IMPLEMENTASI DETEKSI OBJEK KENDARAAN MENGGUNAKAN METODE SSD (*SINGLE SHOT DETECTOR*)

TUGAS AKHIR

DISUSUN OLEH :

KHANIN TITIS WULANDARI

G.211.19.0070

PROGRAM STUDI S1 – TEKNIK INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI DAN KOMUNIKASI

UNIVERSITAS SEMARANG

2023

PERNYATAAN PENULIS TUGAS AKHIR

DENGAN JUDUL

IMPLEMENTASI DETEKSI OBJEK KENDARAAN MENGGUNAKAN METODE SSD (*SINGLE SHOT DETECTOR*)

Dengan ini saya :

NAMA : KHANIN TITIS WULANDARI

NIM : G.211.19.0070

PROGRAM STUDI : TEKNIK INFORMATIKA

"Saya menyatakan dan bertanggung jawab dengan sebenarnya bahwa Tugas Akhir (TA) ini adalah hasil karya saya sendiri kecuali cuplikan dan ringkasan yang masing-masing telah saya jelaskan sumbernya. Jika pada waktu selanjutnya ada pihak lain yang mengklaim Tugas Akhir (TA) ini sebagai karyanya, yang disertai dengan bukti-bukti yang cukup, maka saya bersedia untuk dibatalkan gelar Sarjana Komputer saya beserta segala hak dan kewajiban yang melekat pada gelar tersebut".

Semarang, 1 Agustus 2023

Penulis,

Khanin Titis Wulandari

PENGESAHAN TUGAS AKHIR

DENGAN JUDUL

IMPLEMENTASI DETEKSI OBJEK KENDARAAN MENGGUNAKAN METODE SSD (*SINGLE SHOT DETECTOR*)

OLEH

NAMA : Khanin Titis Wulandari
NIM : G.211.19.0070

DISUSUN DALAM RANGKA MEMENUHI SYARAT GUNA MEMPEROLEH GELAR SARJANA KOMPUTER PROGRAM STUDI S1 – TEKNIK INFORMATIKA JURUSAN TEKNOLOGI INFORMASI FAKULTAS TEKNOLOGI INFORMASI DAN KOMUNIKASI UNIVERSITAS SEMARANG

TELAH DIPERIKSA DAN DISETUJUI

SEMARANG, 25 Agustus 2023

KETUA PROGRAM STUDI S1 TEKNIK INFORMATIKA	PEMBIMBING TUGAS AKHIR
Khoirudin, S.Kom., M.Eng	Aria Hendrawan, M.Kom
NIS. 06557003102173	NIS. 06557003102159

DEKAN

Prind Triajeng Pungkasanti, S.Kom, M.Kom.
NIS. 06557003102110

PENGESAHAN UJIAN TUGAS AKHIR

DENGAN JUDUL

IMPLEMENTASI DETEKSI OBJEK KENDARAAN MENGGUNAKAN METODE SSD (*SINGLE SHOT DETECTOR*)

OLEH

NAMA : KHANIN TITIS WULANDARI

NIM : G.211.19.0070

Telah diujikan dan dipertahankan dihadapan Dewan Penguji pada Sidang Tugas Akhir (TA)

Hari Selasa tanggal 5/9 2023

Menurut pandangan kami, Tugas Akhir (TA) ini memadai dari segi kualitas maupun kuantitas untuk tujuan penganugrahan gelar Sarjana Komputer (S.Kom).

	Tanggal	Tanda Tangan
Ketua Tim Penguji		
Sri Handayani, S.T., M.T. NIS. 06557003102116	5 September '23	
Penguji Pendamping		
1. ARIA HENDRAWAN, S.T., M.Kom NIS. 06557003102159	5 September '23	
2. ASTRID NOVITA PUTRI, S. Kom., M. Kom. NIS. 06557003102179	5 September '23	

UNIVERSITAS SEMARANG
USM

ABSTRACT

Technological developments at the present time are very rapid, in Indonesia the population density is very high and the highways are always busy with various types of vehicles. Based on the recapitulation of accidents, a total of 1,457 accidents were recorded with 189 deaths, 186 serious injuries, 2,013 minor injuries in 2023. Unstable vehicle conditions such as failed brakes and failure of the public to maintain a safe distance can cause traffic accidents. Therefore, data on the number and classification of passing vehicles is needed. Therefore, we need a system that can calculate and recognize the types of vehicles passing by. Future research will create an implementation of the Single Shot Detector (SSD) method to identify types of vehicles. The aim of object detection is to detect 6 types of vehicles, namely motorbikes, cars, buses, trucks, box trucks, pickup cars. The problem that will be discussed is how to identify the type of vehicle using the SSD (Single Shot Detector) method. Single Shot Detector involves real-time object detection using the Vgg16 architecture, which is a CNN model that utilizes small (3x3) convolutional filters. The software in implementing this system uses LabelImg to process images in real-time. The VGGNet architecture combines the most important features of convolutional neural networks. The pre-drilled network can classify images into 1000 object categories.

Keywords : *Object Detection, Single Shot Detector (SSD), Real-Time, VGG-16, CNN*

UNIVERSITAS SEMARANG

USM

ABSTRAK

Perkembangan teknologi di masa sekarang ini sangat pesat, di indonesia kepadatan penduduk sangat tinggi dan jalan raya selalu ramai dilewati oleh berbagai macam jenis kendaraan. Berdasarkan rekapitulasi kecelakaan tercatat total sebanyak 1.457 kecelakaan dengan 189 korban meninggal dunia, luka berat 186, luka ringan 2.013 korban pada tahun 2023. Kondisi kendaraan yang tidak stabil seperti rem blong dan gagalnya masyarakat dalam menjaga jarak aman dapat menyebabkan kecelakaan lalu lintas. Oleh karena itu dibutuhkan data jumlah dan klasifikasi kendaraan yang lewat. Maka dari itu diperlukan sebuah sistem yang dapat menghitung dan mengenali jenis kendaraan yang lewat. Penelitian selanjutnya akan dibuat sebuah implementasi metode *Single Shot Detector* (SSD) untuk mengidentifikasi jenis-jenis kendaraan. Tujuan *object detection* adalah untuk mendeteksi 6 jenis kendaraan yaitu sepeda motor, mobil, *bus*, Truk, Truk *Box*, Mobil *Pickup*. Permasalahan yang akan dibahas, bagaimana mengidentifikasi jenis kendaraan dengan menggunakan metode SSD (*Single Shot Detector*). *Single Shot Detector* melibatkan deteksi objek *real-time* menggunakan arsitektur Vgg16, yang merupakan model CNN yang memanfaatkan konvolusional filter yang kecil (3x3). Perangkat lunak dalam pengimplementasian sistem ini digunakan LabelImg untuk pemrosesan gambar secara *real-time*. Arsitektur VGGNet menggabungkan fitur jaringan saraf konvolusi yang paling penting. Jaringan yang telah dilatih sebelumnya dapat mengklasifikasikan gambar ke dalam 1000 kategori objek.

Kata Kunci : Object Detection, Single Shot Detector (SSD), Real-Time, VGG-16, CNN

PEMBIMBING TUGAS AKHIR

Aria Hendrawan, M.Kom
NIS. 06557003102159

UNIVERSITAS SEMARANG
USM

KATA PENGANTAR

Puji syukur saya panjatkan kehadirat Allah SWT, karena berkat rahmat- Nya lah saya dapat menyelesaikan laporan Tugas Akhir dengan judul **"Implementasi Deteksi objek Kendaraan Menggunakan Metode SSD (Single Shot Detector)"**.

Saya menyadari bahwa laporan ini jauh dari kesempurnaan mengingat keterbatasan pengalaman dan kemampuan dalam menyusun laporan ini. Namun berkat bantuan dari semua pihak baik secara langsung maupun tidak langsung sehingga dapat terselesaikan laporan ini. Oleh karena itu pada kesempatan ini penulis mengucapkan terima kasih kepada :

1. Allah SWT yang telah melimpahkan karunia, rahmat, taufik, hidayah serta inayah-Nya.
2. Dr. Supari, S.T., M.T. selaku Rektor Universitas Semarang.
3. Prind Triajeng Pungkasanti, S.Kom., M.Kom, selaku Dekan Fakultas Teknologi Informasi dan Komunikasi Universitas Semarang.
4. Khoirudin, S.Kom., M.Eng. selaku Ketua Program Studi S1 Teknik Informatika Fakultas Teknologi Informasi dan Komunikasi Universitas Semarang.
5. Siti Asmiatun, S.Kom., M.Kom. selaku Koordinator Tugas Akhir.
6. Sri Handayani, S.T, M.T selaku dosen wali yang telah memberikan arahan dan mendampingi penulis hingga sampai ke tahap akhir ini.
7. Aria Hendrawan., M.Kom. selaku Dosen Pembimbing atas waktu yang telah diluangkan untuk arahan dan bimbingan dalam proses pembuatan laporan Tugas Akhir hingga selesai.
8. Seluruh Dosen dan Staf Fakultas Teknologi Informasi dan Komunikasi Universitas Semarang.

9. Kedua orang tua, Yermia Bogonoto dan Yuli Puspowati yang memberikan doa, perhatian, semangat dan dukungan baik secara moralmaupun material bagi penulis.
10. Novanto Bogo Prakoso yang selalu memberikan doa, perhatian dan dukungan kepada penulis.
11. Aditya Warman Wahyu Tri Nugroho yang selalu menemani danmenghibur penulis.
12. Muhammad Anis Rofiuddin yang senantiasa memberi dukungan, doa, kasih sayang dan nasehat.
13. Untuk teman – teman program studi Teknik Informatika angkatan 2019 yang tidak dapat disebutkan satu persatu, yang telah memberi semangat dan dukungan selama proses pembuatan Tugas Akhir dan penulisan laporan ini.
14. Semua pihak yang telah membantu dalam penyusunan laporan ini yang tidak dapat penulis sebutkan satu persatu, semoga Allah SWT memberikan balasan atas semua kebaikannya dengan yang baik.

Saya sadar bahwa dalam pengerjaan laporan ini masih terdapat kekurangan, untuk itu penulis mengharapkan kritik dan saran yang sifatnya membangun. Semoga laporan ini bisa bermanfaat khususnya bagi penulis sendiri dan umumnya bagi pembaca.

Semarang, 1 Agustus 2023

Penulis

DAFTAR ISI

UNIVERSITAS SEMARANG
USM

UNIVERSITAS SEMARANG
USM

UNIVERSITAS SEMARANG
USM

DAFTAR GAMBAR

UNIVERSITAS NEGERI SEMARANG
USM

UNIVERSITAS SEMARANG
USM

DAFTAR TABEL

UNIVERSITAS SEMARANG
USM

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi di masa sekarang ini sangat pesat, di indonesia kepadatan penduduk sangat tinggi dan jalan raya selalu ramai dilewati oleh berbagai macam jenis kendaraan. Berdasarkan rekapitulasi kecelakaan tercatat total sebanyak 1.457 kecelakaan dengan 189 korban meninggal dunia, luka berat 186, luka ringan 2.013 korban pada tahun 2023. Kondisi kendaraan yang tidak stabil seperti rem blong dan gagalnya masyarakat dalam menjaga jarak aman dapat menyebabkan kecelakaan lalu lintas. Untuk mengetahui lokasi titik kecelakaan dibutuhkan data jumlah kendaraan yang melewati lokasi tersebut. Semakin banyak kendaraan yang lewat berarti semakin tinggi resiko terjadinya kecelakaan. Di setiap jalan raya terdapat berbagai jenis kendaraan yang lewat seperti mobil, truk, bus dan sepeda motor. Data setiap jenis kendaraan yang lewat dapat digunakan untuk mengidentifikasi jenis kendaraan yang mengalami kecelakaan lalu lintas. Untuk menghitung jumlah kendaraan yang lewat setiap hari secara otomatis dapat menggunakan aplikasi dengan menerapkan sistem cerdas

Object Detection atau Deteksi objek dalam digital *image processing* adalah suatu proses yang digunakan untuk menetukan keberadaan objek tertentu di dalam suatu citra digital. Proses deteksi tersebut dapat dilakukan dengan berbagai macam metode yang umumnya melakukan pembacaan fitur-fitur dariseluruh objek pada citra input. Aplikasi atau sistem deteksi objek saat ini banyak digunakan dalam membantu berbagai macam kepentingan manusia, salah satunya pengaturan lalu lintas. Deteksi objek dalam mengatur lalu lintas diantaranya menghitung kendaraan, mendeteksi kecepatan, dan pelanggaran lalu lintas. Terdapat beberapa model yang dapat digunakan dalam mendeteksi objek pada gambar ataupun video, diantaranya adalah *Faster R- CNN*, *YOLO*, *MobileNet*, *Mask R-CNN* dan sebagainya. Kebanyakan model objek deteksi memanfaatkan wilayah pada gambar dalam menentukan objek. Sehingga model tidak melihat gambar secara

lengkap. *Faster R-CNN* adalah model objek yang bagus dengan komputasi yang tidak terlalu berat untuk menyelesaikan masalah deteksi objek. Model *Mask R-CNN* merupakan perkembangan dari *Faster R-CNN* namun lebih fokus untuk masalah *Object Segmentation* (Nagataries et al., 2012).

Pada model *Mask RCNN* objek dalam gambar tidak hanya diberikan *bounding box*, namun objek juga dilapisi semacam topeng yangmembentuk persis objek tersebut. Sampai saat ini sudah banyak penelitian yang mengarah pada permasalahan ini, khususnya penggunaan metode *face detection* menggunakan *haar-cascade classifier* yang bisa menghasilkan akurasi 94%, *tracking* menggunakan *extended CAMSHIFT* oleh Bobby Yuliandra dari Fakultas Informatika – Telkom University (Yuliandra, 2012). Adapun Penelitian lainnya sistem penghitung dan identifikasi wajah manusia dengan metode *Background Substraction* dan *haar cascade* oleh Ijon Posmarohatta Sinaga dari Fakultas Teknik Elektro, Universitas Telkom (Sinaga, 2017). Pada penelitian Nafi Ur Rashid, Niluthpol Chowdhury Mithun, Bhadhan Roy Joy,dan S. M. Mahbubur Rahman yang berjudul "*Detection And Classification of Vehicles from A Video Using Time-Spatial Image*" deteksi hitung kendaraan memiliki rata-rata akurasi 59.74%. Penelitian lain oleh Katanyoo Klubsuwan dan Wittaya Koodtalang yang berjudul "*Traffic Violation Detection Using Multiple Trajectories Evaluation of Vehicle*" menunjukan hasil deteksi pelanggaran lalu lintas untuk tipe kendaraan kecil sebesar 78%, sedang 85% dan besar 74%. Penelitian lainnya pendeteksian dan penghitungan kendaraan secara *real-time* menggunakan algoritma *Haar Feature-Based Classifier*, dalampenelitian ini juga terdapat biaya sensor, prosessor dan CCTV sangat mempengaruhi pendeteksian. Kemungkinan akurasi pendeteksian dipengaruhi oleh kecepatan pendeteksian objek dari pendeteksian *frame* per *second* karena kecepatan ekstraksi fitur dari setiap *frame* mempengaruhi akurasi pendeteksian objek.

Single Shot Multibox Detector (SSD) merupakan salah satu algoritma pendeteksian *object* berupa gambar paling populer, hal tersebut dikarenakan algoritma SSD mudah diimplementasikan. Hasil akurasi pada algoritma SSD cukup

baik relatif terhadap komputasi yang dibutuhkan. Metode ini termasuk kedalam pendeteksian objek *real time*. Cara kerja pada algoritma SSD didasarkan *pada feed-forward convolutional network*, yaitu pada suatu objek akan ditentukan *bounding boxes* yang berukuran tetap dan pada area *bounding boxes* tersebut akan menampilkan nilai skor masing-masing pada setiap kelas objek. arsitektur VGG16 dan *extra feature layers*, kedua arsitektur jaringan inilah yang ada pada arsitektur algoritma SSD. Kinerja pada jaringan VGG-16 sangat kuat dengan kualitas yang tinggi untuk citra gambar sehingga digunakan sebagai *based network*. Bagian lapisan fitur konvolusional ditambahkan ke ujung *base network* oleh SSD sebanyak enam layer konvolusi ekstra yang berfungsi memprediksi dengan aspek rasio berbeda (Doavers, 2018).

SSD dirancang untuk deteksi objek secara *real-time*. *R-CNN* yang lebih cepat menggunakan jaringan proposal wilayah untuk membuat kotak batas dan menggunakan kotak tersebut untuk mengklasifikasikan objek. Meskipun dianggap sebagai yang paling awal dalam akurasi, seluruh proses berjalan pada 7 *frame* per detik, jauh di bawah kebutuhan pemrosesan *real-time*. SSD mempercepat proses dengan meniadakan kebutuhan akan jaringan proposal wilayah. Untuk memulihkan penurunan akurasi, SSD menerapkan beberapa peningkatan termasuk fitur multi-skala dan kotak *default*. Peningkatan ini memungkinkan SSD untuk menyamai akurasi *Faster R-CNN* menggunakan gambar beresolusi lebih rendah, yang selanjutnya mendorong kecepatan lebih tinggi.

Pada penelitian ini mengangkat judul “Implementasi Deteksi Objek Kendaraan menggunakan Metode SSD (*Single Shot Detector*)”. Dimulai dengan analisis penerapan kecerdasan buatan untuk identifikasi setiap kendaraan yang terdeteksi oleh kamera keamanan, penyelesaian tugas akhir ini hanya terbatas pada identifikasi jenis kendaraan. Deteksi ini bisa dilakukan dengan cara perekaman secara *real-time* dengan kamera pengawas otomatis, yang kemudian diproses pada kecerdasan buatan. Metode *Single Shot Detector* (SSD) dikarenakan memiliki akurasi tinggi untuk mendeteksi kendaraan, maka penelitian selanjutnya akan

dibuat sebuah implementasi metode *single shot detector* (SSD) untuk pengenalan jenis kendaraan seperti motor, mobil, truk, truk *box*, mobil *pick up* dan *bus*.

1.2 Rumusan Masalah

Berdasarkan latar belakang penelitian, permasalahan yang akan dibahas dalam penelitian ini adalah bagaimana mengolah *dataset* kendaraan di lalu lintas menggunakan Metode *Single Shot Detector* (SSD)?

1.3 Batasan Masalah

Mengenai penerapan metode SSD (*Single Shot Detector*) untuk pendeteksian jenis kendaraan, penulis membuat beberapa batasan masalah agar pembahasan lebih fokus dan sesuai dengan tujuan yang ingin dicapai, adapun batasan masalahnya adalah sebagai berikut:

1. Metode yang digunakan untuk deteksi kendaraan adalah Metode *Single Shot Detector* (SSD)
2. *Dataset* yang digunakan untuk pengujian diambil dari penelitian sebelumnya (Hendrawan et al., 2022)
3. Tujuan pendeteksian objek adalah untuk mendeteksi suatu jenis kendaraan dengan 6 jenis kendaraan yaitu sepeda motor, mobil, *bus*, Truk, Truk *Box*, Mobil *Pickup*.
4. Penelitian ini memiliki ruang lingkup khusus yaitu kamera keamanan yang dipasang pada sudut tertentu, terdeteksi objek yang tidak tertutupi oleh apapun, mengambil gambar saat kendaraan berada dalam jangkauan kamera, pencahayaan ideal.

1.4 Tujuan Tugas Akhir

Berdasarkan rumusan masalah yang telah diuraikan di atas, maka tujuan dari penelitian ini adalah implementasi pada metode *Single Shot Detector* (SSD) untuk mengidentifikasi 6 jenis kendaraan (Mobil/*Car*, Truk/*Truck*, *Bus*, Sepeda Motor/*Motorcycle*, Mobil Pengangkut/*Pickup Car*, Truk Bok/*Truck Box*) di lalu

lintas. Sehingga akurasi deteksi dapat dimaksimalkan dan menghasilkan model yang bermanfaat untuk studi kasus yang serupa dimasa mendatang.

1.5 Manfaat Tugas Akhir

Manfaat yang didapat dan diharapkan dari penyusunan Laporan Tugas Akhir ini adalah sebagai berikut:

1. Bagi Penulis

   Penulis dapat lebih memahami cara kerja pembelajaran *Machine learning* terutama dalam hal deteksi objek dengan metode *Single Shot Detector* (SSD)

2. Bagi Akademik

   Menjadi bahan referensi yang dapat digunakan untuk pembanding serta acuan terhadap kerangka acuan untuk masalah sejenis.

3. Bagi Universitas Semarang

   Menambah pembendaharaan literatur di perpustakaan, terutama perpustakaan Fakultas Teknologi Informasi dan Komunikasi Universitas Semarang serta sebagai salah satu tolak ukur dalam topik pengembangan implementasi pembelajaran *Machine Learning*.

1.6 Metodologi Penelitian

1. Jenis Data

   a. Data Primer

      Dalam penelitian ini, data yang digunakan adalah data gambar yang telah melalui proses pelabelan, setelah itu melalui *preprocessing* data dengan mengubah ukuran setiap gambar menjadi 480x480 piksel dan melalui metode *augmentasi* data *mosaic* untuk data orisisnil dari penelitian sebelumnya (Hendrawan dkk., 2022).

   b. Data Sekunder

Kemudian ada 134 data gambar tambahan, yang melalui proses pelabelan dan *preprocessing* dengan mengubah ukuran menjadi 480x480 piksel mengikuti data asli. Format pelabelan atau anotasi data yang digunakan dalam penelitian ini adalah COCO (.json) dan juga PascalVOC(.xml).

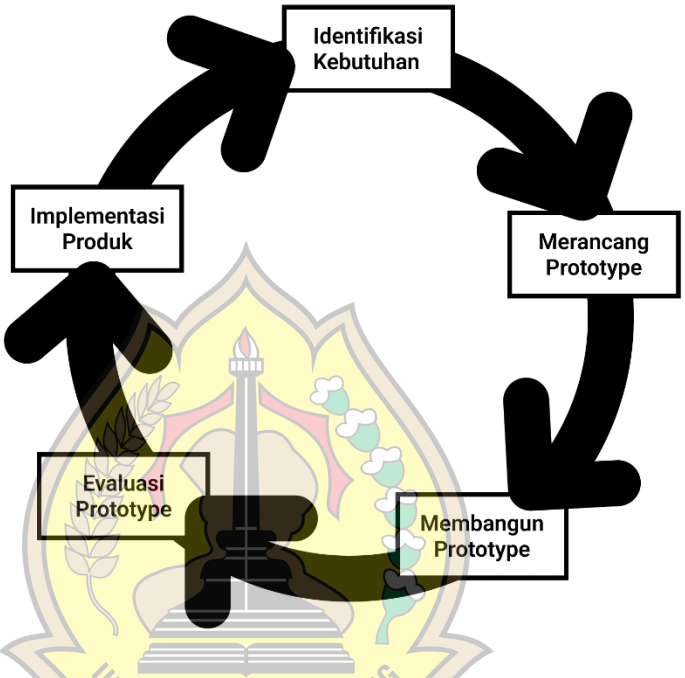
2. Metode Pengumpulan Data

Metode pengumpulan data yang digunakan dalam penelitian ini adalah Studi Literatur. Dimana data yang digunakan dalam penelitian ini menggunakan data dari penelitian sebelumnya (Hendrawan dkk., 2022), yang mana diambil dari kamera CCTV Jalan Perintis Kemerdekaan, Banyumanik, Semarang dengan jumlah sebanyak 1104 data gambar.

UNIVERSITAS SEMARANG
USM

3. Metode Pengembangan Sistem

Dalam penelitian ini, fokus penulis hanya mencari tahu cara implementasi Metode *Single Shot Detector* terhadap *dataset* lalu lintas.



Gambar 1. 1 Tahap Prototyping

Pada Gambar 1. 1 adalah proses tahapan metode *prototype* yang digunakan penulis. Penjelasan dari tahap-tahap tersebut adalah sebagai berikut:

1. Pemahaman Konsep Dasar

   Pemahaman mengenai SSD (*Single Shot Detector*) dan arsitektur VGG16. SSD adalah jaringan saraf tiruan yang dapat mendeteksi objek dalam berbagai skala dan lokasi secara efisien, sementara VGG16 adalah arsitektur jaringan saraf tiruan yang sering digunakan untuk pengenalan gambar.

2. Merancang *Prototype*

   Dalam tahap ini, penulis melakukan perancangan *prototype* dengan mempersiapkan *dataset* yang akan digunakan dalam proses *training*.

Selain itu penulis juga mempersiapkan kebutuhan *software* dan *hardware*.

3. Pemodelan Jaringan

   Bangun model jaringan dengan menggunakan arsitektur VGG16 sebagai bagian dari model SSD. dengan menggunakan berbagai *framework deep learning* seperti *TensorFlow* atau *PyTorch* untuk membangun model ini. Anda perlu menghapus lapisan akhir (lapisan klasifikasi) dari VGG16 dan menggantinya dengan lapisan yang sesuai untuk tugas deteksi objek.

4. Membangun *Prototype*

   Dalam tahap ini, penulis melakukan proses *training* dengan *dataset*, *software*, dan *hardware* yang digunakan. Proses *training* menggunakan fungsi kerugian (*loss function*) yang sesuai, seperti kerugian SSD, yang menggabungkan komponen kerugian deteksi objek dan kerugian pembatas. Pelatihan dilakukan dengan menggunakan stokastik gradien turun (*stochastic gradient descent*) atau optimasi lainnya. Setelah proses *training* selesai, nantinya akan dihasilkan model *deep learning* menggunakan metode *Single Shot Detector*.

5. Evaluasi *Prototype*

   Dalam tahap ini, penulis melakukan evaluasi terhadap model *deep learning* yang telah disimpan dengan melakukan proses kalkulasi *precision*, *recall*, *Average Precision* (AP), IoU (*Intersection over Union*) dan *mean Average Precision* (mAP).

6. Implementasi Aplikasi

   Pada tahap ini, penulis melakukan implementasi model deteksi objek kedalam aplikasi atau sistem yang digunakan.

1.7 Sistematika Penulisan

Rangkuman dan uraian singkat yang memuat struktur penulisan secara umum pada penelitian ini dijelaskan pada sistematika penulisan sebagai berikut:

BAB I PENDAHULUAN

Pendahuluan menjelaskan mengenai masalah - masalah yang terdapat pada sub bab yang berada didalamnya. Sub bab tersebut meliputi dari latar belakang, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, metodologi penelitian dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Bab ini menjelaskan tinjauan pustaka dari penilitian-penelitian sebelumnya dan teori-teori terkait Tugas Akhir (TA). Peneliti akan mengidentifikasi area yang perlu diteliti dan diterpakan, seputar penelitian sebelumnya.

BAB III METODOLOGI PENELITIAN

Menjabarkan metode dan alur yang digunakan untuk melakukan penelitian.

BAB IV HASIL DAN PEMBAHASAN

Pada bab ini berisi hasil dari penelitian yang telah dilakukan, serta memberikan pembahasan lebih lanjut seputar hasil penelitian.

BAB V PENUTUP

Penutup berisi kesimpulan yang menjawab rumusan masalah dari hasil penelitian oleh peneliti dan saran atas apa yang sudah dilakukan sehingga dapat dilaksanakan pada penelitian ke depannya.

DAFTAR PUSTAKA

Berisi daftar yang digunakan untuk dirujuk dalam penulisan Tugas Akhir.

LAMPIRAN

Berisi informasi yang ada hubunganya dengan isi laporan Tugas Akhir.

BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Sebelumnya

Rujukan penelitian sebelumnya digunakan sebagai referensi *Deep Learning* menggunakan Metode *Single Shot Detector* (SSD) dalam mendeteksi kendaraan di jalan raya. Seluruh penelitian pada Tabel 2. 1 menunjukkan beberapa metode yang digunakan untuk deteksi objek, baik objek kendaraan maupun objek lain.

Tabel 2. 1 Telaah Artikel

No	Nama Peneliti dan Tahun	Judul	Metode Penelitian	Hasil
1.	(Pang, Yanwei Wang, Tiancai Anwer, Rao Muhammad Khan, Fahad Shahbaz Shao, Ling, 2019)	*Efficient featurized image pyramid network for single shot detector*	Arsitektur deteksi keseluruhan yang digunakan LFIP-SSD menggunakan VGG-16 sebagai tulang punggung dan menambahkan serangkaian lapisan konv yang semakin kecil. Berbeda dengan SSD standar, LFIP berisi *downsampling* iteratif dan *convolutional block*. Fitur dari LFIP kemudian dimasukkan ke dalam lapisan SSD	100ms untuk memproses gambar. Untuk input 512 × 512, mengikuti strategi mengambil sampel peta fitur conv7 dan menggabungkanya dengan conv4 3 setelah menerapkan modul LFIP. RetinaNet dan RFBNet memperoleh skor AP 34,4. (ukuran *input* 512×512) mencapai AP sebesar 34,6%

Tabel 2. 1 Telaah Artikel (lanjutan)

			standar menggunakan attention module. Fitur yang dihasilkan dari lapisan saat ini kemudian digabungkan dengan pasangan lapisan sebelumnya dalam modul fusi maju.	sekaligus relatif lebih cepat (29 mdtk) dibandingkan dengan RFBNet (33 mdtk). Selanjutnya, memberikan peningkatan 3 kali lipat melalui RetinaNet.
2.	(Fu, Cheng-Yang Shvets, Mykhailo Berg, Alexander C., 2019)	*RetinaMask: Learning to predict masks improves state-of-the-art single-shot detection for free*	YOLOv3 dilatih dengan pelatihan multi-skala RetinaNet. RetinaMask memiliki amplop yang lebih tinggi untuk akurasi-vs- waktu daripada RetinaNet saat menggunakan ResNet-50 dan ResNet-101 untuk model backbone.	Model menunjukkan peningkatan 1,84 mAP dan 1,52 mAP pada ResNet-50 dan ResNet-101 dibandingkan dengan RetinaNet. Hasil deteksi lebih baik dari angka asli dari Mask R-CNN dan sangat dekat dengan hasil implementasi terbaru. Semua hasil menggunakan

Tabel 2. 1 Telaah Artikel (lanjutan)

				ResNet-101 dan Feature Pyramid Network sebagai model tulang punggung. Model dilatih dengan cara yang sangat mirip dengan pelatihan +e2e. Mask R-CNN masih menunjukkan akurasi prediksi mask yang lebih baik, namun selisihnya hanya sekitar 1,2 mAP
3.	(Sukusvier i, Andrianto, 2020)	Implemen-tasi Metode *Single Shot Detector* untuk Pengenalan Wajah	Metode penelitian yang digunakan adalah *image acquisition*, yaitu pengambilan *frame* pada saat perekaman video secara *realtime*. Citra RGB di ubah menjadi citra *grayscale* menggunakan metode SSD (*single shot detector*). Tahap	Dalam proses deteksi wajah dengan berbagai sudut pandang menggunakan metode *Single Shot Detector* (SSD) memiliki tingkat akurasi 100%. Perubahan jarak antara object terhadap kamera tidak terlalu

UNIVERSITAS SEMARANG
USM

Tabel 2. 1 Telaah Artikel (lanjutan)

			selanjutnya *classification*, yaitu proses pengklasifikasian program untuk melakukan *template matching* dengan menggunakan *face recognition*. Proses terakhir merupakan kesimpulan hasil akhir.	berpengaruh terhadap proses pendeteksian wajah. Tingkat akurasi pengenalan atau identifikasi wajah dengan berbagai sudut pandang memiliki akurasi 88%. Sedangkan untuk pengujian pengenalan wajah menghadap ke depan (*frontal face*) dengan jarak antara 60 cm hingga 250 cm memiliki akurasi 100%.
4.	(Fuady, Samratul Nehru, Nehru Anggraeni ,Gina, 2020)	Deteksi Objek Menggu-nakan Metode *Single Shot Multibox Detector* Pada Alat	Input pada sistem ini yaitu berasal dari kamera yang digunakan menggambil citra digital. Kamera akan menangkap citra digital secara *realtime* dan	Data ini dihitung akurasi, sensitifitas, dan spesifisitas. Secara keseluruhan, akurasi sistem berada di angka 92%, kesalahan banyak terjadi saat

UNIVERSITAS SEMARANG
USM

Tabel 2. 1 Telaah Artikel (lanjutan)

		Bantu Tongkat Tunanetra Berbasis Kamera	mengirimkanya ke Raspberry Pi. Perangkat lunak dalam mengimplementasik-an sistem ini digunakan OpenCV. Neural Network akan mengklasifikasikan objek sesuai dengan *dataset* yang sudah tersedia. Klasifikasi objek yang dilatih pada *dataset Common Object in Context Detection Challenge*. Proses ini membentuk sebuah filter dengan panjang dan tinggi (*pixels*). Citra akan diubah ke 300x300 pixels dengan kedalaman 3 *channels* yaitu RGB (*Red, Green, Blue*). Selanjutnya adalah tahap *pooling* yang menggunakan metode *max pooling*	pendeteksian hewan. Untuk sensitifitas adalah sebesar 83% dan spesifisitasnya 100%. Pada penelitian ini telah dirancang alat bantu tongkat tunanetra untuk pendeteksian objek berbasis kamera.

Tabel 2. 1 Telaah Artikel (lanjutan)

			menentukan nilai maksimum pada setiap nilai *input* yang telah dikonvolusi dengan filter. Terakhir, adalah proses *fully connected layer* yaitu tahap yang menghubungkan konvolusi dan *pooling*. Pada proses ini piksel yang dianggap sebagai halangan akan menjadi sebuah *output* yang terdiri dari salah satu kelas label.	
5.	(Hendrawan, Aria Gernowo, Rahmat Nurhayati, Oky Dwi Warsito, Budi Wibowo, Adi, 2022)	*Improve-ment Object Detection Algorithm Based on YoloV5 with Bottleneck CSP*	YOLOV5-BottleNeckCSP yang ditingkatkan dengan *augmentation data mosaic* untuk mendeteksi kendaraan di jalan raya.	Hasil YOLOv5 yang ditingkatkan mencapai mAP dengan IoU 50 sebesar 0.984, sedangkan mAP dengan IoU 50-95 sebesar 0.696. Nilai presisi 0.95, dan recall 0.98.

Tabel 2. 1 Telaah Artikel (lanjutan)

6.	(Assidhiqi, Fatih, 2021)	Pengembangan Sistem Deteksi Hunian Parkir Menggunakan Metode *Convolutional Neural Network*	Menggunakan Metode *Convolutional Neural Network* dengan arsitektur VGG16 yang diimplementasikan menggunakan aplikasi android, sehingga pengendara dapat menggunakannya untuk mencari informasi dimana letak tempat parkir yang kosong.	Hasil akurasi dan loss terbaik yang dipilih oleh peneliti adalah modifikasi VGG16 dengan akurasi pada training set sebesar 99,75% dan loss sebesar 0,0090, sedangkan pada validation set akurasi yang didapat yaitu 99,89% dan loss sebesar 0,0045, serta pada test set hasil akurasinya adalah 99,09% dan loss sebesar 0,0365.

Penelitian pada Tabel 2. 1 menunjukkan beberapa metode deteksi objek yang digunakan, seperti YOLO, CNN, dan *Single Shot Detector* (SSD). Metode-metode tersebut diterapkan untuk mendeteksi berbagai objek, termasuk kendaraan, bentik laut, musik/suara, dan manusia. Perbedaan yang dimiliki dari penelitian ini dapat menjadi referensi dalam *object detection* pada beberapa jenis kendaraan dijalan

raya dengan menggunakan metode *Single Shot Detector* (SSD) dengan arsitektur VGG16.

2.2 Object Detection

Salah satu pengembangan dari *Task Image Classification* adalah *Object Detection*. *Task* dari *Object Detection* adalah bagaimana membuat mesin dapat mengenali beberapa objek dan menentukan posisi objek-objek tersebut di dalam sebuah gambar. Konsep *Object Detection* secara sederhana yaitu dengan melakukan *scanning* pada seluruh bagian gambar dan menentukan mana yang objek dan mana yang bukan objek (*background*).

Deteksi objek adalah proses menemukan *instance* objek dari kelas tertentu, seperti wajah, mobil, dan pohon, dalam gambar atau video. Tidak seperti klasifikasi, deteksi objek dapat mendeteksi banyak objek, serta lokasinya di gambar. Detektor objek akan mengembalikan daftar objek yang terdeteksi dengan informasi kelas objek, probabilitas, dan koordinatnya untuk setiap objek. (Vasilev, Ivan. 2019). Perbedaan *classification* dan *object detection* bisa dilihat pada Gambar 2. 1 berikut:

Gambar 2. 1 *Object Detection*

2.3 Pengolahan Citra

Pengolahan citra digital merupakan pemrosesan gambar maupun gambar bergerak berdimensi dua melalui komputer digital. Secara umum hal ini dibagi menjadi dua proses yaitu memperbaiki kualitas suatu gambar sehingga dapat lebih mudah diinterpretasi oleh manusia dan mengolah informasi yang terdapat pada

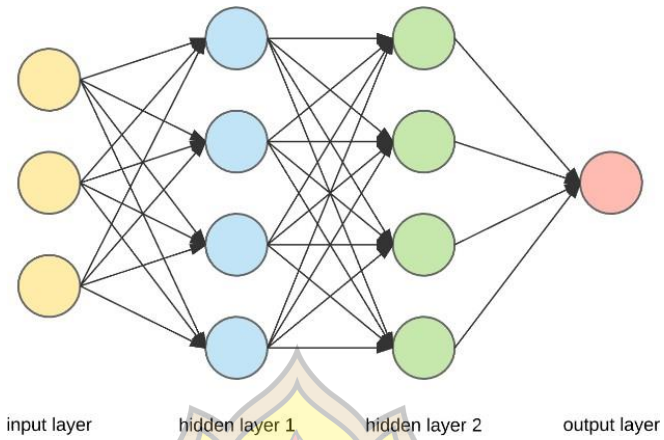
suatu gambar untuk pengenalan objek secara otomatis. Pengolahan citra melibatkan berbagai teknik dan algoritma, seperti konvolusi, transformasi, segmentasi, deteksi tepi, ekstraksi fitur, dan masih banyak lagi.

2.4 Deep Learning

Deep Learning merupakan cabang ilmu dari *Machine Learning* berbasis jaringan saraf tiruan untuk implementasi suatu permasalahan tertentu. Dalam *Deep Learning*, sebuah komputer dapat belajar mengklasifikasikan gambar, suara, teks atau video secara langsung. Komputer dilatih menggunakan *dataset* berlabel dan berjumlah besar yang kemudian mengubah nilai piksel dari sebuah gambar menjadi suatu representasi untuk mendeteksi atau mengklasifikasi pola pada masukan *input* (Lecun, Bengio, and Hinton 2015).

Dalam *Machine Learning* terdapat teknik untuk menggunakan ekstraksi fitur dari sebuah data untuk klasifikasi citra atau mengenali suara. Namun, metode ini masih memiliki beberapa kekurangan dalam hal kecepatan dan akurasi. Konsep jaringan syaraf tiruan yang dalam (banyak lapisan) dapat diimplementasikan pada *Machine Learning* yang sudah ada, sehingga komputer dapat belajar dengan kecepatan, akurasi, dan skala yang besar. Prinsip tersebut berkembang hingga *Deep Learning* semakin sering digunakan pada komunitas riset dan industri untuk membantu memecahkan banyak permasalahan dengan skala yang besar, seperti *Computer Vision*, *Speech Recognition*, dan *Natural Language Processing*. *Feature Engineering* merupakan salah satu fitur utama dari *Deep Learning* untuk ekstraksi pola yang berguna dari sebuah data yang dapat memudahkan model untuk membedakan kelas. *Feature Engineering* juga merupakan teknik yang paling penting untuk mencapai hasil yang baik pada tugas prediksi. Namun, hal ini sulit untuk dipelajari dan dikuasai, dikarenakan kumpulan jenis data yang berbeda, maka memerlukan pendekatan teknik yang berbeda juga.

Lapisan pada *Deep Learning* terdiri atas tiga bagian yaitu *input layer*, *hidden layer*, dan *output layer* seperti pada Gambar 2. 2 berikut:



Gambar 2. 2 Layer pada *Deep Learning*

Input layer berisi *node-node* yang menyimpan sebuah nilai *input* yang tidak berubah pada fase pelatihan dan hanya bisa berubah jika diberikan nilai *input* baru. Pada *hidden layer* dapat dibuat berlapis-lapis untuk menemukan komposisi algoritma yang tepat agar dapat mengurangi nilai *error* pada *output*. *Output layer* berfungsi untuk menampilkan hasil perhitungan sistem oleh fungsi aktivasi pada *hidden layer* berdasarkan nilai *input*. Dengan menambahkan lebih banyak lapisan menjadikan model pembelajaran yang bisa mewakili citra berlabel dengan lebih baik.

USM

2.5 Machine Learning

Machine Learning atau pembelajaran mesin pada prosesnya mementingkan generalisasi sebagai pencarian melalui kemungkinan hipotesis. Teknik mesin pembelajaran dibedakan dua yaitu Regresi dan Klasifikasi (Ian H. Witten, Eibe, Frank, 2011). (Dasgupta & Nath, 2016) menjelaskan bahwa Algoritma pembelajaran yang diawasi melalui data pelatihan dan menghasilkan aturan umum (fungsi), yang dapat 22 digunakan untuk memetakan input baru. Pembelajaran yang diawasi pada pelatihan terdiri dari vektor *input* x dan *output* target yang terbagi menjadi 2 cara yaitu:

1. Regresi : *Output* target merupakan bilangan real atau seluruh vektor bilangan real.
2. Klasifikasi : *Output* target adalah memberikan label kelas pada setiap kasus yang ada dengan melihat tingkat alternatif yang ada.

Beberapa *machine learning* berbasis jaringan tiruan sering dipergunakan dalam berbagai hal salah satunya sebagai fungsi peramalan atau prakiraan. Jaringan tiruan seperti *Artificial Neural Network* (ANN) sudah sering dipergunakan tetapi pada tipe ANN ditemukan beberapa kelemahan seperti penggunaan waktu yang relatif lama, untuk itu mulai dikembangkannya beberapa model jaringan tiruan untuk mengatasi masalah tersebut (Sacchetti et al., 2017; Ahmad et al., 2014).

Machine Learning merupakan salah satu cabang dari disiplin ilmu Kecerdasan Buatan (*Artificial Intellegence*) yang membahas mengenai pembangunan sistem yang berdasarkan pada data. Banyak hal yang dipelajari, akan tetapi pada dasarnya ada 4 hal pokok yang dipelajari dalam *machine learning*.

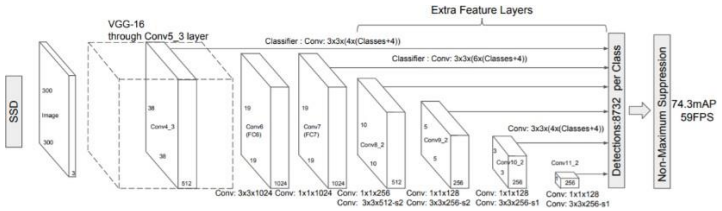
1. Pembelajaran Terarah (*Supervised Learning*)
2. Pembelajaran Tak Terarah (*Unsupervised Learning*)
3. Pembelajaran Semi Terarah (*Semi-supervised Learning*)
4. *Reinforcement Learning*

UNIVERSITAS SEMARANG
USM

2.6 Single Shot Detector

Single Shot Detector (SSD) adalah metode untuk mendeteksi atau mengidentifikasi objek dalam gambar menggunakan jaringan saraf dalam tunggal dan merupakan salah satu algoritma pendeteksian objek yang paling populer karena kemudahan penggunaan dan akurasi yang baik. Metode SSD (*Single Shot Detector*) melibatkan deteksi objek *real-time* menggunakan arsitektur Vgg16, yang merupakan model CNN yang memanfaatkan konvolusional filter yang kecil (3x3). Untuk perangkat lunak dalam pengimplementasian sistem ini digunakan LabelImg untuk pemrosesan gambar secara *realtime*, menggunakan metode SSD metode

Single Shot Detector (SSD) gambar asli dari kamera akan diekstrasi oleh *layer* konvolusi, setelah itu *neural network* mengklasifikasikan objek sesuai dengan *dataset* yang ada. Lapisan SSD menggunakan arsitektur VGG16 bisa dilihat pada Gambar 2. 3 berikut:



Gambar 2. 3 *Single Shot Detector* (SSD) Arsitektur VGG16

Pada Arstitektur SSD termasuk kedalam jenis *Convolutation Neural Network*, *Convolutional Neural Network* (CNN) adalah salah satu jenis *Neural Network* yang biasa digunakan pada *data image*. CNN bisa digunakan untuk mendeteksi dan mengenali *object* pada sebuah *image*.

Arsitektur dari CNN dibagi menjadi 2 bagian besar, *Feature Extraction Layer* dan *Convolutional Layer*. Dimana pada bagian *Feature Extraction Layer* ini adalah melakukan "*encoding*" dari sebuah *image* menjadi *features* yang berupa angka-angka yang merepresentasikan *image* tersebut, sedangkan dibagian *Convolutional Layer* terdiri dari neuron yang tersusun sedemikian rupa sehingga membentuk sebuah filter dengan panjang dan tinggi (*pixels*). Secara matematis, *Convolutional layer* atau yang dalam Bahasa Indonesianya konvolusi, adalah integral yang mencerminkan jumlah lingkaran dari sebuah sudut fungsi F yang digeser atas fungsi g sehingga menghasilkan fungsi h.

Konvolusi dilambangkan dengan arsterik (*). Sehingga, *F*g=h* berarti fungsi F dikonvolusikan dengan fungsi g menghasilkan fungsi h. konvolusi dua buah fungsi *F(x)* dan *g(x)* di definiskan seperti Gambar 2. 4 berikut:

$$h\,(x) = F(x) * g(x) = \int F\,(a) g(x - a)$$

Gambar 2. 4 Konvolusi dua buah fungsi *F(x)* dan *g(x)*

Untuk fungsi diskrit, konvolusi di definisikan seperti Gambar 2. 5 berikut:

$$h\,(x) = F(x) * g(x) = \sum_{-\infty}^{\infty} F\,(a)g(x-a)$$

Gambar 2. 5 Konvolusi fungsi diskrit

g(x) disebut dengan kernel konvolusi (filter). Kernel *g(x)* merupakan jendela yang dioperasikan secara bergeser pada sinyal masukan *F(x)*. Hasil konvolusi dinyatakan dengan keluaran *h(x)*. Contoh, misal citra *F(x,y)* yang berukuran 5x5 sebuah kernel dengan 3x3 matriks seperti Gambar 2. 6 berikut:

$$g(x,y) = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad F(x,y) = \begin{bmatrix} 4 & 4 & 3 & 5 & 4 \\ 6 & 6 & 5 & 5 & 2 \\ 5 & 6 & 6 & 6 & 2 \\ 6 & 7 & 5 & 5 & 3 \\ 3 & 5 & 2 & 4 & 4 \end{bmatrix}$$

Gambar 2. 6 Citra *(x,y)* berukuran 5x5 dan sebuah kernel dengan 3x3 matriks

Secara umum metode *Single Shot Detector* (SSD) mempunyai sebuah rumus sederhana dalam menentukan *default boxes* dan *scale default boxes*, dimana **N** merupakan jumlah *default boxes*, **Lconf** = *loss classification*, **Lloc** = *loss localization*, **L** = *prediction box* dan **g** = *truth ground box*. untuk menentukan *default boxes* bisa di lihat pada Gambar 2. 7 berikut:

$$L(x,\,c,\,l,\,g) = \frac{1}{N}\left(L_{conf}(x,c) + \alpha L_{loc}(x.\,l,g)\right)$$

Gambar 2. 7 *Default Box*

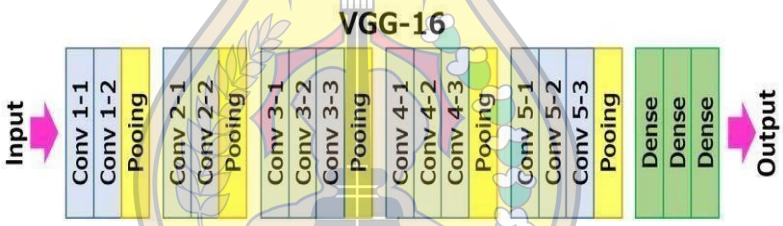
Sedangkan untuk menentukan *scale default boxes* bisa dilihat pada Rumus pada Gambar 2. 8 dimana **smin** adalah lapisan skala terendah, **smax** lapisan skala tertinggi dan **Sk** adalah *input pixels*:

$$S_k = S_{\min} + \frac{S_{max} - S_{min}}{m-1}(k-1), kc[1,m]$$

Gambar 2. 8 Rumus Scale Default Boxes

2.7 Arsitektur VGG16

Jaringan saraf *convolutional* juga dikenal sebagai ConvNet, yang merupakan sejenis jaringan saraf tiruan. Jaringan saraf *convolutional* memiliki lapisan *input*, lapisan *output*, dan berbagai lapisan tersembunyi. VGG16 adalah jenis CNN (*Convolutional Neural Network*) yang dianggap sebagai salah satu model visi komputer terbaik hingga saat ini. Pembuat model ini mengevaluasi jaringan dan meningkatkan kedalaman menggunakan arsitektur dengan filter konvolusi yang sangat kecil (3 × 3), VGG16 yang menunjukkan peningkatan signifikan pada konfigurasi sebelumnya. VGG16 mendorong kedalaman hingga 16-19 lapisan berat sehingga kira-kira 138 parameter yang dapat dilatih.



Gambar 2. 9 Lapisan Arsitektur VGG16

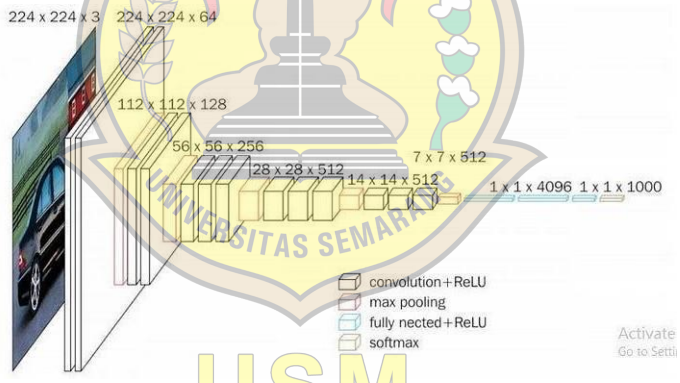
Seperti pada Gambar 2. 9, VGG16 adalah salah satu VGGnet model yang menggunakan 16 layer sebagai model arsitekturnya. VGG16 normalnya menggunakan 5 *convolutional blocks* yang kemudian terhubung ke 3 *MLP classifiers*. *Layer output* menggunakan *sigmoid activation function* apabila terdapat 2 atau kurang kategori, dan *softmax activation function* apabila terdapat 3 atau lebih kategori pada *dataset* (Hridayami et al., 2019)

VGG16 adalah algoritma deteksi dan klasifikasi objek yang mampu mengklasifikasikan 1000 gambar dari 1000 kategori berbeda dengan akurasi 92,7%. Ini adalah salah satu algoritma populer untuk klasifikasi gambar dan mudah digunakan dengan pembelajaran transfer. Alasan VGG-16 digunakan sebagai jaringan dasar adalah karena kinerjanya yang kuat dalam tugas-tugas klasifikasi gambar berkualitas tinggi dan popularitasnya untuk masalah-masalah di mana transfer membantu dalam meningkatkan hasil. Alih-alih VGG asli sepenuhnya terhubung lapisan, satu set lapisan konvolusional tambahan (dari conv6 dan

seterusnya) ditambahkan, sehingga memungkinkan untuk mengekstrak fitur pada berbagai skala dan secara progresif mengurangi ukuran *input* ke setiap lapisan berikutnya.

Angka 16 dalam VGG16 mengacu pada 16 lapisan yang memiliki bobot. Di VGG16 ada tiga belas lapisan konvolusional, lima lapisan *Max Pooling*, dan tiga lapisan *Dense* yang berjumlah hingga 21 lapisan tetapi hanya memiliki enam belas lapisan berat yaitu, lapisan parameter yang dapat dipelajari.

VGG16 mengambil ukuran tensor input sebagai 224, 244 dengan 3 saluran RGB. Hal yang paling unik tentang VGG16 adalah bahwa alih-alih memiliki sejumlah besar *hyper-parameter*, mereka berfokus untuk memiliki lapisan konvolusi filter 3x3 dengan *stride* 1 dan selalu menggunakan lapisan *padding* dan *maxpool* yang sama dari filter 2x2 stride 2.



Gambar 2. 10 Filter pada Arsitektur VGG16

Lapisan konvolusi dan kumpulan maks secara konsisten diatur di seluruh arsitektur. Seperti pada Gambar 2. 10, Conv-1 Layer memiliki 64 jumlah filter, Conv-2 memiliki 128 filter, Conv-3 memiliki 256 filter, Conv 4 dan Conv 5 memiliki 512 filter.

Tiga lapisan *Fully-Connected* (FC) mengikuti tumpukan lapisan konvolusional: dua yang pertama masing-masing memiliki 4096 saluran, yang ketiga melakukan klasifikasi ILSVRC 1000 arah dan dengan demikian berisi 1000 saluran (satu untuk setiap kelas).

Lapisan terakhir adalah lapisan *soft-max*. Arsitektur VGGNet menggabungkan fitur jaringan saraf konvolusi yang paling penting. Jaringan VGG terdiri dari filter konvolusi kecil. VGG16 memiliki tiga lapisan yang terhubung sepenuhnya dan 13 lapisan konvolusional. VGG-16 adalah jaringan saraf konvolusional dengan kedalaman 16 lapisan. Jaringan yang telah dilatih sebelumnya dapat mengklasifikasikan gambar ke dalam 1000 kategori objek, seperti keyboard, mouse, pensil, kendaraan dan banyak hewan.

UNIVERSITAS SEMARANG
USM

BAB III

ANALISA DAN PERANCANGAN

Metode penelitian yang digunakan pada tugas akhir ini melalui beberapa tahap. Dengan menggunakan *dataset* yang diperoleh dari penelitian sebelumnya oleh Aria Hendrawan, ST, M.Kom. , data diambil dari rekaman video pada kamera keamanan di jalan Perintis Kemerdekaan, Banyumanik, Semarang.

3.1 Perangkat Keras dan Perangkat Lunak

Sistem operasi yang digunakan untuk penelitian ini adalah LENOVO ideapad330, Core i5 Gen8, dan memory 12Gb. Sedangkan perangkat lunak yang digunakan dalam penelitian ini adalah sebagai berikut:

1. *Browser* (Chrome)
2. *GoogleColaboratory* (GPU T4, RAM 13GB)
3. *Python* 3
4. *Opencv-python*
5. *Matplotlib*

3.2 Tahap Penelitian

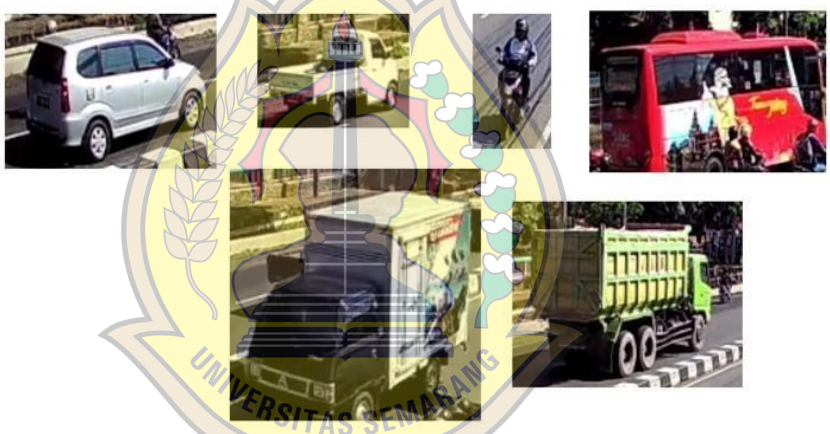
Tahap penelitian untuk objek deteksi menggunakan metode *Single Shot Detector* (SSD) dengan arsitektur VGG16 sebagai berikut:

3.2.1 Input *Dataset*

Dataset dalam penelitian ini mengakusisi dari penelitian "*Improvement Object Detection Algorithm Based on YoloV5 with BottleneckCSP*" yang dilakukan oleh Bapak Aria Hendrawan, Rahmat Gernowo, Oky Dwi Nurhayati, Budi Warsito, dan Adi Wibowo pada tahun 2022. Dalam penelitian tersebut, terdapat 1.104 (Seribu Seratus Empat) gambar dengan ukuran frame 480x480 piksel. *Dataset* telah dibagi menjadi tiga kelompok

(*testing*, *training*, *validation*) dengan pembagian 80% data *training*, 15% data validasi, dan 5% data *testing*. Serta telah diolah menggunakan *Mosaic data augmentation*, yang merupakan Teknik *Machine Learning* yang digunakan untuk meningkatkan ukuran dan keragaman kumpulan data dengan menggabungkan empat gambar menjadi satu.

Adapun *dataset* yang digunakan berbentuk COCO dengan format .xml dan terdiri dari 6 jenis kendaraan yang meliputi: Mobil/*Car*, Truk/*Truck*, *Bus*, Sepeda Motor/*Motorcycle*, Mobil Pengangkut/*Pickup Car*, Truk Bok/*Truck Box* (Hendrawan dkk., 2022) seperti pada Gambar 3. 1 berikut:



Gambar 3. 1 Jenis Kendaraan pada *Dataset*

3.2.2 Pra-Pemrosesan Data

Hal ini meliputi *resizing* gambar agar memiliki ukuran yang seragam, normalisasi nilai pixel, dan lain-lain sesuai dengan kebutuhan model VGG16.

3.2.3 Pelatihan Model VGG16

Pelatihan model VGG16 untuk tugas klasifikasi objek menggunakan *dataset* yang sudah dipersiapkan untuk melatih model. Proses pelatihan ini melibatkan pemberian *input* gambar beserta label objek, dan model akan belajar mengenali kelas objek tersebut.

3.2.4 Integrasi SSD

Setelah model VGG16 terlatih, selanjutnya integrasi metode SSD ke dalam arsitektur VGG16. SSD dirancang untuk melakukan deteksi objek secara *real-time* dengan memprediksi *bounding box* dan kelas objek pada berbagai skala.

3.2.5 Pelatihan Model SSD-VGG16

Setelah integrasi, selanjutnya pelatihan model SSD-VGG16 dengan menggunakan *dataset* yang telah dipersiapkan dengan anotasi *bounding box*. Proses ini melibatkan pembelajaran model untuk memprediksi *bounding box* dan kelas objek dalam satu tahap (*single shot*).

3.2.6 Evaluasi dan *Fine-Tuning*

Setelah model SSD-VGG16 terlatih, lakukan evaluasi kinerja model dengan menggunakan metrik deteksi objek *Mean Average Precision* (mAP).

3.2.7 Output Penelitian

Dari hasil *Preprocessing* deteksi objek kendaraan menggunakan metode *Single Shot Detector* (SSD) , langkah selanjutnya ialah mencoba menggunakan model yang dihasilkan untuk mendeteksi objek dalam bentuk gambar. Pada tahap ini akan menampilkan hasil deteksi objek 6 jenis kendaraan (Mobil/*Car*, Truk/*Truck*, *Bus*, Sepeda Motor/*Motorcycle*, Mobil Pengangkut/*Pickup Car*, Truk Bok/*Truck Box*).

3.2.8 Evaluasi

Dalam penelitian ini, ada proses evaluasi yang dilakukan guna mendapat skor performa dari setiap model yang ada. Proses evaluasi yang digunakan oleh penelitian ini merupakan *mean Average Precision* atau biasa disebut mAP, yang merupakan salah satu metrik yang digunakan untuk mengevaluasi performa dari model deteksi objek (Liu dkk., 2020). Metrik ini berguna untuk mengukur seberapa baik sebuah model dengan algoritma

tertentu dapat mengidentifikasi objek pada suatu gambar, dengan cara menghitung *precision* (rasio jumlah objek yang terdeteksi secara benar) dan *recall* (rasio objek sebenarnya yang terdeteksi).

Skor mAP ini didapat setelah menghitung rata-rata skor *precision* pada tingkat recall tertentu, dimana jika skor mAP yang didapat tinggi maka mengindikasikan bahwa model deteksi objek yang ada memiliki performa yangbagus. Berikut ini merupakan rumus dari *precision* dan *recall*:

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

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

Dimana TP merupakan *True Positive*, FP adalah *False Positive*, dan FN adalah *False Negative* (Developers, 2022). Selanjutnya jika skor *precision* dan *recall* sudah didapatkan, maka akan dilakukan perhitungan *Average Precision* (AP) dan terakhir mAP. Rumusnya adalah sebagai berikut:

$$AP = \int_{r=0}^{1} p(r)dr$$

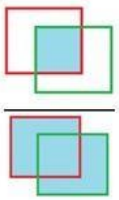
$$mAP = \frac{1}{N} \times \sum_{i}^{N} APi$$

Dimana $p(r)$ pada rumus AP merupakan *precision* pada tingkat *recall* tertentu r, r adalah *recall*, dan simbol integral mewakili integrasi matematika pada rentang nilai tertentu, dalam hal ini integrasi dilakukan dari $r = 0$ hingga

r = 1, yang berarti perlu mengintegrasikan nilai-nilai *precision* pada seluruh rentang *recall* dari 0 hingga 1. Sedangkan pada rumus mAP, N merupakan jumlah dari kategori objek, dan AP_i merupakan AP atau *Average Precision* untuk kategori i (kategori tertentu) (Anwar, 2022).

Intersection of Union atau IoU merupakan salah satu metrik yang digunakan untuk mengevaluasi performa model deteksi objek. IoU (*Intersection over Union*) atau *Jaccard Index* adalah metrik yang digunakan untuk mengukur sejauh mana prediksi *bounding box* model tumpang tindih dengan *bounding box ground truth* (anotasi). IoU digunakan sebagai acuan untuk proses evaluasi mAP , menyesuaikan skor IoU dan dilakukan deteksi objek pada kumpulan data yang belum pernah dilihat oleh model saat proses pelatihan, yang nantinya mAP yang dihitung adalah mAP dari hasil proses pendeteksian objek dengan skor IoU tersebut. IoU menghasilkan nilai antara 0 hingga 1, di mana nilai 1 menunjukkan bahwa prediksi *bounding box* sepenuhnya tumpang tindih dengan *ground truth*, sedangkan nilai 0 menunjukkan tidak ada tumpang tindih sama sekali. Metrik tersebut bisa dilihat pada Gambar 3. 2 berikut :

$$IOU = \frac{area\ irisan}{area\ gabunngan} =$$



Gambar 3. 2 *IoU = Intersection Area / Union Area*

Prediksi yang memiliki IoU lebih besar dari suatu *threshold* (misalnya, 0,5) sebagai deteksi yang benar (*true positive*) untuk menghitung metrik evaluasi seperti *Precision*, *Recall*, atau *F1 Score*. *Threshold* ini biasanya disesuaikan dengan kebutuhan spesifik tugas dan kasus penggunaan.

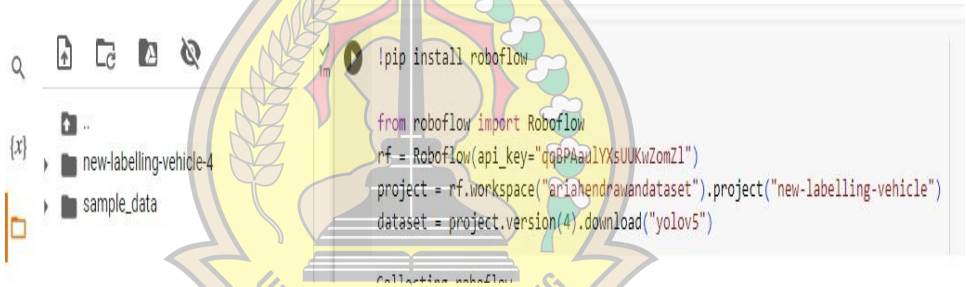
BAB IV

HASIL PEMBAHASAN

4.1 Implementasi

4.1.1 Input *Dataset*

Pertama akan dilakukan proses instalasi objek yang sebelumnya sudah di beri label pada website *roboflow*, terdapat 1014 *image* yang telah di beri label dan di jadikan *dataset*. Berikut baris kode program untuk melakukan instalasi *dataset* yang ditunjukkan pada Gambar 4. 1.

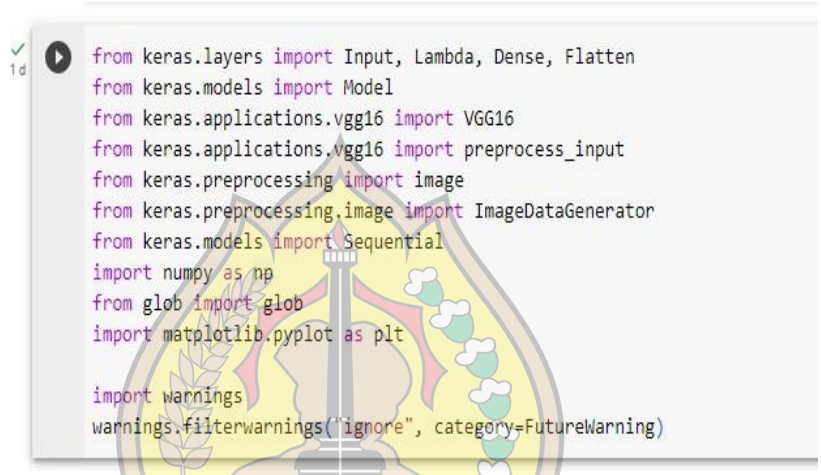


Gambar 4. 1 Proses instalasi *dataset*

Dalam konteks pengembangan model deteksi objek dengan menggunakan *platform Roboflow*, kode tersebut mengilustrasikan langkah-langkah yang perlu diambil. Pertama, langkah instalasi pustaka "*roboflow*" dilakukan dengan perintah !pip install roboflow. Selanjutnya, melalui penggunaan kelas *Roboflow*, kunci API digunakan untuk mengautentikasi akun pengguna dan mengakses proyek dan *dataset*. Dengan merinci workspace "ariahendrawan*dataset*" dan proyek "*new-labelling-vehicle*", kode tersebut mengunduh versi ke-4 *dataset* yang relevan dengan model deteksi objek YOLOv5. Ini adalah langkah-langkah penting dalam menyusun dasar *dataset* untuk keperluan pelatihan model deteksi objek pada *platform Roboflow*. Selanjutnya *dataset* yang telah terinstall namanya di ubah menjadi "*dataset*" agar lebih mudah dalam melakukan pemanggilan program.

4.1.2 Pra Pemodelan

Pada tahap pra pemodelan dilakukan beberapa penyesuaian ukuran (*size*) agar memliki ukuran yang seragam, menormalisasi nilai *pixel*, dan lain-lain disesuaikan dengan kebutuhan model VGG16 yang di gunakan.



Gambar 4. 2 Kode Program *Import* Pustaka Keras

Kode Program pada Gambar 4. 2 merupakan tahap pertama dalam Pra Pemodelan *dataset* yaitu mengimpor berbagai komponen yang diperlukan dari pustaka Keras, termasuk lapisan- lapisan jaringan saraf, model-model, alat-alat pra-pemrosesan gambar dari arsitektur VGG16, serta modul lain seperti pengaturan generator data, manipulasi data numerik, dan visualisasi dengan tujuan membangun, melatih, dan menguji model klasifikasi gambar. Dengan mengabaikan pesan peringatan kategori *Future Warning*.

```
#Give dataset path
train_path = '/content/dataset/train'
test_path = '/content/dataset/test'
```

```
[12] IMAGE_SIZE = [224, 224]
```

```
[13] from PIL import Image
     import os
     from IPython.display import display
     from IPython.display import Image as _Imgdis
     # creating a object


     folder = train_path+'/images'


     onlybenignfiles = [f for f in os.listdir(folder) if os.path.isfile(os.path.join(folder, f))]
     print("Working with {0} images".format(len(onlybenignfiles)))
     print("Image examples: ")


     for i in range(10):
         print(onlybenignfiles[i])
         display(_Imgdis(filename=folder + "/" + onlybenignfiles[i], width=240, height=240))
```

Gambar 4. 3 Kode Program Penampilan Contoh Gambar Pada *dataset train*

Tahap selanjutnya adalah penampilan contoh gambar pada *dataset train*. Fungsi dari kode program pada Gambar 4. 3 adalah untuk membaca dan menampilkan contoh-contoh gambar dari direktori *dataset* pelatihan. Kode ini juga mencetak jumlah total gambar yang sedang diolah untuk memberikan informasi tentang ukuran *dataset* yang digunakan.

Gambar 4. 4 Hasil Penampilan Contoh gambar pada *dataset train*

Pada Gambar 4. 4 memunculkan hasil yang ditampilkan oleh kode tersebut berupa Jumlah total gambar yang sedang diolah dalam *dataset*

pelatihan. Judul "*Image examples*: ". 10 contoh gambar pertama dari direktori *dataset* pelatihan, ditampilkan secara interaktif dengan dimensi tampilan gambar setinggi 240 piksel dan lebar 240 piksel. Gambar-gambar ini dapat dilihat dalam lingkungan *notebook* atau tempat di mana kode tersebut dijalankan.

Gambar 4. 5 Klasifikasi Multikeras

Dalam klasifikasi multikeras yang di tunjukan pada Gambar 4. 5 terdapat 5 baris kode. Dimana pada baris kode pertama sebuah model arsitektur VGG16 dibangun menggunakan pustaka Keras. Dimensi *input* gambar ditentukan dengan IMAGE_SIZE dan saluran warna 3 (RGB). Berat yang telah di-*pretrained* pada *dataset ImageNet* digunakan sebagai parameter awal model, dan lapisan penuh (*fully connected*) di bagian atas model tidak disertakan dengan argumen *include_top=False.*

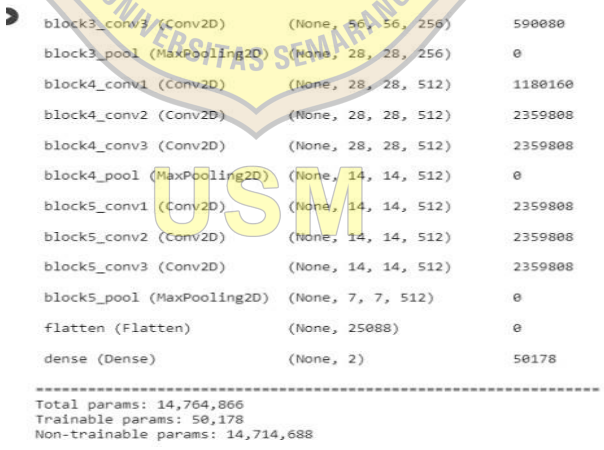
Pada baris kode kedua dilakukan Pemanggilan *vgg.input* menghasilkan objek *Input Layer* yang merupakan *input* dari model VGG16 yang telah dibuat sebelumnya. Objek ini merepresentasikan titik masukan untuk data gambar dalam model. Dalam konteks kode yang diberikan, ini mengacu pada *layer input* dari model VGG16 yang akan digunakan untuk menerima data gambar untuk proses inferensi atau pelatihan lebih lanjut.

Pada baris kode ke tiga penggunaan *Loop* ini digunakan untuk membekukan (*freeze*) bobot dari setiap lapisan dalam model VGG16.

Selama *loop*, atribut *trainable* dari masing-masing lapisan diatur menjadi *False*, sehingga memastikan bahwa bobot lapisan tersebut tidak akan diperbarui selama pelatihan lebih lanjut. Hal ini berguna untuk menjaga integritas bobot yang telah di-*pretrained* pada *dataset ImageNet*, sehingga model dapat digunakan sebagai ekstraktor fitur tanpa mengubah bobot yang ada.

Pada baris kode ke empat digunakan untuk menghitung dan mencetak jumlah sub direktori (folder) yang terdapat di dalam direktori */content/dataset/train/* dalam *dataset* pelatihan.

Pada baris kode kelima merupakan kode yang digunakan untuk mengkonstruksi sebuah model *neural network* yang menggunakan arsitektur VGG16 sebagai ekstraktor. Model ini kemudian dilengkapi dengan lapisan penuh (*fully connected*) untuk tujuan klasifikasi multikelas.



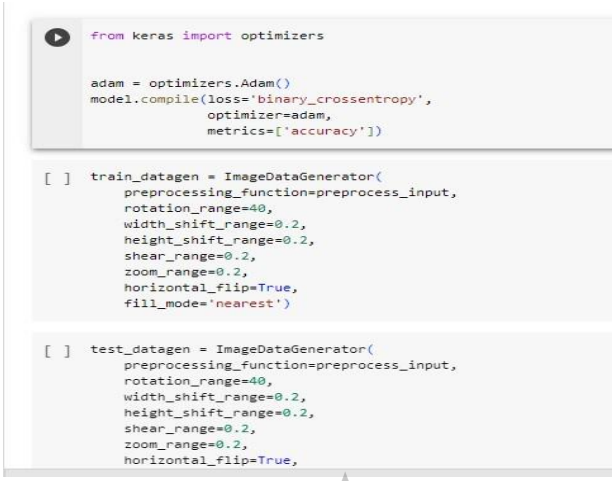
Gambar 4. 6 Ringkasan Arsitektur Model

Hasil akhirnya adalah model yang dapat menerima *input* gambar seperti pada Gambar 4. 6, menghasilkan prediksi berdasarkan jumlah sub direktori yang ada dalam direktori pelatihan *dataset*, dan ringkasan arsitektur model ini ditampilkan dengan *model.summary()*.

4.1.3 Pelatihan Model VGG16

Pelatihan model VGG16 melibatkan serangkaian langkah yang penting untuk menghasilkan model yang mampu mendeteksi objek dengan akurasi tinggi. Langkah awal adalah mempersiapkan *dataset* yang sesuai, mengumpulkan gambar-gambar yang mencakup berbagai kelas objek dan label *bounding box* yang relevan. Setelah *dataset* siap, langkah selanjutnya adalah mendefinisikan arsitektur model VGG16 dengan lapisan deteksi SSD yang sesuai untuk tugas deteksi objek. Pemilihan fungsi *loss* juga penting, biasanya berupa kombinasi *loss* klasifikasi dan *loss* regresi *bounding box*.

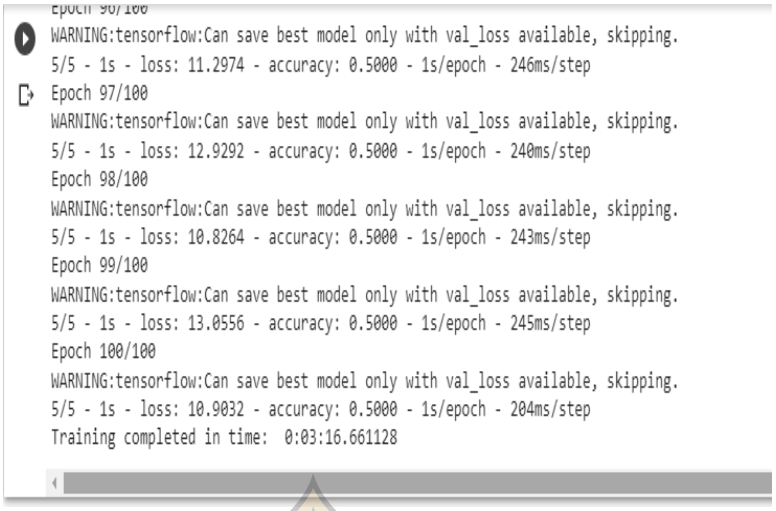
Proses pelatihan melibatkan inisialisasi model dengan *optimizer* dan parameter pelatihan. Selama beberapa *epoch*, gambar-gambar pelatihan dikirim melalui model, *loss* dihitung, dan parameter model diperbarui melalui *backpropagation*. Validasi dilakukan pada setiap *epoch* untuk memantau performa model dan memilih model terbaik. Setelah pelatihan selesai, model diuji pada *dataset* pengujian yang belum pernah dilihat sebelumnya, dan metrik evaluasi seperti akurasi, presisi, *recall*, dan *F1- score* dihitung untuk mengevaluasi kinerja model. Keseluruhan proses pelatihan ini memerlukan pemahaman mendalam tentang arsitektur model, optimisasi parameter, serta pemilihan metrik evaluasi untuk mencapai hasil yang optimal. Berikut adalah kode program Pelatihan Model VGG16:



Gambar 4. 7 Kode Program Pelatihan Mode VGG16

Pada Gambar 4. 7 dapat di jelaskan bahwa dalam skenario pelatihan model VGG16 untuk klasifikasi gambar menggunakan *dataset*, pertama-tama kita melakukan persiapan data dengan mengubah gambar-gambar ke dalam bentuk *tensor* dan melakukan normalisasi. Kemudian, *dataset* ini dipecah menjadi *batch-batch* kecil menggunakan *data loader*. Model VGG16 yang telah ada di dalam pustaka *PyTorch* digunakan sebagai arsitektur dasar, dan kita mengaktifkannya untuk melakukan klasifikasi dengan 6 kelas yang ada dalam *dataset*. Fungsi *Cross-Entropy Loss* digunakan untuk menghitung seberapa dekat prediksi model dengan label yang sebenarnya. Proses optimisasi dilakukan menggunakan algoritma *Stochastic Gradient Descent* (SGD), dimana parameter-parameter model diperbarui berdasarkan *gradien loss*. Selama beberapa *epoch*, model ini memperbarui parameter-parameter tersebut dan mengurangi *loss* yang dihasilkan.

Hasilnya adalah model yang telah dilatih secara iteratif untuk mengenali dan mengklasifikasikan objek-objek pada gambar dengan akurasi yang semakin meningkat. Hasil pelatihan model dalam *google colab* dapat dilihat pada Gambar 4. 8 berikut:



Gambar 4. 8 Hasil Pelatihan Model VGG-16 dengan 100 *epoch*

Sedangkan untuk nilai *Accuracy*, *Precision*, *Recall*, dan *F1-Score* dapat dilihat pada Gambar 4. 9 berikut:

Gambar 4. 9 *Accuracy*, *Precision*, *Recall*, *dan F1-Score*

4.1.4 Pelatihan Model SSD-VGG16

Dalam konteks pelatihan model SSD dengan arsitektur VGG16 untuk proyek deteksi objek, skrip tersebut terdiri dari komponen-komponen penting yang mendefinisikan aliran kerja pelatihan. Pertama, melalui pengimporan modul seperti *torch* dan *torchvision*, kerangka kerja *PyTorch* disiapkan. Transformasi data, termasuk konversi gambar menjadi *tensor* dan

normalisasi, memberikan prapemrosesan esensial sebelum digunakan dalam pelatihan. Penggunaan *Data Loader* memungkinkan *dataset* pelatihan untuk dimuat dalam *batch* kecil, meningkatkan efisiensi pelatihan.

Kode program pelatihan model SSD-VGG 16 dapat dilihat pada Gambar 4. 10 berikut:

```
transform_train = transforms.Compose([
    transforms.ToTensor(),
])

train_dataset = YourDataset(root='path/to/train_data', transform=transform_train)
train_loader = DataLoader(train_dataset, batch_size=8, shuffle=True)

model = VGG16_SSD(num_classes=your_num_classes, num_anchors=your_num_anchors)

criterion = Loss()
optimizer = optim.SGD(model.parameters(), lr=0.001, momentum=0.9)

for epoch in range(your_num_epochs):
    running_loss = 0.0
    model.train()
    for images, targets in train_loader:
        optimizer.zero_grad()
        outputs = model(images)
        loss = criterion(outputs, targets)
        loss.backward()
        optimizer.step()

        running_loss += loss.item()

    print(f'Epoch [{epoch + 1}/{your_num_epochs}], Loss: {running_loss / len(train_loader):.4f}')
```

Gambar 4. 10 Kode Program Pelatihan Model SSD-VGG 16

Langkah selanjutnya adalah inisialisasi model SSD yang telah sebelumnya didefinisikan. Fungsi *loss* yang ditetapkan menyediakan ukuran perbedaan antara hasil prediksi model dengan target sebenarnya. Pemilihan *optimizer*, seperti SGD atau Adam, membimbing proses optimasi dengan memperbarui parameter model berdasarkan nilai *loss* yang dihitung.

Dalam *loop* pelatihan, model disetel untuk mode pelatihan, dan data dalam setiap *batch* dari *dataset* pelatihan diproses. Proses propagasi maju dan mundur dilakukan untuk menghitung *loss* dan menghitung *gradien loss* terhadap parameter model. Setelahnya, parameter model disesuaikan melalui *optimizer*.

Seusai tiap *epoch*, informasi seperti nomor *epoch* dan *loss* rata-rata dicetak sebagai indikator kemajuan. Saat semua *epoch* selesai dilakukan, skrip memberikan pesan "Pelatihan selesai". Secara keseluruhan, skrip

tersebut memberikan kerangka kerja yang mendasari pelatihan model SSD dengan arsitektur VGG16 dalam proyek deteksi objek, yang dapat diadaptasi dan diperluas sesuai dengan kebutuhan dan tujuan spesifik tugas akhir ini.

4.1.5 Evaluasi dan *Fine Tunning*

```
transform_eval = transforms.Compose([
    transforms.ToTensor(),
    transforms.Normalize((0.485, 0.456, 0.406), (0.229, 0.224, 0.225))
])

eval_dataset = YourDataset(root='path/to/eval_data', transform=transform_eval)
eval_loader = DataLoader(eval_dataset, batch_size=8, shuffle=False)

model = VGG16_SSD(num_classes=your_num_classes, num_anchors=your_num_anchors)
model.load_state_dict(torch.load('path/to/pretrained_model.pth'))
model.eval()

correct = 0
total = 0
with torch.no_grad():
    for images, targets in eval_loader:
        outputs = model(images)

model.train()
optimizer = optim.SGD(model.parameters(), lr=0.001, momentum=0.9)

for epoch in range(your_num_epochs):85
    running_loss = 0.0
    for images, targets in train_loader:
        optimizer.zero_grad()

        outputs = model(images)
        loss = criterion(outputs, targets)
        loss.backward()
        optimizer.step()
        running_loss += loss.item()
    print(f'Epoch [{epoch + 1}/{your_num_epochs}], Loss: {running_loss / len(train_loader):.4f}')

print('Fine-tuning selesai')
```

Gambar 4. 11 *Script* Evaluasi dan *Fine Tuning*

Skrip pada Gambar 4. 11 berperan dalam melakukan evaluasi dan *fine-tuning* terhadap model SSD dengan arsitektur VGG16. Pada tahap evaluasi, gambar-gambar dari *dataset* evaluasi diubah menggunakan transformasi data yang telah ditentukan, kemudian dimuat dalam *batch-batch* melalui *DataLoader*. Model SSD yang telah terlatih sebelumnya dimuat dan dinilai kinerjanya pada *dataset* evaluasi. Dengan mengaplikasikan model pada gambar-gambar, prediksi deteksi objek diperoleh, yang akan menjadi dasar untuk menghitung metrik evaluasi.

Setelah evaluasi, tahap *fine-tuning* dimulai. Dalam proses ini, model ditempatkan kembali dalam mode pelatihan. Selama beberapa *epoch*, data pelatihan diproses secara berulang melalui model. Propagasi maju dan mundur digunakan untuk menghitung *loss* dan *gradien loss*, yang selanjutnya

digunakan untuk mengoptimalkan parameter-parameter model melalui *optimizer* SGD. Hasilnya, *loss* rata-rata dihitung dan dicetak setiap *epoch*. Setelah seluruh *epoch* selesai, skrip memberikan konfirmasi bahwa proses *fine-tuning* telah selesai.

Menggunakan skrip ini, model SSD dengan arsitektur VGG16 dianalisis kinerjanya pada *dataset* evaluasi, dan kemudian dioptimalkan lebih lanjut pada *dataset* pelatihan melalui proses *fine-tuning*. Ini merupakan langkah penting dalam pengembangan dan peningkatan model deteksi objek yang tepat dan akurat.

UNIVERSITAS SEMARANG
USM

4.2 Hasil dan Pembahasan

Tabel 4. 1 Hasil Evaluasi dengan IoU 0,75

Metric	*Presision*	*Recall*	*MaP*
ResNet50	*0,87*	*0,92*	*0,81*
ResNet50V2	*0,88*	*0.95*	*0.83*
MobileNetV30 Large	*0,90*	*0.95*	*0.86*
MobileNetV3 Large 320	*0,95*	*1.0*	*0.95*

Tabel 4. 2 Hasil Evaluasi dengan IoU 0,90

Metric	*Presision*	*Recall*	*MaP*
ResNet50	*0,93*	*0,92*	*0,85*
ResNet50V2	*0.96*	*0.95*	*0.91*
MobileNetV30 Large	*0,90*	*0.97*	*0.88*
MobileNetV3 Large 320	*0,95*	*0.90*	*0.86*

Berdasarkan hasil uji yang dilakukan pada Tabel 4. 1 dan Tabel 4. 2 dapat di simpulkan bahwa tingkat akurasi dari *dataset* yang di uji dengan IoU 0,75 dan IoU 0,9 pendeteksian dan penghitungan 1014 *dataset* pada pengujian *testing* mendapatkan nilai yang cukup tinggi pada uji IoU 0,75 dengan nilai *recall* rata-rata 92% dan data IoU 0,9 nilai *recall* 90%. Dari uji IoU 0,75 presisi mendapatkan nilai rata-rata 90% dan data *multiple* dengan nilai 92%, dan terakhir Uji MaP mendapatkan nilai 81% dan data *multiple* akurasi dengan nilai 85%. Kemudian

hasil dari deteksi objek ditampilkan pada Gambar 4. 12 Hasil DeteksiGambar 4. 12 berikut:

Gambar 4. 12 Hasil Deteksi

Ada beberapa faktor yang mempengaruhi skor deteksi objek antara lain: kualitas data, arsitektur model, kuantitas data pelatihan, kualitas label, parameter dan *hyperparameter*, *preprocessing* data, dan kapasitas komputasi. Semua faktor ini berinteraksi satu sama lain dan berkontribusi pada hasil deteksi objek.

Dalam konteks kumpulan data pelatihan yang digunakan dalam penelitian ini, ada beberapa aspek yang berpotensi tinggi mempengaruhi skor deteksi. Pertama, ukuran objek (besar, sedang, kecil) dalam data pelatihan dapat berdampak signifikan pada skor prediksi. Selain itu, posisi objek yang tumpang tindih juga berpotensi memengaruhi skor prediksi. Ini menyoroti pentingnya mempertimbangkan variasi dan kompleksitas dalam data pelatihan untuk mencapai skor prediksi yang baik.

BAB V

PENUTUP

5.1 Kesimpulan

Dari hasil penelitian yang telah penulis lakukan, penulis mengambil beberapa kesimpulan pada penelitian ini, yaitu: Berdasarkan hasil uji yang sudah dilakukan dapat di simpulkan bahwa tingkat akurasi dari *dataset* yang di uji dengan IoU 0,75 dan IoU 0,9 pendeteksian dan penghitungan 1014 *dataset* pada pengujian *testing* mendapatkan nilai yang cukup tinggi pada uji IoU 0,75 dengan nilai *recall* rata-rata 92% dan data IoU 0,9 nilai *recall* 90%. Dari uji IoU 0,75 presisi mendapatkan nilai rata- rata 90% dan data *multiple* dengan nilai 92%, dan terakhir Uji MaP mendapatkan nilai 81% dan data *multiple* akurasi dengan nilai 85%.

5.2 Saran

1. Penulis mengharapkan saran perbaikan atau pengembangan oleh penulis seperti mengurangi jumlah *dataset* yang mengakibatkan terlalu lamanya proses *training*.
2. Menambahkan objek gambar lain dengan dilakukan secara *realtime* dan menggunakan metode yang lain seperti YOLO sebagai perbandingan pada hasil pendeteksian kendaraan menggunakan metode SSD Arsitektur Vgg16 dengan *training* model SSD *Mobilenet*.
3. Menambahkan *num_step* 100.000 agar tingkat akurasi yang didapat semakin baik dan tinggi.
4. Untuk mendapatkan performa yang stabil dalam berbagai situasi, sebaiknya digunakan *dataset* yang cukup serupa namun memiliki berbagai jenis posisi kamera.
5. Sebaiknya menggunakan data yang jelas dan konsisten, contohnya adalah objek yang ada pada data tidak terlalu kecil, dan juga tidak terlalu besar serta memiliki berbagai sudut pandang dari objek yang ada.

6. Sebaiknya menggunakan lebih banyak *dataset* pada proses pelatihan. Karena semakin banyak *dataset*, terutama jika *dataset* yang digunakan konsisten, maka algoritma akan belajar mengidentifikasi objek secara lebih baik.

UNIVERSITAS SEMARANG
USM

DAFTAR PUSTAKA

A. Hendrawan, R. Gernowo, O. Nurhayati et al. (2022). Improvement Object Detection Algorithm Based on YoloV5 with BottleneckCSP.

Assidhiqi, F. (2021). Pengembangan Sistem Deteksi Hunian Parkir Menggunakan Metode Convolutional Neural Network.

Felix, Wijaya, Jeffry, Sutra, Stephen Putra, Kosasih, Pyter Wahyu, Sirait, Pahala. (2020). Implementasi Convolutional Neural Network Untuk Identifikasi Jenis Tanaman Melalui Daun. *Jurnal SIFO Mikroskil, 21*(1).

Fuady, Samratul, Nehru, Nehru, Anggraeni, Gina. (2020). Deteksi Objek Menggunakan Metode Single Shot Multibox Detector Pada Alat Bantu Tongkat Tunanetra Berbasis Kamera. *Journal of Electrical Power Control and Automation (JEPCA), III*(2). doi:10.33087/jepca.v3i2.38

Kumar, Ashwani, Srivastava, Sonam. (2020). Object Detection System Based on Convolution Neural Networks Using Single Shot Multi-Box Detector. *171*(2019). doi:10.1016/j.procs.2020.04.283

Kusnantoro, Rohana, Tatang, Kusumaningrum, Dwi Sulistya. (2020). Implementasi Metode Tesseract OCR(Optical Character Recognition)untuk Deteksi Plat Nomor Kendaraan Pada Sistem Parkir. *Scientific Student Journal for Information, Technology and Science, III*.

Pang, Yanwei, Wang, Tiancai, Anwer, Rao Muhammad, Khan, Fahad Shahbaz, Shao, Ling. (2019). Efficient featurized image pyramid network for single shot detector. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*(c). doi:10.1109/CVPR.2019.00751

Qu, Zhong, Shang, Xue, Xia, Shu Fang, Yi, Tu Ming, Zhou, Dong Yang. (2022). A method of single-shot target detection with multi-scale feature fusion and

feature enhancement.

Sukusvieri, Andrianto. (2020). Implementasi Metode Single Shot Detector untuk Pengenalan Wajah.

Susilo, Agus. (2019). Implementasi Metode Ssd (Single Shot Multibox Detector) Untuk Mendeteksi Pelanggaran Jalur Busway Menggunakan Masukan Citra Digital Program Studi Teknik Elektro.

Wahyu, Ari Purno, Suhendri. (2019). Peningkatan Sistem Keamanan Parkir dengan Teknologi Artificial Intelligence Imaging. *JOINT*.

UNIVERSITAS SEMARANG
USM

YAYASAN ALUMNI UNIVERSITAS DIPONEGORO
UNIVERSITAS SEMARANG
FAKULTAS TEKNOLOGI INFORMASI DAN KOMUNIKASI
Sekretariat : Jl. Soekarno Hatta Tlogosari Semarang 50196 Telp. (024) 6702757 Fax. (024) 6702272
Web site : www.usm.ac.id E-mail : univ_smg@usm.ac.id

SURAT PENUNJUKAN PEMBIMBING

Nomor : 271 /USM.H5.FTIK/I/2023 24 MAR 2023
Lampiran. : Form Nilai
Hal : **Bimbingan Tugas Akhir**

Kepada
Yth. Bapak / Ibu Dosen Pembimbing Tugas Akhir
Aria Hendrawan, ST, M.Kom
Jurusan Teknologi Informasi
UNIVERSITAS SEMARANG
Di Semarang

Dengan hormat,
Untuk menempuh mata kuliah Tugas Akhir pada Program S1 Teknik Informatika, mohon kepada mahasiswa yang tersebut di bawah ini :

Nama	:	KHANIN TITIS WULANDARI
NIM	:	G.211.19.0070
Program Studi	:	S1 Teknik Informatika
Judul TA	:	Implementasi Object Detection Kendaraan menggunakan Metode SSD (Single Shot Detector)
Tahun Akademik	:	Semester Genap Tahun Akademik 2022/2023

Dapat diberikan bimbingan dalam pembuatan Tugas Akhir berupa konsultasi dan asistensi. Perlu kami sampaikan bahwa penyelesaian Tugas Akhir paling lama 1 tahun terhitung sejak dilakukan pembayaran Tugas Akhir. Apabila dalam jangka waktu tersebut belum selesai, maka harus mengurus Perpanjangan Tugas Akhir dengan judul dan pembimbing yang ditetapkan ulang oleh Koordinator Tugas Akhir. Perpanjangan dilakukan paling banyak 2 (dua) kali periode.

Demikian untuk menjadikan periksa, atas bimbingan dan kerjasamanya diucapkan terimakasih.

Mengetahui,
a.n. Dekan
Wakil Dekan I

Fajriannoor Fanani, S.Sos., M.I.Kom.
NIS. 06557000606017

Ketua Program Studi
Teknik Informatika

Khairudin, S.Kom, M.Eng
NIS. 06557003102173

Tembusan :
1. Yth. Koordinator TA
2. Mahasiswa
3. Arsip

YAYASAN ALUMNI UNIVERSITAS DIPONEGORO
UNIVERSITAS SEMARANG
Sekretariat : Jl. Soekarno Hatta Tlogosari Semarang 50196 Telp.(024)6702757 Fax.(024)6702272

LEMBAR BIMBINGAN

Tugas Akhir

Nama Mahasiswa : KHANIN TITIS WULANDARI
N I M : G.211.19.0070
Judul : Implementasi Object Detection Kendaraan menggunakan Metode SSD (Single Shot Detector)

NO	TANGGAL	PEMBAHASAN	VALIDASI
1	26-03-2023	**Proposal** * Uraian Mahasiswa : Halaman Judul, Abstrak, Latar Belakang, Rumusan Masalah, Batasan Masalah, Tujuan Tugas Akhir, Manfaat Penelitian, Landasan Teori * Uraian Dosen Pembimbing : silakan dilihat d komen review Pak Aria	Revisi
2	13-05-2023	**Proposal** * Uraian Mahasiswa : Revisi Proposal * Uraian Dosen Pembimbing : silakan ditambahi jadwal pelaksanaan dan daftar pustaka	Revisi
3	08-06-2023	**Proposal** * Uraian Mahasiswa : Proposal * Uraian Dosen Pembimbing : silakan ditiru dan dikembangkan proposalnya sesuai kebutuhan penelitiannya menggunakan SSD	Revisi
4	01-08-2023	**Proposal** * Uraian Mahasiswa : Revisi Proposal * Uraian Dosen Pembimbing : Lanjut penelitiam dan laporannya	Acc
5	04-08-2023	**BAB III** * Uraian Mahasiswa : bab 1, bab 2 dan bab 3 * Uraian Dosen Pembimbing : silakan dilanjut penelitiannya dan dirapikan bab 3 ini	Acc
6	18-08-2023	**Laporan Lengkap** * Uraian Mahasiswa : Revisi BAB 3, BAB 4 dan BAB 5 * Uraian Dosen Pembimbing : hasil ssd algorithm nya ditampilkan, penelitiannya dilanjutkan dengan epoch 100	Revisi
7	21-08-2023	**Laporan Lengkap** * Uraian Mahasiswa : Laporan Lengkap * Uraian Dosen Pembimbing : silakan daftar sidang skripsi	Acc

Semarang, 5 September 2023.
Pembimbing,

ARIA HENDRAWAN, S.T., M.Kom
NIS. 06557003102159

YAYASAN ALUMNI UNIVERSITAS DIPONEGORO
UNIVERSITAS SEMARANG
Sekretariat : Jl. Soekarno Hatta Tlogosari Semarang 50196 Telp.(024)6702757 Fax.(024)6702272

LEMBAR PERSETUJUAN REVISI

Nama Mahasiswa : KHANIN TITIS WULANDARI
N I M : G.211.19.0070
Judul Skripsi : Implementasi Object Detection Kendaraan menggunakan Metode SSD (Single Shot Detector)
Tanggal Ujian : Jumat, 25 Agustus 2023
Materi Yang Direvisi : Cek catatan saya di laporan, revisi yang saya catatakan!
Redaksional : semua kata asing ketik miring!, semua gambar & tabel wajib dirujuk! Typo cek semua direvisi,

Telah direvisi oleh Mahasiswa yang bersangkutan dan telah disetujui oleh Tim Penguji :

KETUA TIM PENGUJI

Nama : SRI HANDAYANI, S.T., M.T.
Tanda Tangan : .....

PENGUJI PENDAMPING 1

Nama : ARIA HENDRAWAN, S.T., M.Kom
Tanda Tangan : .....

PENGUJI PENDAMPING 2

Nama : ASTRID NOVITA PUTRI, S. Kom., M. Kom.
Tanda Tangan : .....

YAYASAN ALUMNI UNIVERSITAS DIPONEGORO
UNIVERSITAS SEMARANG
Sekretariat : Jl. Soekarno Hatta Tlogosari Semarang 50196 Telp.(024)6702757 Fax.(024)6702272

LEMBAR PERSETUJUAN REVISI

Nama Mahasiswa : KHANIN TITIS WULANDARI
N I M : G.211.19.0070
Judul Skripsi : Implementasi Object Detection Kendaraan menggunakan Metode SSD (Single Shot Detector)
Tanggal Ujian : Jumat, 25 Agustus 2023
Materi Yang Direvisi : Bab IV Hasil ujuplikasi juga dijelaskan secara detail lengkap

Telah direvisi oleh Mahasiswa yang bersangkutan dan telah disetujui oleh Tim Penguji :

KETUA TIM PENGUJI

Nama : SRI HANDAYANI, S.T., M.T.
Tanda Tangan : .....

PENGUJI PENDAMPING 1

Nama : ARIA HENDRAWAN, S.T., M.Kom
Tanda Tangan : .....

PENGUJI PENDAMPING 2

Nama : ASTRID NOVITA PUTRI, S. Kom., M. Kom.
Tanda Tangan : .....

YAYASAN ALUMNI UNIVERSITAS DIPONEGORO
UNIVERSITAS SEMARANG
Sekretariat : Jl. Soekarno Hatta Tlogosari Semarang 50196 Telp.(024)6702757 Fax.(024)6702272

LEMBAR PERSETUJUAN REVISI

Nama Mahasiswa : KHANIN TITIS WULANDARI
N I M : G.211.19.0070
Judul Skripsi : Implementasi Object Detection Kendaraan menggunakan Metode SSD (Single Shot Detector)
Tanggal Ujian : Jumat, 25 Agustus 2023
Materi Yang Direvisi : Revisi di Laporan

Telah direvisi oleh Mahasiswa yang bersangkutan dan telah disetujui oleh Tim Penguji :

KETUA TIM PENGUJI

Nama : SRI HANDAYANI, S.T., M.T.
Tanda Tangan : .....

PENGUJI PENDAMPING 1

Nama : ARIA HENDRAWAN, S.T., M.Kom
Tanda Tangan : .....

PENGUJI PENDAMPING 2

Nama : ASTRID NOVITA PUTRI, S. Kom., M. Kom.
Tanda Tangan : .....

USM

YAYASAN ALUMNI UNIVERSITAS DIPONEGORO
UNIVERSITAS SEMARANG
Sekretariat : Jl. Soekarno Hatta Tlogosari Semarang 50196 Telp.(024)6702757 Fax.(024)6702272

BERITA ACARA UJIAN TUGAS AKHIR

Pada hari ini Jumat, tanggal 25 Bulan Agustus Tahun 2023 jam 09.00 WIB telah dilaksanakan Ujian Tugas Akhir / Sarjana Program Studi S1 Teknik Informatika , Fakultas Teknologi Informasi Dan Komunikasi

Untuk dibacakan kepada peserta ujian

1. Apakah Anda dalam kondisi sehat ?
2. Apakah Anda dalam keadaan tanpa tekanan / paksaan ?
3. Apakah Anda bersedia menerima apapun keputusan para penguji ?

Nama / Nim	Judul Skripsi	Jawab	Tanda Tangan
KHANIN TITIS WULANDARI G.211.19.0070 Kelas : PAGI	Implementasi Object Detection Kendaraan menggunakan Metode SSD (Single Shot Detector)	1. Ya / ~~Tidak~~ 2. Ya / ~~Tidak~~ 3. Ya / ~~Tidak~~	

Dengan Hasil :

NO	NAMA PENGUJI	JABATAN	NILAI	TANDA TANGAN
1	SRI HANDAYANI, S.T., M.T.	Ketua Tim Penguji	78	
2	ARIA HENDRAWAN, S.T., M.Kom	Penguji Pendamping 1	78	
3	ASTRID NOVITA PUTRI, S. Kom., M. Kom.	Penguji Pendamping 2	75	

Setelah diadakan sidang, dengan ini para Dosen Penguji menetapkan nilai 77/B (Revisi / ~~tdk~~)
Demikian Berita Acara ini dibuat untuk dapat dipergunakan sebagaimana mestinya.

Semarang, 25 Agustus 2023
Ketua Tim Penguji,

SRI HANDAYANI, S.T., M.T.
NIS. 06557003102116

Dibuat Rangkap 3 :
1) Untuk Jurusan
2) Untuk Dosen Wali
3) Arsip

80 - keatas	: A
70 - 79	: B
60 - 69	: C
40 - 59	: D
40 kebawah	: E