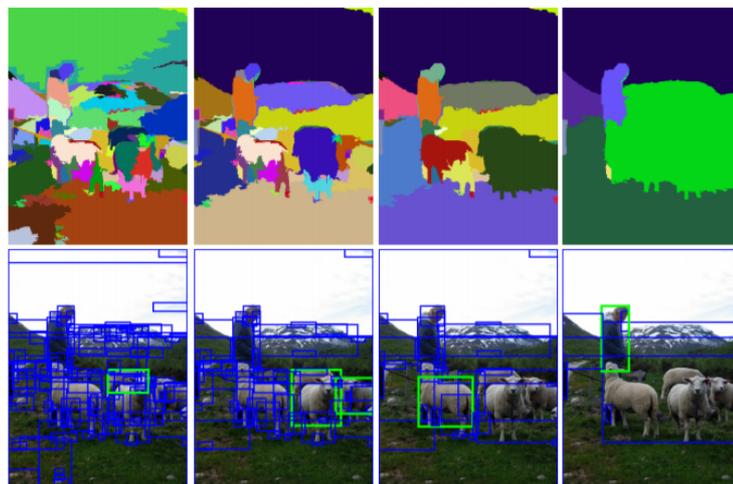
*Region proposal* adalah proses dimana beberapa wilayah dalam citra diusulkan sebagai daerah yang kemungkinan terdapat objek dalam wilayah tersebut. Perhatikan Gambar 2.8, terdapat 5 wilayah yang ditangkap sebagai yang memiliki objek di dalamnya, tetapi pada praktik yang sebenarnya akan ada lebih banyak wilayah yang tertangkap.



**Gambar 2.10** Region Proposal

Beberapa wilayah yang diberi boundingbox kemungkinan merupakan objek, jadi wilayah tersebut diusulkan sebagai objek yang diprediksi. Setelah wilayah tersebut diprediksi ada objek didalamnya.

Algoritma yang digunakan dalam region proposal adalah menggunakan *selective search* (Uijlings et al., 2013). Langkah-langkah algoritma *Selective Search* diilustrasikan pada Gambar 2.10.



**Gambar 2.11** Ilustrasi *Selective Search* (Uijlings et al., 2013)

Untuk menjelaskan metode *Selective Search* pada Gambar 2.10, berikut langkah-langkahnya:

- Melakukan segmentasi menggunakan metode yang diajukan oleh Felzenszwalb (P. F. Felzenszwalb & Huttenlocher, 2004).
- Selanjutnya, diterapkan algoritma *brute* untuk menggabungkan daerah dengan cara melihat *similarity*, seperti warna, ukuran, tekstur, dan komposisi.
- Selanjutnya didapat objek dari segmentasi tersebut beserta koordinatnya.

### 2.5.2. Feature Extraction

Pada *feature extraction*, hasil citra yang didapat pada region proposal diekstrak fitur menggunakan CNN. Untuk melatih jaringan dengan R-CNN diperlukan *dataset* beranotasi yang sangat besar, tetapi *dataset* yang tersedia pada saat itu sangat langka. Untuk mengatasi masalah kelangkaan *dataset*, *pre-trained* CNN digunakan bersamaan dengan dataset klasifikasi ImageNet 2012 yang hanya memiliki anotasi tingkat citra (tanpa data bounding box). Kemudian, jaringan disesuaikan untuk dua dataset yang berbeda, dataset PASCAL VOC 2012 dan dataset ImageNet 2013, dengan *bounding box*.

Hasil pendekatan yang dianut di atas cukup positif. Pada dataset PASCAL VOC 2012, R-CNN mencapai skor mAP (Mean Average Precision) 62,4%, yang merupakan 22,0% lebih banyak dari model terbaik kedua. Demikian pula, pada dataset ImageNet 2013, R-CNN mencapai skor mAP 31,4%, lebih tinggi 7% dari yang terbaik kedua. Oleh karena itu, pendekatan yang diadopsi oleh penulis diawasi pra-pelatihan pada dataset sekunder yang besar, diikuti oleh fine-tuning khusus-domain meningkatkan kinerja jauh (Girshick et al., 2014).

## 2.6. Penelitian Terkait

Secara umum, prosedur pendeteksian objek meliputi preprocessing citra, ekstraksi ciri, pelatihan classifier, dan regresi bounding box. Pada deteksi objek terdapat 2 metode deteksi yaitu: metode tradisional dan metode berbasis CNN.

Metode tradisional rata-rata menggunakan *descriptor* seperti *Scale Invariant Feature Transform* (SIFT) dan *Histogram of Oriented Gradient* (HOG). Sedangkan untuk *classifier* menggunakan *Support Vector Machine* (SVM). Pada metode tradisional, terdapat limitasi yang membuat kinerja deteksi menjadi buruk seperti efek bayangan, skenario yang kompleks, dan terhalang oleh objek lain (*occlusion*). Dengan lahirnya CNN pada tahun 2012, masalah yang terjadi pada metode tradisional tersebut dapat diselesaikan dan mencapai kesuksesan untuk mendeteksi objek serta deteksi kendaraan (Huang et al., 2017).

Penelitian yang dilakukan (Girshick et al., 2014) membahas tentang *Region-based Convolutional Neural Network* (R-CNN), yang telah menjadi referensi bagi banyak peneliti dalam bidang deteksi objek. R-CNN merupakan solusi dalam memecahkan kendala pada CNN yang diharuskan memiliki citra berukuran tetap. R-CNN menggabungkan *region proposal* dengan CNN. Jaringan konvolusi diterapkan pada *region proposal* untuk mengidentifikasi dan melokalisasi objek.

Penelitian yang dilakukan (Girshick, 2015) membahas tentang perbaikan dari metode R-CNN yang disebut dengan Fast R-CNN. Fast R-CNN mengadopsi *Region of Interest* (RoI) pooling layer dan *multi-task loss* untuk memperkirakan kelas objek dan memprediksi posisi *bounding box*.

Penelitian yang dilakukan (Ren et al., 2015) membahas tentang perbaikan dari metode sebelumnya, dengan memanfaatkan *Region Proposal Network* (RPN) untuk

dijadikan pengganti dari algoritma *selective search* yang masih lamban dalam menemukan *region proposal*. Perbaikan tersebut menjadikan Faster R-CNN sebagai referensi banyak peneliti lain. Namun, kecepatan deteksi pada Faster R-CNN Karena keunggulan ini, penulis melakukan penelitian untuk Faster R-CNN.

Dengan kehandalan yang dihasilkan oleh FasterR-CNN peneliti memilih Faster R-CNN untuk digunakan sebagai sistem deteksi pengguna jalan. Namun kecepatan Faster R-CNN masih belum secepat detektor *one-shot detector*. Berikut mengenai beberapa penelitian yang telah dilakukan untuk meningkatkan kecepatan maupun akurasi deteksi metode Faster R-CNN.

Penelitian yang dilakukan (Zhang et al., 2019) membahas perbaikan metode Faster R-CNN untuk menangani *occlusion* atau okulasi. Okulasi merupakan kondisi dimana objek pada citra tertutupi oleh objek lainnya, sehingga tidak mudah untuk dikenali karena sebagian objek saja yang tampak. Metode yang diusulkan yaitu dengan mengubah RPN original menjadi Part-Aware RPN dan menggunakan Part-Aware NMS sebagai pengganti NMS. *Base network* atau juga dikenal dengan *backbone* yang digunakan yaitu VGG16 dengan dataset KITTI. Hasil pengukuran akurasi pada penelitian tersebut sebesar 90,21% dan waktu deteksi 2,1 detik.

Penelitian yang dilakukan (Cao et al., 2019) membahas cara mengatasi masalah deteksi pada objek kecil dengan melakukan perbaikan pada Faster R-CNN. Penelitian tersebut mengusulkan menggunakan *loss function* yang diperbaiki berdasarkan *Intersection over Union* (IoU) untuk regresi *bounding box* dan menggunakan *bilinear interpolation* untuk meningkatkan proses *Region of Interest Pooling* (RoI Pooling) untuk menyelesaikan masalah deviasi posisi. Menggunakan VGG16 sebagai *base network* dan menggunakan SoftNMS sebagai pengurangan

duplikasi *region proposal*. Pada dataset TT100K metode yang diusulkan tersebut mendapatkan akurasi sebesar 87%.

Penelitian yang dilakukan (Mo & Yan, 2020) membahas bagaimana mendeteksi kendaraan dalam citra udara yang umumnya berukuran kecil, sehingga kinerja pendeteksian menjadi kurang baik. Oleh karena itu dilakukan perbaikan pada Faster R-CNN untuk mendeteksi citra udara. Perubahan *base network* dari VGG16 menjadi ResNet101 digunakan pada penelitian tersebut. Dataset yang digunakan adalah dataset VEDAI dengan hasil akurasi yang didapat sebesar 75,7%.

Penelitian yang dilakukan (Liu et al., 2020) membahas tentang pendeteksian pada perangkat *catenary* untuk memudahkan melakukan inspeksi. Faster R-CNN digunakan sebagai acuan sistem dan dilakukan perbaikan dengan mengubah *base network* menggunakan ResNet101 dan SoftNMS sebagai pengurangan duplikasi *region proposal*. Dengan menggunakan dataset *catenary* dihasilkan akurasi sebesar 80%.

Dari ringkasan singkat mengenai penelitian yang sudah dilakukan, penulis mengusulkan melakukan perbaikan pada metode Faster R-CNN dengan menggunakan arsitektur MobileNet dan algoritma SoftNMS sebagai pengganti NMS tradisional. MobileNet dipilih karena jaringan tersebut ringan sehingga efektif mengurangi biaya komputasi dan jumlah parameter konvolusi. Berikut tabel perbandingan antara penelitian terkait dengan metode yang diusulkan oleh penulis.

Tabel 2.1 Penelitian terkait perbaikan metode Faster R-CNN

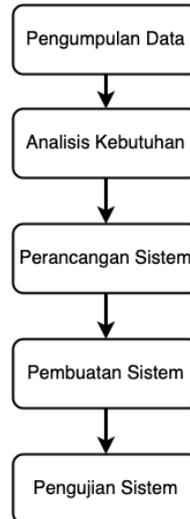
Penelitian	Base Network	Region Algoritm	NMS	Dataset	Waktu Deteksi(s)	Akurasi
(Zhang et al., 2019)	VGG16	Part-Aware RPN	Part-Aware NMS	KITTI	2,1	90,21%
(Cao et al., 2019)	VGG16	RPN	SoftNMS	TT100K	-	87%
(Mo & Yan, 2020)	ResNet101	RPN	-	VEDAI	-	75,7%
(Liu et al., 2020)	ResNet101	RPN	SoftNMS	Catenary	-	80%
(Penulis, 2021)	MobileNet	RPN	SoftNMS	PascalVOC + dataset pribadi	0,24	96,84%

## BAB III

### METODOLOGI PENELITIAN

#### 3.1. Prosedur Penelitian

Dalam melakukan penelitian, ada beberapa tahapan yang dilakukan oleh peneliti. Alur dari tahapan tersebut dapat dilihat pada Gambar 3.1.



**Gambar 3.1** Alur Penelitian

Berdasarkan gambar 3.1 dapat dijabarkan sebagai berikut: 1) Pada tahap pengumpulan data, terdapat 2 jenis data yang dikumpulkan yaitu data primer dan data sekunder. 2) Analisis kebutuhan sistem bertujuan untuk mengidentifikasi kebutuhan perangkat yang digunakan, meliputi *software* (perangkat lunak) dan *hardware* (perangkat keras). 3) Pada proses perancangan sistem ini, didefinisikan bagaimana proses dari sistem tersebut sehingga dapat berjalan dengan baik. 4) Pada tahap pembuatan sistem, sistem dibangun berdasarkan rancangan yang telah dibuat. 5) Pada tahap pengujian sistem, sistem diuji dengan beberapa tahap skenario pengujian.

### 3.2. Pengumpulan Data

Sebelum melakukan proses pengumpulan data, terlebih dahulu peneliti mendefinisikan pengguna jalan.



**Gambar 3.2** Contoh citra pengguna jalan

Untuk pengguna jalan yang akan diidentifikasi pada penelitian ini dibagi menjadi 3 kategori, yaitu: 1) Mobil, 2) Sepeda motor, 3) Manusia.

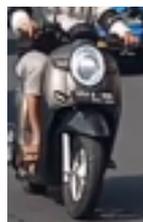
Berikut penjelasan mengenai ciri-ciri dari variabel yang akan dideteksi:

- a. Citra mobil, citra mobil memiliki ciri beroda 4 dan berpostur lebar. Berikut contoh dari citra mobil.



**Gambar 3.3** Citra mobil

- b. Citra sepeda motor, citra sepeda motor memiliki ciri beroda 2 dan berpostur ramping. Berikut contoh dari citra sepeda motor.



**Gambar 3.4** Citra sepeda motor

- c. Citra manusia, citra manusia memiliki ciri memiliki 2 kaki dan berpostur tinggi.



**Gambar 3.5** Citra manusia

Terdapat 2 jenis data yang akan dikumpulkan, yaitu: data primer dan data sekunder.

### **3.2.1. Data Primer**

Peneliti mendapatkan data primer dengan cara melakukan pengambilan video menggunakan kamera pada *smartphone*. Pengambilan video dilakukan dengan cara mengendarai sepeda motor dan menyusuri jalan di beberapa titik di wilayah kota Malang. Dari video yang didapat kemudian dijadikan frame-frame. Dari banyak video yang diambil menghasilkan sekitar 1000 citra frame. Dari 1000 citra tersebut akan dibagi menjadi 2 dengan rasio 50%:50%. Bagian pertama sejumlah 500 akan dilakukan labeling yang nantinya akan digabungkan dengan data sekunder untuk dijadikan input data training. Sementara bagian kedua sejumlah 500 tidak dilakukan proses labeling karena frame tersebut akan dijadikan sebagai input dari data testing.

Adapun langkah-langkah yang perlu dilakukan untuk mendapatkan data primer adalah sebagai berikut:

a. Pengambilan Video

Video diambil dengan menggunakan kamera smartphone dengan menyusuri jalan di beberapa titik di wilayah kota Malang, diantaranya: Merjosari, Tlogomas, Dinoyo, Sukun, dan Klojen.

b. Mengubah Video Menjadi Frame

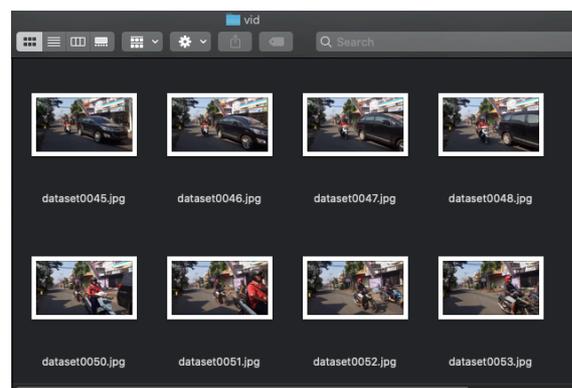
Ada beberapa aplikasi yang dapat digunakan dalam pengambilan frame dari video. Pada penelitian ini, peneliti menggunakan aplikasi ffmpeg. Aplikasi ffmpeg digunakan karena aplikasi tersebut ringan karena dijalankan pada terminal pada linux ataupun command prompt pada windows dan kemudahan dalam mengoprasikannya.



```
tegarap@MacBook-Pro: ~
Last login: Thu Dec 3 19:02:41 on ttys000
> ffmpeg -i VID_4.mp4 -vf fps=10 dataset%04d.jpg
```

**Gambar 3.6** Menjalankan aplikasi ffmpeg pada terminal

Maka akan dihasilkan frame sebagai berikut:



**Gambar 3.7** Hasil proses mengubah video ke frame

c. Labeling Frame

Data yang telah dikumpulkan sebelumnya akan dilakukan *labeling*. Proses *labeling* ini bertujuan untuk menentukan *ground truth* dari objek pada citra.

*Labeling* dapat dilakukan dengan menggunakan *software* labeling yaitu LabelImg. Pemberian label juga dimaksudkan untuk meningkatkan akurasi sistem yang dibuat. Contoh menggunakan aplikasi LabelImg terdapat pada Gambar 3.3.



**Gambar 3.8** Proses labeling menggunakan LabelImg

### 3.2.2. Data Sekunder

Sedangkan pada data sekunder, peneliti menggunakan dataset PascalVOC2007. Dataset tersebut didapat dengan cara mengunduh pada laman Pascal VOC (<http://host.robots.ox.ac.uk/pascal/VOC/voc2007/index.html>, 2020). Pada dataset PascalVOC2007 terdapat 20 kelas objek citra yang dapat digunakan. Tetapi pada penelitian ini, peneliti hanya menggunakan 3 kelas objek yang sesuai dengan variabel yang diamati pada penelitian, yaitu: kelas mobil, kelas motor, dan kelas manusia.

Dataset PascalVOC2007 berupa citra dengan format “.jpg” dan file anotasi dengan format “.xml”. Semua dataset sekunder tersebut nantinya digunakan sebagai inputan data training.

### **3.3. Analisis Kebutuhan Sistem**

Tujuan dari analisis kebutuhan sistem ini sebagai identifikasi kebutuhan apa saja yang diperlukan atau digunakan untuk membantu kelancaran penelitian yang meliputi perangkat lunak (*software*) dan perangkat keras (*hardware*). Berikut merupakan perangkat lunak dan perangkat keras yang digunakan.

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

Perangkat lunak yang digunakan untuk membangun sistem ini antara lain:

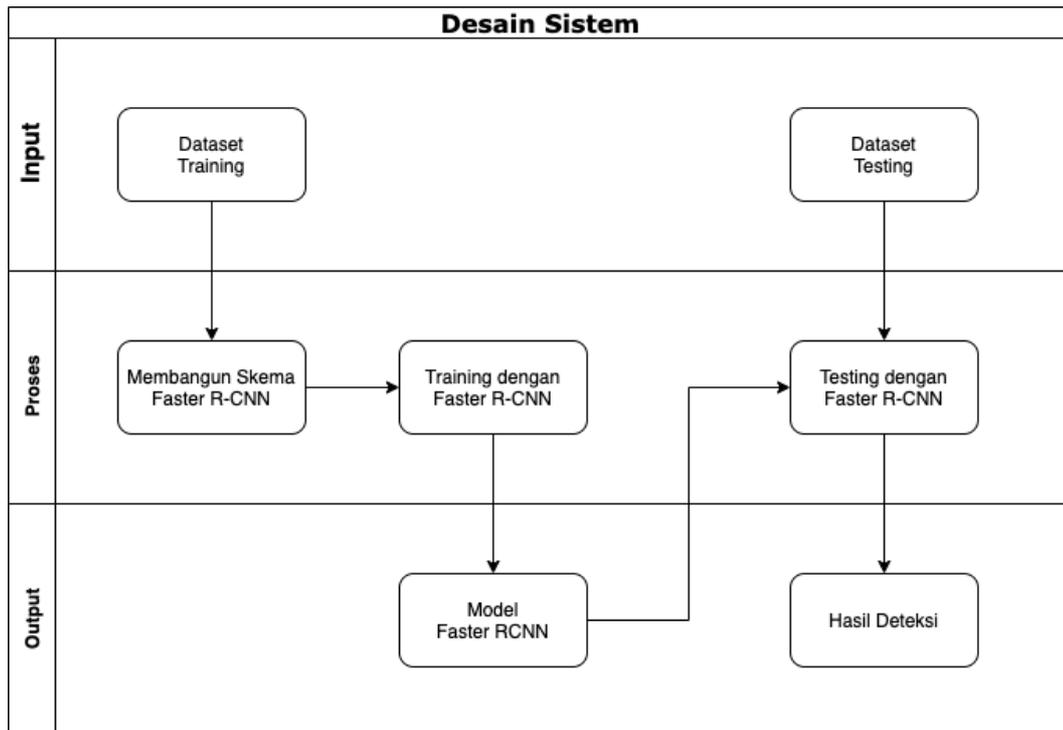
1. Sistem Operasi Windows 10
2. Notepad++
3. Google Colab (mirip jupyter notebook)
4. Online diagram maker (drawio)

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

*Hardware* (Perangkat keras) yang digunakan untuk membangun sistem ini yaitu laptop Thinkpad T460 dan *smartphone* Redmi 6A.

### **3.4. Desain Sistem**

Sebelum sebuah sistem dibangun, perlu adanya desain untuk sebagai gambaran seperti apa sistem yang akan dibangun nantinya pada penelitian ini.



**Gambar 3.9** Desain Sistem

### 3.4.1. Input

#### 3.4.1.1. Dataset Training

Dataset *training* yang digunakan adalah setengah dari data primer dan seluruhnya dari data sekunder. Kemudian data *training* ini dijadikan *input* pada pelatihan metode Faster R-CNN.

##### a. Simpan dataframe dari file xml

Pada dataset Pascal VOC, format labeling berupa file xml. File dengan format xml tersebut berisi informasi seperti berikut.

---

#### Anotasi 1 anotasi file xml

---

```

1. <annotation>
2.   <filename>vid20092.jpg</filename>
3.   <size>
4.     <width>1920</width>
5.     <height>1080</height>
6.     <depth>3</depth>

```

---

---

```
7.     </size>
8.     <object>
9.         <name>mobil</name>
10.        <bndbox>
11.            <xmin>819</xmin>
12.            <ymin>379</ymin>
13.            <xmax>1183</xmax>
14.            <ymin>674</ymin>
15.        </bndbox>
16.    </object>
17.    <object>
18.        <name>motor</name>
19.        <bndbox>
20.            <xmin>644</xmin>
21.            <ymin>495</ymin>
22.            <xmax>781</xmax>
23.            <ymin>623</ymin>
24.        </bndbox>
25.    </object>
26. </annotation>
```

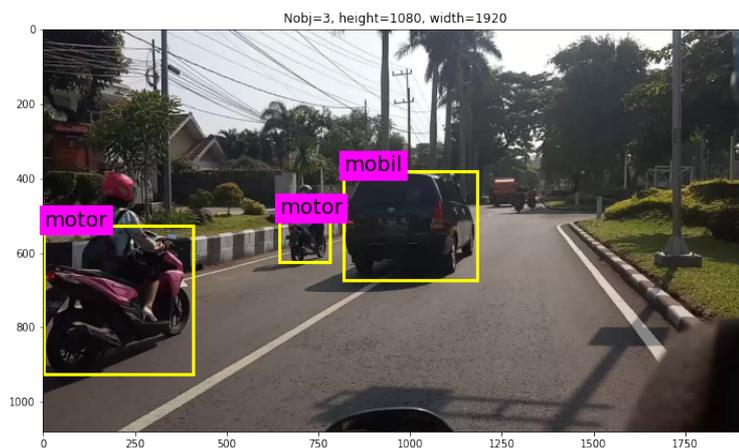
---

Pada Anotasi 1 dilihat bahwa file xml tersebut memuat berbagai informasi, diantaranya: filename, size, object, name, xmin, ymin, xmax, ymax. Dari informasi tersebut dapat dijadikan sebagai *groundtruth*.

Dari Anotasi tersebut terdapat variabel tag yang berisi informasi yang dijadikan sebagai *groundtruth*. Berikut penjelasan tentang tag tersebut.

1. **filename**, merupakan tag yang berisi informasi nama file.
2. **size**, merupakan tag yang berisi informasi ukuran dari citra.
3. **name**, merupakan tag yang berisi informasi nama objek anotasi.
4. **xmin**, merupakan tag yang berisi informasi koordinat pixel x minimum.
5. **ymin**, merupakan tag yang berisi informasi koordinat pixel y minimum.

6. **xmax**, merupakan tag yang berisi informasi koordinat pixel x maksimum.
7. **ymax**, merupakan tag yang berisi informasi koordinat pixel y maksimum.



**Gambar 3.10** *Bounding Box* dari file anotasi

Pada Gambar 3.10 divisualisasikan penerapan *boundingbox* dari file xml yang didapat. Untuk memudahkan dalam membaca dan mengolah informasi yang dibutuhkan, semua file xml dirubah ke dalam format csv.

#### **b. Simpan dataframe ke format csv**

File csv berbentuk tabel dengan nama kolom diambil dari informasi yang ada pada file xml. Berikut pseudocode pengambilan informasi dari file xml dan disimpan ke format csv.

---

#### **Algoritma 4** Pseudocode simpan dataframe ke csv

---

df\_anno:TUPLE

**FOR** fnm **IN** listdir of dir\_anno **IN** os **THEN**

    parse of (os.path.join(dir\_anno, fnm)) **IN** ET  $\leftarrow$  tree

    extract\_single\_xml\_file of tree  $\leftarrow$  row

    split of (".")[0] **IN** fnm  $\leftarrow$  row["fileID"]

    append of row **IN** df\_anno

DataFrame of df\_anno **IN** pd  $\leftarrow$  df\_anno

max of df\_anno["Nobj"] **IN** np  $\leftarrow$  maxNobj

"/content/drive/My Drive/Colab Notebooks"  $\leftarrow$  dir\_preprocessed

to\_csv(os.path.join(dir\_preprocessed,"df\_anno.csv"), index=False) **IN** df\_anno

---

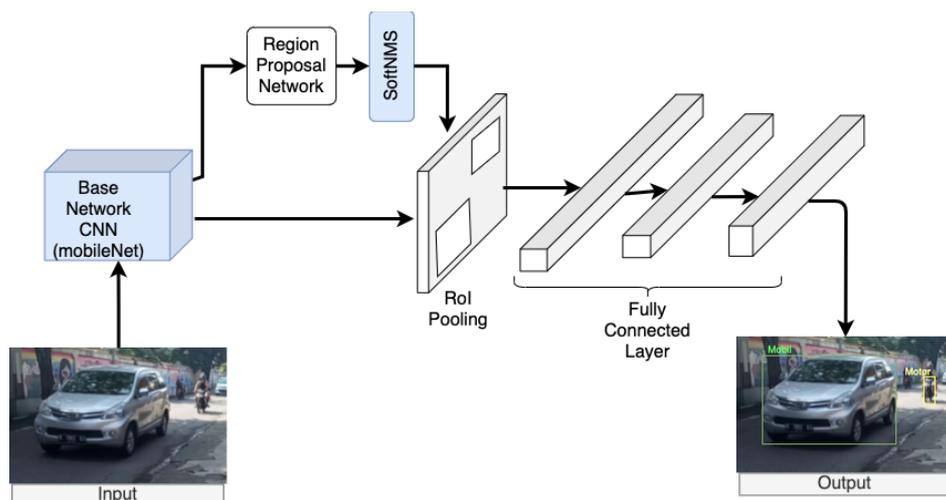
### 3.4.1.2. Data Testing

Data *testing* yang digunakan merupakan data yang telah dipersiapkan sebelumnya. Data *testing* yang digunakan adalah setengah dari data primer. Kemudian data *testing* ini dijadikan *input* pada pengujian model R-CNN

### 3.4.2. Proses

#### 3.4.2.1. Pemodelan Faster R-CNN

Pada poses ini peneliti membangun sistem deteksi menggunakan metode Faster R-CNN yang dilakukan modifikasi dari metode *original* dari Faster R-CNN. Untuk membedakan Faster R-CNN yang diusulkan dengan Faster R-CNN yang *original*, peneliti memberi warna biru pada bagian tersebut. Pada penelitian (Ren et al., 2015), Faster R-CNN menggunakan VGG16 sebagai arsitektur dasarnya dan menggunakan NMS tradisional, sedangkan pada penelitian ini penulis mengusulkan menggunakan arsitektur dasar MobileNet dengan SoftNMS.



**Gambar 3.11** Skema Faster R-CNN

## 1) Base Network

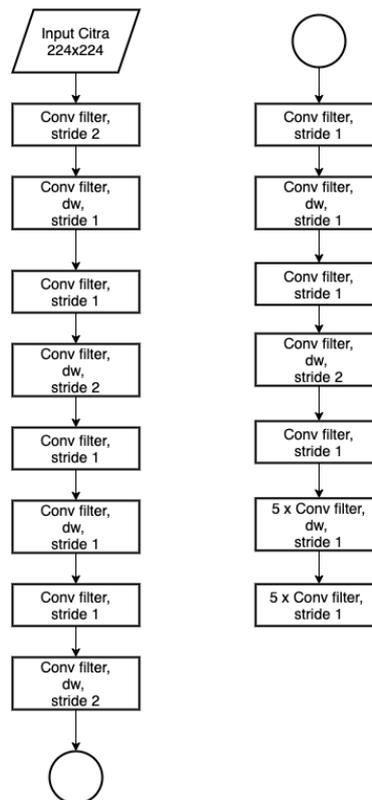
Faster R-CNN yang asli menggunakan VGG16 sebagai jaringan dasar CNN. Penelitian yang dilakukan oleh (Howard et al., 2017) membuktikan bahwa sekitar 80% dari waktu penerusan dihabiskan di jaringan dasar sehingga menggunakan jaringan dasar yang lebih cepat dapat sangat meningkatkan kecepatan seluruh kerangka kerja. Arsitektur MobileNet adalah jaringan efisien yang membagi konvolusi menjadi konvolusi  $3 \times 3$  kedalaman dan konvolusi titik  $1 \times 1$ , secara efektif mengurangi biaya komputasi dan jumlah parameter. Tabel 3.1 menunjukkan perbandingan MobileNet dan VGG-16 pada ImageNet. Seperti yang ditunjukkan, MobileNet hampir seakurat VGG-16 sementara 32 kali lebih kecil dan 27 kali lebih sedikit komputasi intensif.

Tabel 3.1 Perbandingan MobileNet dengan VGG16 (Howard et al., 2017)

Model	Akurasi ImageNet (%)	Parameter (juta)
MobileNet	70,6	4,2
VGG16	71,5	138

Dengan tujuan pendeteksian kendaraan secara real-time dalam adegan lalu lintas, arsitektur MobileNet digunakan sebagai jaringan dasar dalam penelitian ini. MobileNet memperkenalkan dua parameter yang dapat digunakan untuk menyesuaikan agar sesuai dengan trade-off sumber daya/akurasi, termasuk *width multiplier* dan *resolution multiplier*. *Width multiplier* memungkinkan kita untuk menipiskan jaringan, sedangkan *resolution multiplier* mengubah dimensi input gambar, sehingga mengurangi representasi internal di setiap

lapisan. Dalam studi ini, MobileNet diadopsi untuk membangun lapisan konvolusi dasar di Faster R-CNN daripada VGG-16 dalam kerangka asli untuk deteksi kendaraan cepat. Karena makalah ini hanya menggunakan lapisan konvolusi dalam arsitektur MobileNet, ukuran gambar input tidak harus diperbaiki. Misalkan ukuran gambar input adalah  $224 \times 224 \times 3$ , arsitektur jaringan dasar didefinisikan, seperti yang ditunjukkan pada Gambar 3.12.



**Gambar 3.12** Arsitektur MobileNet

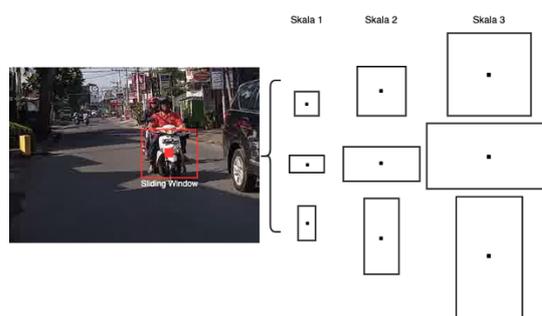
Pada tabel 2, “Conv” diartikan sebagai standar dari *convolution*, sedangkan “Conv dw” diartikan sebagai *depthwise separable convolution*. Pada tabel tersebut, *stride* disimbolkan sebagai “s”, “s1” diartikan sebagai stride dengan ukuran 1x1, sedangkan “s2” sebagai stride dengan ukuran 2x2.

Konvolusi *separable depthwise* terdiri dari dua lapisan: konvolusi *depthwise* dan konvolusi *pointwise*. Konvolusi Depthwise digunakan untuk menerapkan filter

tunggal per setiap saluran input, sedangkan konvolusi pointwise, konvolusi  $1 \times 1$  sederhana, digunakan untuk membuat kombinasi linier dari output lapisan depthwise. Arsitektur MobileNet menggunakan norma batch dan nonlinier ReLU untuk kedua lapisan. Pengurangan komputasi sebanding dengan jumlah saluran *feature map* keluaran dan kuadrat ukuran kernel.

## 2) Region Proposal Network

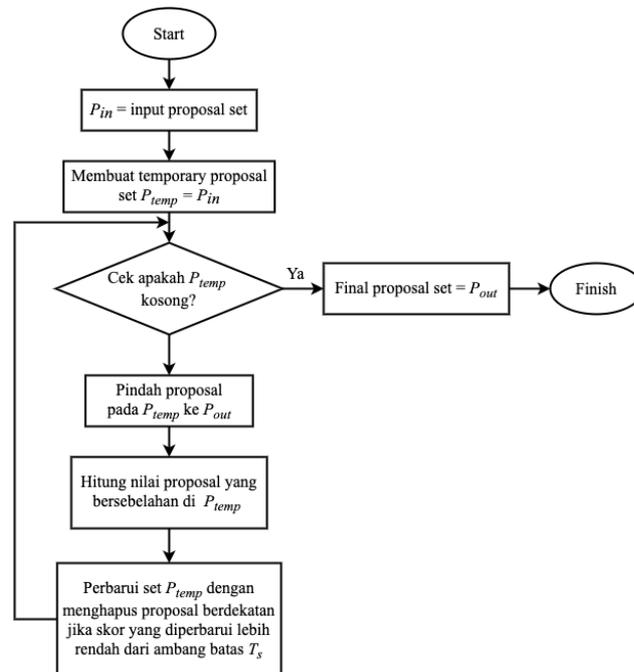
RPN pertama-tama menghasilkan satu set *anchor boxes* dari peta fitur konvolusi yang dihasilkan oleh *base network*. *Anchor* dipusatkan di *sliding window* dan dikaitkan dengan skala dan rasio aspek. Untuk pertukaran antara penarikan dan kecepatan pemrosesan, tiga skala *anchor box*, yaitu 128, 256, dan 512 dan tiga rasio *anchor box* 1:1, 1:2, dan 2:1 digunakan untuk setiap *anchor* dalam penelitian ini, menghasilkan 9 *anchor* pada setiap posisi geser.



**Gambar 3.13** *Anchor* pada *sliding window*

RPN kemudian mengambil semua *anchor box* dan mengeluarkan dua keluaran berbeda untuk masing-masing *anchor*. Yang pertama adalah skor objektivitas, yang berarti probabilitas bahwa *anchor* adalah objek. Output kedua adalah regresi *bounding box* untuk menyesuaikan anchor agar lebih sesuai dengan objek. Menggunakan koordinat proposal akhir dan skor objektivitasnya, kumpulan proposal yang baik untuk kendaraan dibuat. Karena *anchor* biasanya tumpang tindih, proposal akhirnya juga tumpang tindih pada objek yang sama. Algoritma

SoftNMS digunakan untuk mengatasi masalah duplikasi proposal. Di sebagian besar deteksi objek, termasuk Faster R-CNN, algoritma NMS digunakan untuk menghapus proposal duplikat. NMS tradisional menghapus proposal lain yang memiliki tumpang tindih lebih dari ambang batas yang telah ditentukan dengan proposal pemenang. Karena terdapat *occlusion* yaitu objek tertutup oleh objek di depannya, algoritma NMS tradisional dapat menghapus proposal positif secara tidak sengaja. Untuk mengatasi masalah NMS dengan kendaraan yang tertutupi, penelitian ini menggunakan algoritma softNMS. Dengan soft-NMS, proposal objek yang tertutupi objek lain akan dideteksi dengan baik.



**Gambar 3.14** Flowchart SoftNMS (Bodla et al., 2017)

Pada rumus  $P_{in} = \{P_1, P_2, P_3, \dots, P_n\}$  menunjukkan proposal awal set output dari lapisan *object proposal*, di mana proposal diurutkan berdasarkan skor objektifitasnya. Proposal  $P_i$  merupakan proposal lain yang memiliki overlap lebih dari nilai yang telah ditentukan *threshold*  $T$  dengan proposal  $P_i$  disebut proposal tetangga. Dalam penelitian ini, *threshold* proposal tetangga  $T$  diisi dengan nilai 0,5

dengan *cross-validation*. Kemudian  $S_i$  menunjukkan skor objektifitas  $P_i$ , yang merupakan nilai maksimum dalam vektor skor klasifikasi  $P_i$ . Untuk proposal yang ditetapkan, proposal dengan skor objektifitas tertinggi disebut proposal pemenang. Biarkan  $P_i$  menjadi proposal pemenang dan  $P_j$  menjadi proposal tetangga  $P_i$ . Skor objektifitas terupdate  $P_j$  (dilambangkan dengan  $S_j^u$ ) dihitung dengan rumus berikut:

$$S_j^u = S_i(1 - O_{P_i, P_j}) \quad (3.1)$$

Di mana  $O_{P_i, P_j}$  menunjukkan *intersection of union* (IoU) antara proposal  $P_i$  dan proposal  $P_j$  dan dihitung dengan rumus berikut:

$$O_{P_i, P_j} = \frac{\text{area}(P_i \cap P_j)}{\text{area}(P_i \cup P_j)} \quad (3.2)$$

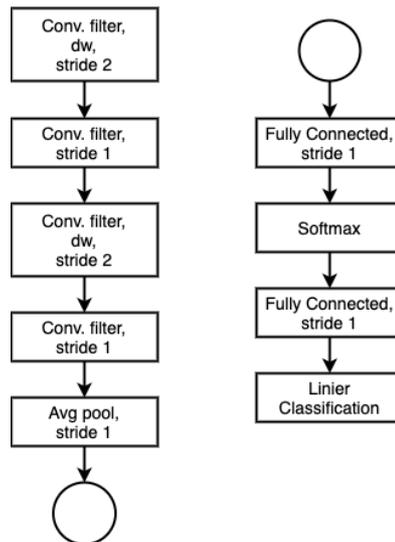
### 3) RoI Pooling

Di sebagian besar algoritma deteksi objek dua tahap, seperti Fast R-CNN, Faster R-CNN, dan sebagainya, lapisan RoI pooling digunakan untuk menyesuaikan ukuran proposal ke ukuran tetap. Lapisan pooling RoI menggunakan max pooling untuk mengonversi fitur di dalam wilayah minat yang valid menjadi peta fitur kecil dengan tingkat spasial tetap  $H \times W$ . RoI max pooling bekerja dengan membagi proposal RoI  $h \times w$  menjadi grid subwindows  $H \times W$  ukuran perkiraan  $(h/H) \times (w/W)$ , dan kemudian mengumpulkan nilai-nilai di setiap subwindow ke dalam sel grid keluaran yang sesuai. Jika proposal lebih kecil dari  $H \times W$ , itu akan diperbesar menjadi  $H \times W$  dengan menambahkan nilai yang direplikasi untuk mengisi ruang baru. Penggabungan RoI menghindari komputasi lapisan konvolusi berulang kali, sehingga dapat mempercepat waktu kereta dan pengujian secara

signifikan. Namun, menambahkan nilai yang direplikasi ke proposal kecil tidak tepat, terutama dengan kendaraan kecil, karena dapat merusak struktur asli kendaraan kecil. Selain itu, menambahkan nilai yang direplikasi untuk proposal kecil akan menyebabkan representasi yang tidak akurat dalam propagasi maju dan akumulasi kesalahan dalam propagasi mundur selama proses pelatihan. Dengan demikian, kinerja pendeteksian kendaraan kecil akan berkurang. Untuk menyesuaikan ukuran proposal ke ukuran tetap tanpa merusak struktur asli kendaraan kecil dan meningkatkan kinerja pendekatan yang diusulkan untuk mendeteksi kendaraan kecil.

#### **4) Classifier**

Pengklasifikasi adalah tahap akhir dalam kerangka yang diusulkan. Setelah mengekstrak fitur untuk setiap proposal melalui penyatuan RoI yang sadar konteks, fitur ini digunakan untuk klasifikasi. Pengklasifikasi memiliki dua tujuan yang berbeda: mengklasifikasikan proposal ke dalam kelas pengguna jalan dan latar belakang dan menyesuaikan *bounding box* untuk setiap kendaraan yang terdeteksi sesuai dengan kelas yang diprediksi.



**Gambar 3.15** Arsitektur klasifikasi

Dengan menggunakan struktur konvolusi yang dapat dipisahkan secara mendalam dalam arsitektur MobileNet untuk membangun pengklasifikasi. Pengklasifikasi memiliki dua lapisan yang terhubung penuh (FC), lapisan klasifikasi kotak dan lapisan regresi kotak. Lapisan FC pertama dimasukkan ke dalam lapisan softmax untuk menghitung probabilitas kepercayaan menjadi kendaraan dan latar belakang. Lapisan FC kedua dengan fungsi aktivasi linier meregresi *bounding box* kendaraan yang terdeteksi. Semua lapisan convolutional diikuti oleh lapisan normalisasi batch dan lapisan ReLU. Fungsi loss dan parameterisasi koordinat untuk regresi bounding box sama seperti pada metode Faster R-CNN *original*. Fungsi kerugian RPN dan fungsi kerugian pengklasifikasi memiliki bentuk yang sama tetapi dioptimalkan secara terpisah.

#### 3.4.2.2. Training dengan Faster R-CNN

Untuk mengidentifikasi objek pada citra, diperlukan proses training terlebih dahulu. Pada proses training ini algoritma dilatih dengan sejumlah data latih. Pada proses pelatihan ini data citra masukan akan melalui proses training

dengan menggunakan metode Convolutional Neural Network yang akan membentuk suatu model yang nanti akan diuji akurasinya. Tujuan dari pelatihan pada algoritma tersebut adalah untuk menemukan fitur atau ciri dari setiap citra kemudian dari fitur tersebut mengaktifkan neuron-neuron pada saat klasifikasi.

### 3.4.2.3. Testing dengan Faster R-CNN

Setelah melakukan proses *training* selanjutnya model yang telah didapatkan digunakan untuk proses *testing*. Data yang digunakan untuk testing memiliki proses yang sama dengan proses training yaitu melewati proses ekstraksi menggunakan region proposal. Pada proses testing, model yang digunakan adalah model yang sebelumnya didapatkan pada saat melakukan training. Proses testing ini dilakukan dengan cara mengujikan objek yang telah diekstraksi menggunakan region proposal. Hasil dari proses testing ini adalah sebuah *bounding box* pada objek yang di prediksi dan diukur akurasinya. Kemudian untuk mengukur akurasi dilakukan dengan menggunakan rumus berikut.

$$Akurasi = \frac{\sum \text{Prediksi benar}}{\sum \text{Data testing}} \times 100\% \quad (3.3)$$

Plotting dari hasil deteksi dapat dilihat pada Gambar 4.26.



**Gambar 3.16** Tampilan citra prediksi

Dari semua hasil prediksi dikumpulkan dalam bentuk array list dan dilakukan proses *confusion matrix*. *Confusion matrix* dari hasil data testing divisualkan pada Gambar 4.19. Terdapat 3 index yang mewakili kelas objek yang kita amati. Index 0 mewakili kelas mobil, index 1 mewakili kelas motor, dan index 2 mewakili kelas manusia.

Tabel 3.2 Confusion Matrix proses testing

		Predict Class		
		Mobil	Motor	Manusia
Actual Class	Mobil	224	5	35
	Motor	4	70	18
	Manusia	13	9	613

Dari sejumlah 991 data, menunjukkan hasil yang baik pada kelas mobil dan kelas manusia, sedangkan pada kelas motor menunjukkan hasil yang baik. Pada kelas mobil, dari 264 citra yang diuji, terdapat 224 citra yang diprediksi benar, 5 citra diprediksi sebagai motor, dan 35 citra diprediksi sebagai manusia. Kemudian pada kelas motor, dari 92 citra yang diuji, terdapat 70 citra yang diprediksi benar, 4 citra diprediksi sebagai mobil, dan 18 citra diprediksi sebagai manusia. Selanjutnya pada kelas manusia, dari 635 citra yang diuji, terdapat 613 citra yang diprediksi benar, 13 citra diprediksi sebagai mobil, dan 9 citra diprediksi sebagai motor.

Dapat dilihat pada Gambar 4.33, hasil akurasi sebesar 92%. Selain menggunakan library yang tersedia dapat juga dilakukan penghitungan akurasi secara manual sebagai berikut.

$$Akurasi = \frac{\sum \text{Prediksi benar kelas}}{\sum \text{Data kelas}} \times 100 \% \quad (3.4)$$

$$Akurasi = \frac{224 + 70 + 613}{991} \times 100 \%$$

$$Akurasi = 91,52 \%$$

### 3.4.3. Output

#### 3.4.3.1. Model dan Bobot

Model dan bobot yang dimaksud di sini merupakan hasil dari proses *training* data dengan berupa model CNN yang disimpan berupa file. Model ini nantinya akan digunakan untuk proses *testing*.

#### 3.4.3.2. Hasil Deteksi dan Akurasi

Hasil dari deteksi ini adalah berupa *bounding box* dan label yang sesuai dengan nama objek yang diklasifikasi serta akurasi dari objek yang dideteksi

### 3.5. Pengujian Sistem

Pada tahap pengujian sistem dilakukan 2 tahap pengujian, yaitu pengujian akurasi dan pengujian kecepatan. Pada pengujian akurasi, menggunakan nilai parameter *epoch* dan *learning rate* yang berbeda. Sebagai perbandingannya pengujian sistem ini menggunakan jumlah *epoch* 50, 100, 150, 200, dan 250. Sedangkan nilai dari *learning rate* yaitu: 0,1, 0,01, 0,001, 0,0001. Sedangkan pada pengujian kecepatan, menggunakan ukuran citra 1920x1080 pixel, 1152x648 pixel, dan 576x324 pixel.

## BAB IV

### UJI COBA DAN PEMBAHASAN

#### 4.1. Skenario Uji Coba

Pada sub bab ini peneliti menjelaskan langkah apa saja dalam melakukan uji coba deteksi pengguna jalan berbasis citra pada studi kasus daerah Kota Malang, diantaranya:

- a. Proses Pembuatan Dataset, pada tahap pembuatan dataset ini dilakukan pengumpulan data dan pelabelan. Kemudian setelah data berupa citra yang sudah diberi label diubah anotasinya yang awalnya file xml menjadi file csv.
- b. Proses pelatihan, pada tahap ini dilakukan pelatihan dari *dataset training* yang telah disiapkan sebelumnya untuk dilatih menggunakan metode Faster R-CNN yang diperbarui. Ada 20 model yang akan dilakukan training dengan parameter learning yang berbeda. Setelah dilakukan pelatihan didapatkan model Faster R-CNN yang nantinya digunakan untuk mengukur seberapa bagus metode tersebut.
- c. Proses pengujian akurasi, pada tahap ini model yang didapat dari hasil pelatihan dilakukan pengujian akurasi menggunakan *dataset testing*. Pengujian model yang didapat tersebut memiliki parameter *learning rate* dan *epoch* yang bervariasi. Berikut merupakan variasi dari model 1 sampai dengan model 20.

Tabel 4.1 Variasi learning rate dan epoch pengujian akurasi

Model	Epoch	Learning Rate	Model	Epoch	Learning Rate
1	50	0,1	11	150	0,001
2	50	0,01	12	150	0,0001
3	50	0,001	13	200	0,1
4	50	0,0001	14	200	0,01
5	100	0,1	15	200	0,001

6	100	0,01	16	200	0,0001
7	100	0,001	17	250	0,1
8	100	0,0001	18	250	0,01
9	150	0,1	19	250	0,001
10	150	0,001	20	250	0,0001

- d. Proses pengujian kecepatan, pada tahap ini model yang memiliki akurasi tertinggi diambil untuk diukur kecepatannya. Pengukuran tersebut dengan menggunakan 3 variasi ukuran citra untuk mengetahui pengaruh dari ukuran citra terhadap kecepatan deteksi. Citra tersebut didapat dari pengambilan *frame* pada *video*. Jumlah citra yang digunakan pada pengujian kecepatan ini sebanyak 100 citra pada setiap pengujian. Ukuran *frame* tersebut adalah 1920x1080 pixel. Dari ukuran *original* tersebut, dikecilkan ukurannya sebesar 60% menjadi 1152x648 pixel untuk dijadikan pengukuran ke-2. Pada pengukuran ke-3 ukuran *original* dikecilkan sebesar 30% menjadi 576x324 pixel.

Tabel 4.2 Tabel pengukuran kecepatan

Pengujian ke-	Ukuran Citra (pixel)
1	1920x1080
2	1152x648
3	576x324

## 4.2. Hasil Uji Coba

### 4.2.1. Akurasi

Uji coba pada sistem ini dilakukan dengan skema pengujian yang telah dijelaskan untuk mengetahui akurasi dari masing-masing model. Berikut hasil pelatihan model 1 sampai 20.

Tabel 4.3 Hasil uji coba model 1

No. data	Kelas Asli	Prediksi			Hasil
		Mobil	Motor	Manusia	
1	Mobil	-	-	✓	Tidak Sesuai
2	Mobil	-	-	✓	Tidak Sesuai
3	Mobil	-	-	✓	Tidak Sesuai
4	Mobil	-	-	✓	Tidak Sesuai
5	Mobil	-	-	✓	Tidak Sesuai
...	...	...			...
503	Manusia	-	-	✓	Sesuai
504	Manusia	-	-	✓	Sesuai
505	Manusia	-	-	✓	Sesuai
506	Manusia	-	-	✓	Sesuai
507	Manusia	-	-	✓	Sesuai
Jumlah		0	0	507	

Tabel 4.4 Confussion Matrix Model 1 (epoch 50, learning rate 0,1 )

		Prediksi		
		Mobil	Motor	Manusia
Aktual	Mobil	0	0	134
	Motor	0	0	54
	Manusia	0	0	319

Perhitungan akurasinya untuk model 1 adalah :

$$\begin{aligned}
 Akurasi &= \frac{0 + 0 + 319}{507} \times 100 \% \\
 &= 62,92\%
 \end{aligned}
 \tag{4.1}$$

Tabel 4.5 Hasil uji coba model 2

No. data	Kelas Asli	Prediksi			Hasil
		Mobil	Motor	Manusia	
1	Mobil	-	-	✓	Tidak Sesuai
2	Mobil	-	-	✓	Tidak Sesuai
3	Mobil	-	-	✓	Tidak Sesuai
4	Mobil	-	-	✓	Tidak Sesuai
5	Mobil	-	-	✓	Tidak Sesuai
...	...	...			...

503	Manusia	-	-	✓	Sesuai
504	Manusia	-	-	✓	Sesuai
505	Manusia	-	-	✓	Sesuai
506	Manusia	-	-	✓	Sesuai
507	Manusia	-	-	✓	Sesuai
Jumlah		0	0	507	

Tabel 4.6 Confussion Matrix Model 2 (epoch 50, learning rate 0,01 )

		Prediksi		
		Mobil	Motor	Manusia
Aktual	Mobil	0	0	134
	Motor	0	0	54
	Manusia	0	0	319

Perhitungan akurasinya untuk model 2 adalah :

$$\begin{aligned}
 Akurasi &= \frac{0 + 0 + 319}{507} \times 100 \% \\
 &= 62,92\%
 \end{aligned}
 \tag{4.2}$$

Tabel 4.7 Hasil uji coba model 3

No. data	Kelas Asli	Prediksi			Hasil
		Mobil	Motor	Manusia	
1	Mobil	✓	-	-	Sesuai
2	Mobil	✓	-	-	Sesuai
3	Mobil	✓	-	-	Sesuai
4	Mobil	✓	-	-	Sesuai
5	Mobil	✓	-	-	Sesuai
...	...	...			...
503	Manusia	✓	-	-	Tidak Sesuai
504	Manusia	✓	-	-	Tidak Sesuai
505	Manusia	-	-	✓	Sesuai
506	Manusia	-	-	✓	Sesuai
507	Manusia	-	-	✓	Sesuai
Jumlah		132	57	318	

Tabel 4.8 Confussion Matrix Model 3 (epoch 50, learning rate 0,001)

		Prediksi		
		Mobil	Motor	Manusia
Aktual	Mobil	126	2	6
	Motor	1	48	5
	Manusia	5	7	307

Perhitungan akurasi untuk model 3 adalah :

$$\begin{aligned}
 Akurasi &= \frac{126 + 48 + 307}{507} \times 100 \% \\
 &= 94,87\%
 \end{aligned}
 \tag{4.3}$$

Tabel 4.9 Hasil uji coba model 4

No. data	Kelas Asli	Prediksi			Hasil
		Mobil	Motor	Manusia	
1	Mobil	✓	-	-	Sesuai
2	Mobil	✓	-	-	Sesuai
3	Mobil	✓	-	-	Sesuai
4	Mobil	✓	-	-	Sesuai
5	Mobil	✓	-	-	Sesuai
...	...	...			...
503	Manusia	-	-	✓	Sesuai
504	Manusia	-	-	✓	Sesuai
505	Manusia	-	-	✓	Sesuai
506	Manusia	-	-	✓	Sesuai
507	Manusia	-	-	✓	Sesuai
Jumlah		144	62	309	

Tabel 4.10 Confussion Matrix Model 4 (epoch 50, learning rate 0,0001)

		Prediksi		
		Mobil	Motor	Manusia
Aktual	Mobil	117	4	13
	Motor	5	45	4
	Manusia	22	5	292

Perhitungan akurasi untuk model 4 adalah :

$$\begin{aligned}
 Akurasi &= \frac{117 + 45 + 292}{507} \times 100 \% \\
 &= 89,55\%
 \end{aligned}
 \tag{4.4}$$

Tabel 4.11 Hasil uji coba model 5

No. data	Kelas Asli	Prediksi			Hasil
		Mobil	Motor	Manusia	
1	Mobil	-	-	✓	Tidak Sesuai
2	Mobil	-	-	✓	Tidak Sesuai
3	Mobil	-	-	✓	Tidak Sesuai
4	Mobil	-	-	✓	Tidak Sesuai
5	Mobil	-	-	✓	Tidak Sesuai
...	...	...			...
503	Manusia	-	-	✓	Sesuai
504	Manusia	-	-	✓	Sesuai
505	Manusia	-	-	✓	Sesuai
506	Manusia	-	-	✓	Sesuai
507	Manusia	-	-	✓	Sesuai
Jumlah		0	0	507	

Tabel 4.12 Confussion Matrix Model 5 (epoch 100, learning rate 0,1)

		Prediksi		
		Mobil	Motor	Manusia
Aktual	Mobil	0	0	139
	Motor	0	0	54
	Manusia	0	0	319

Perhitungan akurasi untuk model 5 adalah :

$$\begin{aligned}
 Akurasi &= \frac{0 + 0 + 319}{507} \times 100 \% \\
 &= 62,92\%
 \end{aligned}
 \tag{4.5}$$

Tabel 4.13 Hasil uji coba model 6

No. data	Kelas Asli	Prediksi			Hasil
		Mobil	Motor	Manusia	
1	Mobil	-	-	✓	Tidak Sesuai
2	Mobil	-	-	✓	Tidak Sesuai
3	Mobil	-	-	✓	Tidak Sesuai
4	Mobil	-	-	✓	Tidak Sesuai
5	Mobil	-	-	✓	Tidak Sesuai
...	...	...			...
503	Manusia	-	-	✓	Sesuai
504	Manusia	-	-	✓	Sesuai
505	Manusia	-	-	✓	Sesuai
506	Manusia	-	-	✓	Sesuai
507	Manusia	-	-	✓	Sesuai
Jumlah		0	0	507	

Tabel 4.14 Confussion Matrix Model 6 (epoch 100, learning rate 0,01)

		Prediksi		
		Mobil	Motor	Manusia
Aktual	Mobil	0	0	139
	Motor	0	0	54
	Manusia	0	0	319

Perhitungan akurasi pada model 6 adalah :

$$\begin{aligned}
 \text{Akurasi} &= \frac{0 + 0 + 319}{507} \times 100 \% \\
 &= 62,92\% \qquad (4.6)
 \end{aligned}$$

Tabel 4.15 Hasil uji coba model 7

No. data	Kelas Asli	Prediksi			Hasil
		Mobil	Motor	Manusia	
1	Mobil	✓	-	-	Sesuai
2	Mobil	✓	-	-	Sesuai
3	Mobil	✓	-	-	Sesuai
4	Mobil	✓	-	-	Sesuai

5	Mobil	-	✓	-	Tidak Sesuai
...	...		...		...
503	Manusia	-	-	✓	Sesuai
504	Manusia	-	-	✓	Sesuai
505	Manusia	-	-	✓	Sesuai
506	Manusia	-	-	✓	Sesuai
507	Manusia	-	-	✓	Sesuai
Jumlah		131	53	323	

Tabel 4.16 Confussion Matrix Model 7 (epoch 100, learning rate 0,001)

		Prediksi		
		Mobil	Motor	Manusia
Aktual	Mobil	125	2	7
	Motor	1	49	4
	Manusia	5	2	312

Perhitungan akurasi pada model 7 adalah :

$$\begin{aligned}
 \text{Akurasi} &= \frac{125 + 49 + 312}{507} \times 100 \% \\
 &= 95,86 \% \qquad (4.7)
 \end{aligned}$$

Tabel 4.17 Hasil uji coba model 8

No. data	Kelas Asli	Prediksi			Hasil
		Mobil	Motor	Manusia	
1	Mobil	✓	-	-	Sesuai
2	Mobil	✓	-	-	Sesuai
3	Mobil	✓	-	-	Sesuai
4	Mobil	-	-	✓	Tidak Sesuai
5	Mobil	-	-	✓	Tidak Sesuai
...	...		...		...
503	Manusia	✓	-	-	Tidak Sesuai
504	Manusia	-	-	✓	Sesuai
505	Manusia	-	-	✓	Sesuai
506	Manusia	-	-	✓	Sesuai
507	Manusia	-	-	✓	Sesuai
Jumlah		137	45	325	

Tabel 4.16 *Confussion Matrix* Model 8 (epoch 100, learning rate 0,0001)

		Prediksi		
		Mobil	Motor	Manusia
Aktual	Mobil	129	0	5
	Motor	1	45	8
	Manusia	7	0	312

Perhitungan akurasi pada model 8 adalah :

$$\begin{aligned}
 \text{Akurasi} &= \frac{129 + 45 + 312}{507} \times 100 \% \\
 &= 95,86 \%
 \end{aligned}
 \tag{4.8}$$

Tabel 4.18 Hasil uji coba model 9

No. data	Kelas Asli	Prediksi			Hasil
		Mobil	Motor	Manusia	
1	Mobil	-	-	✓	Tidak Sesuai
2	Mobil	-	-	✓	Tidak Sesuai
3	Mobil	-	-	✓	Tidak Sesuai
4	Mobil	-	-	✓	Tidak Sesuai
5	Mobil	-	-	✓	Tidak Sesuai
...	...	...			...
503	Manusia	-	-	✓	Sesuai
504	Manusia	-	-	✓	Sesuai
505	Manusia	-	-	✓	Sesuai
506	Manusia	-	-	✓	Sesuai
507	Manusia	-	-	✓	Sesuai
Jumlah		0	0	507	

Tabel 4.19 *Confussion Matrix* Model 9 (epoch 150, learning rate 0,1)

		Prediksi		
		Mobil	Motor	Manusia
Aktual	Mobil	0	0	139
	Motor	0	0	54
	Manusia	0	0	319

Perhitungan akurasi model 9 adalah :

$$\begin{aligned}
 Akurasi &= \frac{0 + 0 + 319}{507} \times 100 \% \\
 &= 62,92\%
 \end{aligned}
 \tag{4.9}$$

Tabel 4.20 Hasil uji coba model 10

No. data	Kelas Asli	Prediksi			Hasil
		Mobil	Motor	Manusia	
1	Mobil	-	-	✓	Tidak Sesuai
2	Mobil	-	-	✓	Tidak Sesuai
3	Mobil	-	-	✓	Tidak Sesuai
4	Mobil	-	-	✓	Tidak Sesuai
5	Mobil	-	-	✓	Tidak Sesuai
...	...	...			...
503	Manusia	-	-	✓	Sesuai
504	Manusia	-	-	✓	Sesuai
505	Manusia	-	-	✓	Sesuai
506	Manusia	-	-	✓	Sesuai
507	Manusia	-	-	✓	Sesuai
Jumlah		0	0	507	

Tabel 4.21 Confussion Matrix Model 10 (epoch 150, learning rate 0,01)

		Prediksi		
		Mobil	Motor	Manusia
Aktual	Mobil	0	0	139
	Motor	0	0	54
	Manusia	0	0	319

Perhitungan akurasi model 10 adalah :

$$\begin{aligned}
 Akurasi &= \frac{0 + 0 + 319}{507} \times 100 \% \\
 &= 62,92\%
 \end{aligned}
 \tag{4.10}$$

Tabel 4.22 Hasil uji coba model 11

No. data	Kelas Asli	Prediksi			Hasil
		Mobil	Motor	Manusia	
1	Mobil	✓	-	-	Sesuai
2	Mobil	✓	-	-	Sesuai
3	Mobil	✓	-	-	Sesuai
4	Mobil	✓	-	-	Sesuai
5	Mobil	✓	-	-	Sesuai
...	...	...			...
503	Manusia	-	-	✓	Sesuai
504	Manusia	-	-	✓	Sesuai
505	Manusia	-	-	✓	Sesuai
506	Manusia	-	-	✓	Sesuai
507	Manusia	-	-	✓	Sesuai
Jumlah		123	47	337	

Tabel 4.23 Confussion Matrix Model 11 (epoch 150, learning rate 0,001)

		Prediksi		
		Mobil	Motor	Manusia
Aktual	Mobil	118	1	15
	Motor	3	44	7
	Manusia	2	2	315

Perhitungan akurasi model 11 adalah :

$$\begin{aligned}
 Akurasi &= \frac{118 + 44 + 315}{507} \times 100 \% \\
 &= 94,08 \%
 \end{aligned}
 \tag{4.11}$$

Tabel 4.24 Hasil uji coba model 12

No. data	Kelas Asli	Prediksi			Hasil
		Mobil	Motor	Manusia	
1	Mobil	✓	-	-	Sesuai
2	Mobil	✓	-	-	Sesuai
3	Mobil	✓	-	-	Sesuai

4	Mobil	✓	-	-	Sesuai
5	Mobil	✓	-	-	Sesuai
...	...	...			...
503	Manusia	-	-	✓	Sesuai
504	Manusia	-	-	✓	Sesuai
505	Manusia	-	-	✓	Sesuai
506	Manusia	-	-	✓	Sesuai
507	Manusia	-	-	✓	Sesuai
Jumlah		129	53	325	

Tabel 4.25 Confussion Matrix Model 12 (epoch 150, learning rate 0,0001)

		Prediksi		
		Mobil	Motor	Manusia
Aktual	Mobil	125	1	8
	Motor	1	47	6
	Manusia	3	5	311

Perhitungan akurasi model 12 adalah :

$$\begin{aligned}
 \text{Akurasi} &= \frac{125 + 47 + 311}{507} \times 100 \% \\
 &= 95,26 \%
 \end{aligned}
 \tag{4.12}$$

Tabel 4.26 Hasil uji coba model 13

No. data	Kelas Asli	Prediksi			Hasil
		Mobil	Motor	Manusia	
1	Mobil	-	-	✓	Tidak Sesuai
2	Mobil	-	-	✓	Tidak Sesuai
3	Mobil	-	-	✓	Tidak Sesuai
4	Mobil	-	-	✓	Tidak Sesuai
5	Mobil	-	-	✓	Tidak Sesuai
...	...	...			...
503	Manusia	-	-	✓	Sesuai
504	Manusia	-	-	✓	Sesuai
505	Manusia	-	-	✓	Sesuai
506	Manusia	-	-	✓	Sesuai
507	Manusia	-	-	✓	Sesuai

Jumlah	0	0	507	
--------	---	---	-----	--

Tabel 4.27 Confussion Matrix Model 13 (epoch 200, learning rate 0,1)

		Prediksi		
		Mobil	Motor	Manusia
Aktual	Mobil	0	0	139
	Motor	0	0	54
	Manusia	0	0	319

Perhitungan akurasi model 13 adalah :

$$\begin{aligned}
 Akurasi &= \frac{0 + 0 + 319}{507} \times 100 \% \\
 &= 62,92\%
 \end{aligned}
 \tag{4.13}$$

Tabel 4.28 Hasil uji coba model 14

No. data	Kelas Asli	Prediksi			Hasil
		Mobil	Motor	Manusia	
1	Mobil	-	-	✓	Tidak Sesuai
2	Mobil	-	-	✓	Tidak Sesuai
3	Mobil	-	-	✓	Tidak Sesuai
4	Mobil	-	-	✓	Tidak Sesuai
5	Mobil	-	-	✓	Tidak Sesuai
...	...	...			...
503	Manusia	-	-	✓	Sesuai
504	Manusia	-	-	✓	Sesuai
505	Manusia	-	-	✓	Sesuai
506	Manusia	-	-	✓	Sesuai
507	Manusia	-	-	✓	Sesuai
Jumlah		0	0	507	

Tabel 4.29 Confussion Matrix Model 14 (epoch 200, learning rate 0,01)

		Prediksi		
		Mobil	Motor	Manusia
Aktual	Mobil	0	0	139
	Motor	0	0	54

	Manusia	0	0	319
--	---------	---	---	-----

Perhitungan akurasi model 14 adalah :

$$\begin{aligned}
 Akurasi &= \frac{0 + 0 + 319}{507} \times 100 \% \\
 &= 62,92\%
 \end{aligned}
 \tag{4.14}$$

Tabel 4.30 Hasil uji coba model 15

No. data	Kelas Asli	Prediksi			Hasil
		Mobil	Motor	Manusia	
1	Mobil	✓	-	-	Sesuai
2	Mobil	✓	-	-	Sesuai
3	Mobil	✓	-	-	Sesuai
4	Mobil	✓	-	-	Sesuai
5	Mobil	✓	-	-	Sesuai
...	...	...			...
503	Manusia	-	-	✓	Sesuai
504	Manusia	-	-	✓	Sesuai
505	Manusia	-	-	✓	Sesuai
506	Manusia	-	-	✓	Sesuai
507	Manusia	-	-	✓	Sesuai
Jumlah		138	48	321	

Tabel 4.31 Confussion Matrix Model 15 (epoch 200, learning rate 0,001 )

		Prediksi		
		Mobil	Motor	Manusia
Aktual	Mobil	132	0	2
	Motor	2	46	6
	Manusia	4	2	313

Perhitungan akurasi model 15 adalah :

$$Akurasi = \frac{132 + 46 + 313}{507} \times 100 \% \tag{4.15}$$

= 96,84 %

Tabel 4.32 Hasil uji coba model 16

No. data	Kelas Asli	Prediksi			Hasil
		Mobil	Motor	Manusia	
1	Mobil	✓	-	-	Sesuai
2	Mobil	✓	-	-	Sesuai
3	Mobil	✓	-	-	Sesuai
4	Mobil	✓	-	-	Sesuai
5	Mobil	✓	-	-	Sesuai
...	...	...			...
503	Manusia	-	-	✓	Sesuai
504	Manusia	-	-	✓	Sesuai
505	Manusia	-	-	✓	Sesuai
506	Manusia	-	-	✓	Sesuai
507	Manusia	-	-	✓	Sesuai
Jumlah		125	49	333	

Tabel 4.33 Confussion Matrix Model 16 (epoch 200, learning rate 0,0001 )

		Prediksi		
		Mobil	Motor	Manusia
Aktual	Mobil	124	2	8
	Motor	0	46	8
	Manusia	1	1	317

Perhitungan akurasi model 16 adalah :

$$\begin{aligned}
 Akurasi &= \frac{124 + 46 + 317}{507} \times 100 \% \\
 &= 96,05 \%
 \end{aligned}
 \tag{4.16}$$

Tabel 4.34 Hasil uji coba model 17

No. data	Kelas Asli	Prediksi			Hasil
		Mobil	Motor	Manusia	
1	Mobil	-	-	✓	Tidak Sesuai

2	Mobil	-	-	✓	Tidak Sesuai
3	Mobil	-	-	✓	Tidak Sesuai
4	Mobil	-	-	✓	Tidak Sesuai
5	Mobil	-	-	✓	Tidak Sesuai
...	...	...			...
503	Manusia	-	-	✓	Sesuai
504	Manusia	-	-	✓	Sesuai
505	Manusia	-	-	✓	Sesuai
506	Manusia	-	-	✓	Sesuai
507	Manusia	-	-	✓	Sesuai
Jumlah		0	0	507	

Tabel 4.35 Confussion Matrix Model 17 (epoch 250, learning rate 0,1 )

		Prediksi		
		Mobil	Motor	Manusia
Aktual	Mobil	0	0	139
	Motor	0	0	54
	Manusia	0	0	319

Perhitungan akurasi model 17 adalah :

$$\begin{aligned}
 Akurasi &= \frac{0 + 0 + 319}{507} \times 100 \% \\
 &= 62,92\%
 \end{aligned}
 \tag{4.17}$$

Tabel 4.36 Hasil uji coba model 18

No. data	Kelas Asli	Prediksi			Hasil
		Mobil	Motor	Manusia	
1	Mobil	-	-	✓	Tidak Sesuai
2	Mobil	-	-	✓	Tidak Sesuai
3	Mobil	-	-	✓	Tidak Sesuai
4	Mobil	-	-	✓	Tidak Sesuai
5	Mobil	-	-	✓	Tidak Sesuai
...	...	...			...
503	Manusia	-	-	✓	Sesuai
504	Manusia	-	-	✓	Sesuai
505	Manusia	-	-	✓	Sesuai

506	Manusia	-	-	✓	Sesuai
507	Manusia	-	-	✓	Sesuai
Jumlah		0	0	507	

Tabel 4.37 Confussion Matrix Model 18 (epoch 250, learning rate 0,01 )

		Prediksi		
		Mobil	Motor	Manusia
Aktual	Mobil	0	0	139
	Motor	0	0	54
	Manusia	0	0	319

Perhitungan akurasi model 18 adalah :

$$\begin{aligned}
 Akurasi &= \frac{0 + 0 + 319}{507} \times 100 \% \\
 &= 62,92\%
 \end{aligned}
 \tag{4.18}$$

Tabel 4.38 Hasil uji coba model 19

No. data	Kelas Asli	Prediksi			Hasil
		Mobil	Motor	Manusia	
1	Mobil	✓	-	-	Sesuai
2	Mobil	✓	-	-	Sesuai
3	Mobil	✓	-	-	Sesuai
4	Mobil	✓	-	-	Sesuai
5	Mobil	✓	-	-	Sesuai
...	...	...			...
503	Manusia	-	-	✓	Sesuai
504	Manusia	-	-	✓	Sesuai
505	Manusia	-	-	✓	Sesuai
506	Manusia	-	-	✓	Sesuai
507	Manusia	-	-	✓	Sesuai
Jumlah		135	50	322	

Tabel 4.39 Confussion Matrix Model 19 (epoch 250, learning rate 0,001 )

		Prediksi		
		Mobil	Motor	Manusia

Aktual	Mobil	128	2	4
	Motor	2	43	9
	Manusia	5	5	309

Perhitungan akurasi model 19 adalah :

$$\begin{aligned}
 Akurasi &= \frac{128 + 43 + 309}{507} \times 100 \% \\
 &= 94,67 \%
 \end{aligned}
 \tag{4.19}$$

Tabel 4.40 Hasil uji coba model 20

No. data	Kelas Asli	Prediksi			Hasil
		Mobil	Motor	Manusia	
1	Mobil	✓	-	-	Sesuai
2	Mobil	✓	-	-	Sesuai
3	Mobil	✓	-	-	Sesuai
4	Mobil	✓	-	-	Sesuai
5	Mobil	✓	-	-	Sesuai
...	...	...			...
503	Manusia	-	-	✓	Sesuai
504	Manusia	-	-	✓	Sesuai
505	Manusia	-	-	✓	Sesuai
506	Manusia	-	-	✓	Sesuai
507	Manusia	-	-	✓	Sesuai
Jumlah		142	60	311	

Tabel 4.41 *Confussion Matrix* Model 20 (*epoch* 250, *learning rate* 0,0001 )

		Prediksi		
		Mobil	Motor	Manusia
Aktual	Mobil	129	0	5
	Motor	1	50	9
	Manusia	12	10	297

Perhitungan akurasi model 20 adalah:

$$\begin{aligned}
 \text{Akurasi} &= \frac{129 + 50 + 297}{507} \times 100 \% \\
 &= 93,89\%
 \end{aligned}
 \tag{4.20}$$

#### 4.2.2. Kecepatan

Uji coba kecepatan deteksi dilakukan dengan skema pengujian yang telah dijelaskan. Berikut hasil pengujian kecepatan yang didapat.

Tabel 4.42 Pengujian ke-1 (ukuran citra 1920x1080 pixel)

No. data	Nama file	Waktu deteksi (detik)
1	VID_20210413_105654.mp41098.jpg	0,23
2	VID_20210413_105654.mp41110.jpg	0,21
3	VID_20210413_105654.mp41122.jpg	0,23
4	VID_20210413_105654.mp41338.jpg	0,23
5	VID_20210413_110256.mp40010.jpg	0,22
...	...	...
96	VID_20210413_110256.mp40521.jpg	0,21
97	VID_20210413_110256.mp40713.jpg	0,22
98	VID_20210413_110256.mp40718.jpg	0,23
99	VID_20210413_110256.mp40749.jpg	0,23
100	VID_20210413_110256.mp40785.jpg	0,23
Kecepatan rata-rata		0,22

Tabel 4.43 Pengujian ke-2 (ukuran citra 1152x648 pixel)

No. data	Nama file	Waktu deteksi (detik)
1	VID_20210413_105654.mp41098.jpg	0,22
2	VID_20210413_105654.mp41110.jpg	0,2
3	VID_20210413_105654.mp41122.jpg	0,15
4	VID_20210413_105654.mp41338.jpg	0,21
5	VID_20210413_110256.mp40010.jpg	0,19
...	...	...
96	VID_20210413_110256.mp40521.jpg	0,15
97	VID_20210413_110256.mp40713.jpg	0,19
98	VID_20210413_110256.mp40718.jpg	0,18
99	VID_20210413_110256.mp40749.jpg	0,15
100	VID_20210413_110256.mp40785.jpg	0,14
Kecepatan rata-rata		0,18

Tabel 4.44 Pengujian ke-3 (ukuran citra 576x324 pixel)

No. data	Nama file	Waktu deteksi (detik)
1	VID_20210413_105654.mp41098.jpg	0,22
2	VID_20210413_105654.mp41110.jpg	0,2
3	VID_20210413_105654.mp41122.jpg	0,17
4	VID_20210413_105654.mp41338.jpg	0,15
5	VID_20210413_110256.mp40010.jpg	0,16
...	...	...
96	VID_20210413_110256.mp40521.jpg	0,15
97	VID_20210413_110256.mp40713.jpg	0,14
98	VID_20210413_110256.mp40718.jpg	0,15
99	VID_20210413_110256.mp40749.jpg	0,14
100	VID_20210413_110256.mp40785.jpg	0,18
Kecepatan rata-rata		0,16

### 4.3. Pembahasan

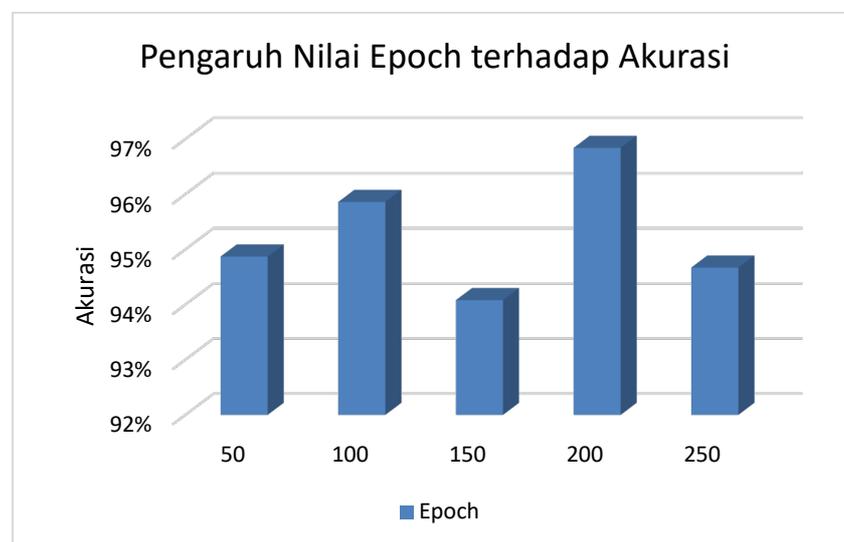
Berdasarkan hasil dari skenario pengujian akurasi dan pengujian kecepatan yang telah dilakukan pada bagian sebelumnya, diketahui bahwa nilai learning rate dan epoch berpengaruh terhadap tingkat akurasi deteksi, sedangkan pada kecepatan deteksi dipengaruhi oleh ukuran citra. Pada tingkat akurasi, peneliti juga memperhatikan penyebab dari tingkat akurasi. Dari hasil pengujian yang telah dilakukan, dihasilkan Tabel perbandingan nilai akurasi sebagai berikut.

Tabel 4.45 Perbandingan Nilai Akurasi

Model	Epoch	Learning Rate	Nilai Loss	Nilai Akurasi
1	50	0,1	0,86	62,92%
2	50	0,01	0,86	62,92%
3	50	0,001	0,01	94,87%
4	50	0,0001	0,23	89,55%
5	100	0,1	0,86	62,92%
6	100	0,01	0,86	62,92%
7	100	0,001	0,01	95,86%
8	100	0,0001	0,02	95,86%
9	150	0,1	0,86	62,92%
10	150	0,01	0,86	62,92%
11	150	0,001	0,06	94,08%
12	150	0,0001	0,01	95,26%

13	200	0,1	0,86	62,92%
14	200	0,01	0,86	62,92%
15	200	0,001	0,01	96,84%
16	200	0,0001	0,01	96,05%
17	250	0,1	0,86	62,92%
18	250	0,01	0,86	62,92%
19	250	0,001	0,01	94,67%
20	250	0,0001	0,01	93,89%

Dari Tabel 4.41 dilihat bahwa nilai epoch dan nilai learning rate dapat mempengaruhi nilai akurasi. Grafik 4.1 dibuat untuk memudahkan pengamatan pada pengaruh epoch terhadap akurasi.

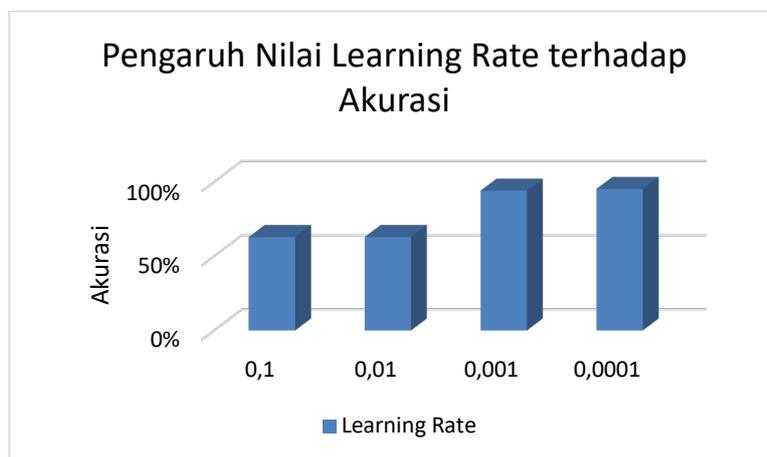


**Gambar 4.4** Grafik Pengaruh Nilai Epoch terhadap Akurasi

Pada grafik tersebut diukur akurasinya mulai dari epoch 50, 100, 150, 200, dan 250 dengan nilai learning rate yang sama yaitu 0,001. Pada grafik tersebut dilihat bahwa ada peningkatan akurasi pada epoch 50 dengan akurasi 94,87% dan epoch 100 dengan 95,86%. Kemudian penurunan terjadi pada epoch 150 dengan akurasi 94,08%. Selanjutnya naik lagi pada epoch 200 dengan akurasi 96,84%. Terjadi penurunan lagi pada epoch 250 dengan akurasi sebesar 94,67%.

Pemilihan jumlah epoch yang pas pada proses pelatihan menjadi penting karena epoch yang sedikit dapat dihasilkan akurasi yang sedikit atau bahkan menghasilkan *underfitting*, begitu juga epoch yang terlalu tinggi akan menghasilkan *overfitting*. *Underfitting* dan *overfitting* merupakan kejadian yang terjadi pada proses pelatihan semua *supervised learning*. Maksud dari *underfitting* adalah keadaan pada saat model pelatihan yang dibuat tidak mewakili keseluruhan data yang digunakan. Sehingga menghasilkan performa yang buruk. Sedangkan *overfitting* kebalikannya, yaitu keadaan dimana model pelatihan yang dibuat terlalu baik hasil yang didapatkan. Sehingga apabila dilakukan tes dengan menggunakan data yang berbeda dapat mengurangi akurasi (hasil yang dibuat tidak sesuai yang diharapkan). (Christian & Griffiths, 2017)

Sedangkan pengaruh nilai learning rate terhadap akurasi dapat dilihat pada Gambar 4.5. Pada grafik tersebut nilai epoch yang digunakan yaitu 150 dan learning rate yang diukur akurasinya yaitu 0,1, 0,01, 0,001, 0,0001.



**Gambar 4.5** Grafik Pengaruh Learning Rate terhadap Akurasi

Pada Gambar 4.5 dilihat bahwa grafik menunjukkan persamaan nilai akurasi pada learning rate 0,1 dan 0,01 yaitu sebesar 62,92%. Pada learning rate 0,001 mengalami kenaikan dengan akurasi menjadi 94,08%. Pada learning rate 0,0001

mengalami kenaikan pada training menjadi 95,26%. Dengan demikian memperkecil nilai *learning rate* dapat meningkatkan akurasi deteksi. Namun, dengan mengecilkan nilai *learning rate* dapat menyebabkan bertambahnya waktu yang diperlukan untuk melakukan pelatihan.

Pada pengukuran kecepatan deteksi, peneliti menggunakan model 15 yang memiliki akurasi tertinggi, yaitu sebesar 96,84%. Model tersebut diukur kecepatannya dengan 3 langkah pengujian.

Tabel 4.46 Perbandingan kecepatan deteksi masing-masing ukuran citra

Pengujian ke-	Ukuran citra (pixel)	Kecepatan deteksi (detik /citra)
1	1920x1080	0,22
2	1152x648	0,18
3	576x324	0,16

Pengujian ke-1, model diujikan dengan citra berukuran 1920x1080 pixel dengan hasil kecepatan rata-rata per citra yaitu 0,22 detik/citra. Pengujian ke-2, model diujikan dengan citra berukuran 1152x648 pixel dengan hasil kecepatan rata-rata per citra yaitu 0,18/citra. Pengujian ke-3, model diujikan dengan citra berukuran 576x324 pixel dengan hasil kecepatan rata-rata per citra yaitu 0,16 detik/citra.

Berdasarkan hasil skenario pengujian kecepatan tersebut, maka dapat diketahui ukuran citra berpengaruh terhadap kecepatan deteksi. Semakin kecil citra yang digunakan dalam deteksi semakin cepat pula kecepatannya. Namun, jika ukuran citra terlalu kecil juga akan menyulitkan proses deteksi dikarenakan citra yang dihasilkan menjadi kurang jelas dan terdapat banyak *noise*. Pada penelitian ini, ukuran citra terkecil yaitu 576x324 pixel cukup jelas jika diamati dan terdeteksi meskipun terdapat *noise*.

Kecepatan deteksi dan akurasi yang baik pada penelitian ini dapat dijadikan referensi untuk membangun *Intelligent Transportation System* (ITS) atau sistem lalu lintas cerdas yang handal guna terciptanya lalu lintas yang aman dan pengendara yang taat pada aturan lalu lintas. Ketaatan pada aturan lalu lintas ini selaras dengan firman Allah dalam Al-Qur'an Surah An-Nisa' ayat 59 yang berbunyi:

يَا أَيُّهَا الَّذِينَ آمَنُوا أَطِيعُوا اللَّهَ وَأَطِيعُوا الرَّسُولَ وَأُولَى الْأَمْرِ مِنْكُمْ

Artinya : “Hai orang-orang yang beriman, ta’atilah Allah dan ta’atilah Rasul (Nya), dan ulil amri di antara kamu.” (Q.S. An-Nisa’ : 59)

Ibnu Katsir dalam tafsirnya menyebutkan perkataan Ibnu Abbas. Bahwa asbabun nuzul dari surah tersebut berkenaan dengan dutusnya Abdullah bin Hudzafah bin Qais oleh Rasulullah SAW untuk memimpin suatu pasukan khusus. Abdullah memerintahkan pasukannya mengumpulkan kayu bakar dan membakarnya. Kemudian ia menyuruh pasukannya untuk memasuki api tersebut. Lalu salah seorang pasukan menjawab, “Sesungguhnya jalan keluar dari api ini hanya Rasulullah. Jangan tergesa-gesa sebelum menemui Rasulullah. Jika Rasulullah memerintahkan kepada kalian untuk memasuki api itu, maka masukilah.” Kemudian mereka menghadap Rasulullah untuk menceritakan hal tersebut. Rasulullah melarang memasuki api itu dan menegaskan bahwa ketaatan hanya dalam kebaikan.

Salah satu poin dalam surah tersebut yaitu, ketaatan pada *ulil amri*. Menurut Ibnu Abbas, Mujahid, Atha’, Hasan Al Basri dan Abul Aliyah, *ulil amri* (أولي الأمر) berarti para ulama. Menurut Syaikh Wahbah Az Zuhaili dalam Tafsir Al Munir, *ulil*

*amri* adalah pemimpin dan para ulama. Sedangkan menurut Ibnu Katsir, *ulil amri* itu bersifat umum baik pemerintah maupun ulama.

Pada redaksi perintah pada surah tersebut terdapat hal yang menarik, yaitu kata *athii'uu* (أَطِيعُوا) sebagaimana perintah taat pada Rasulullah tidak terdapat pada *ulil amri* yang dimaknai ketaatan kepada *ulil amri* hanya ketika perintahnya tidak bertentangan dengan ketaatan kepada Allah dan Rasul-Nya.

Ketaatan kepada *ulil amri* harus dibingkai dengan ketaatan kepada Allah dan Rasul-Nya. Tidak boleh taat dalam perkara buruk atau maksiat sesuai dengan sabda Nabi Muhammad SAW:

لَا طَاعَةَ فِي مَعْصِيَةِ اللَّهِ إِنَّمَا الطَّاعَةُ فِي الْمَعْرُوفِ

*“Tidak ada ketaatan dalam bermaksiat kepada Allah. Sesungguhnya ketaatan itu hanyalah dalam masalah kebaikan”* (HR. Muslim)

Selain hadist di atas, Rasulullah juga bersabda :

السَّمْعُ وَالطَّاعَةُ حَقٌّ ، مَا لَمْ يُؤْمَرْ بِالْمَعْصِيَةِ ، فَإِذَا أُمِرَ بِمَعْصِيَةٍ فَلَا سَمْعَ وَلَا طَاعَةَ

*“Mendengar dan taat (bagi muslim) itu haq, sejauh ia tidak diperintah untuk bermaksiat. Jika diperintah untuk bermaksiat, maka tidak ada kewajiban mendengar dan taat.”* (HR. Bukhari)

Pada Haji Wada', Rasulullah menyampaikan pada khutbahnya sebagai berikut:

وَلَوْ اسْتُعْمِلَ عَلَيْكُمْ عَبْدٌ يَقُودُكُمْ بِكِتَابِ اللَّهِ فَاسْمَعُوا لَهُ وَأَطِيعُوا

*“Seandainya seorang budak memimpin kalian dengan memakai pedoman Kitabullah, maka tunduk dan patuhlah kalian kepadanya.”* (HR. Muslim)

Dari tafsir dan hadist dapat disimpulkan bahwa sebagai seorang muslim diperintahkan untuk menaati seorang *ulil amri* atau pemimpin meskipun seorang budak sekalipun. Menaatin perintah serta kebijakan atau peraturan yang dibuat

merupakan suatu kewajiban selama tidak menimbulkan maksiat. Begitu juga tentang peraturan lalu lintas yang dibuat oleh pemerintah. Peraturan tersebut ditujukan untuk mengatur lalu lintas guna menciptakan ketertiban dalam berkendara. Adanya teknologi pada bidang lalu lintas menjadikan kemudahan dalam melakukan tugas tersebut. Teknologi *Intelligent Transportation System* merupakan salah satu yang sudah diimplementasikan dan terus dikembangkan. Maka dari itu, penulis menjelaskan faktor-faktor yang dapat mempengaruhi akurasi dan kecepatan Faster R-CNN dalam mendeteksi pengguna jalan yang dapat digunakan dalam pengembangan *Intelligent Transportation System (ITS)*.

## BAB V

### PENUTUP

#### 5.1. Kesimpulan

Berdasarkan penelitian yang dilakukan dengan metode Faster R-CNN dengan arsitektur MobileNet dan softNMS dapat ditarik kesimpulan sebagai berikut:

1. Pada hasil pengujian akurasi diketahui bahwa nilai *learning rate* dan *epoch* berpengaruh terhadap akurasi deteksi. Dari model 1 sampai model 20 yang diuji, model 15 merupakan model dengan akurasi tertinggi yaitu sebesar 96,84% dengan *epoch* 200 dan nilai *learning rate* 0,0001. Pemilihan jumlah *epoch* yang sesuai pada proses pelatihan menjadi penting karena *epoch* yang sedikit dapat dihasilkan akurasi yang sedikit atau bahkan menghasilkan *underfitting*, begitu juga *epoch* yang terlalu tinggi akan menghasilkan *overfitting*. Pemilihan nilai *learning rate* dapat mempengaruhi akurasi yang dihasilkan. Nilai *learning rate* yang kecil dapat meningkatkan tingkat akurasi yg didapat. Namun, nilai *learning rate* yang semakin kecil menjadikan proses pelatihan membutuhkan waktu lebih lama.
2. Sedangkan pada pengujian kecepatan, ukuran citra berpengaruh pada tingkat kecepatan deteksi. Model dengan ukuran citra 1920x1080 pixel merupakan terlambat dari model yang diujikan dengan kecepatan rata-rata 0,22 detik/citra. Sedangkan ukuran citra 576x324 pixel merupakan model tercepat dari model yang diuji, yaitu dengan kecepatan 0,16 detik/citra. Berdasarkan hasil tersebut, maka dapat disimpulkan bahwa ukuran citra berpengaruh terhadap kecepatan proses deteksi. Semakin kecil citra yang digunakan dalam deteksi semakin cepat pula kecepatannya. Namun, perlu

diperhatikan ukuran citra yang terlalu kecil juga akan menyulitkan proses deteksi dikarenakan citra yang dihasilkan menjadi kurang jelas dan terdapat banyak *noise*.

## 5.2. Saran

Penulis menyadari bahwa terdapat kekurangan pada penelitian yang telah dilakukan, maka dari itu perlu pengembangan agar hasil yang didapatkan lebih baik lagi. Berikut saran penulis untuk penelitian selanjutnya:

Penulis menyadari bahwa terdapat kekurangan dan kelemahan pada penelitian ini, sehingga masih perlu dilakukan pengembangan untuk mendapatkan hasil yang lebih baik. Beberapa saran penulis untuk penelitian selanjutnya adalah sebagai berikut:

1. Dataset pada penelitian ini menggunakan Pascal VOC dan dataset diambil sendiri di Kota Malang dengan kondisi siang hari dengan cuaca cerah, akan lebih baik pada pengambilan dataset di berbagai kondisi baik siang maupun malam untuk penelitian selanjutnya.
2. Arsitektur yang digunakan bisa ditingkatkan kembali dengan menggunakan data yang peneliti hasilkan dijadikan *transfer learning* untuk penelitian berikutnya.
3. Dapat digunakan metode *RoI Pooling* atau metode klasifikasi lain untuk mengambil fitur sehingga didapat hasil akurasi yang lebih bagus hasilnya.

## DAFTAR PUSTAKA

- Athiwaratkun, B., & Kang, K. (2015). Feature representation in convolutional neural networks. *ArXiv Preprint ArXiv:1507.02313*.
- Bengio, Y. (2009). *Learning deep architectures for AI*. Now Publishers Inc.
- Bodla, N., Singh, B., Chellappa, R., & Davis, L. S. (2017). *Soft-NMS--improving object detection with one line of code*. 5561–5569.
- Cao, C., Wang, B., Zhang, W., Zeng, X., Yan, X., Feng, Z., Liu, Y., & Wu, Z. (2019). An improved faster R-CNN for small object detection. *IEEE Access*, 7, 106838–106846.
- Christian, B., & Griffiths, T. (2017). *Algorithms to live by: The computer science of human decisions*. William Collins.
- Eda, K. (2020). *AI vs. Machine Learning vs. Deep Learning vs. Neural Networks: What's the Difference?* <https://www.ibm.com/cloud/blog/ai-vs-machine-learning-vs-deep-learning-vs-neural-networks>
- Felzenszwalb, P. F., & Huttenlocher, D. P. (2004). Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2), 167–181.
- Felzenszwalb, P., McAllester, D., & Ramanan, D. (2008). *A discriminatively trained, multiscale, deformable part model*. 1–8.
- Girshick, R. (2015). *Fast r-cnn*. 1440–1448.
- Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). *Rich feature hierarchies for accurate object detection and semantic segmentation*. 580–587.
- Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). *Mobilenets: Efficient convolutional*

- neural networks for mobile vision applications. *ArXiv Preprint ArXiv:1704.04861*.
- Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., & Guadarrama, S. (2017). *Speed/accuracy trade-offs for modern convolutional object detectors*. 7310–7311.
- Jalled, F., & Voronkov, I. (2016). Object detection using image processing. *ArXiv Preprint ArXiv:1611.07791*.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems, 25*, 1097–1105.
- Liu, J., Wang, Z., Wu, Y., Qin, Y., Cao, X., & Huang, Y. (2020). An Improved Faster R-CNN for UAV-Based Catenary Support Device Inspection. *International Journal of Software Engineering and Knowledge Engineering, 30(07)*, 941–959.
- Mo, N., & Yan, L. (2020). Improved faster RCNN based on feature amplification and oversampling data augmentation for oriented vehicle detection in aerial images. *Remote Sensing, 12(16)*, 2558.
- Nielsen, M. A. (2015). *Neural networks and deep learning* (Vol. 25). Determination press San Francisco, CA.
- Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *ArXiv Preprint ArXiv:1506.01497*.

Uijlings, J. R., Van De Sande, K. E., Gevers, T., & Smeulders, A. W. (2013).

Selective search for object recognition. *International Journal of Computer Vision*, *104*(2), 154–171.

Zhang, W., Zheng, Y., Gao, Q., & Mi, Z. (2019). Part-aware region proposal for vehicle detection in high occlusion environment. *IEEE Access*, *7*, 100383–100393.