



**SOCIAL DISTANCING DETEKTOR
DENGAN METODE
YOLOv4**

**Lutfiyah Nur Fadhilah
NRP. 3120640018**

DOSEN PEMBIMBING

**Dr. Tita Karlita, S.Kom., M.Kom.
NIP. 197910142002122002**

**M. Udin Harun Al Rasvid, Ph.D
NIP. 198108082005011001**

**PROGRAM STUDI SARJANA TERAPAN
TEKNIK INFORMATIKA
DEPARTEMEN TEKNIK INFORMATIKA DAN KOMPUTER
POLITEKNIK ELEKTRONIKA NEGERI SURABAYA
2021**



PROYEK AKHIR

SOCIAL DISTANCING DETEKTOR DENGAN METODE YOLOv4

**Lutfiyah Nur Fadhillah
NRP. 3120640018**

Dosen Pembimbing :

**Dr. Tita Karlita, S.Kom., M.Kom.
NIP. 197910142002122002**

**M. Udin Harun Al Rasyid, Ph.D
NIP.198108082005011001**

**PROGRAM STUDI SARJANA TERAPAN
TEKNIK INFORMATIKA
DEPARTEMEN TEKNIK INFORMATIKA DAN KOMPUTER
POLITEKNIK ELEKTRONIKA NEGERI SURABAYA
2021**

Social Distancing Detector

dengan Metode Yolov4

Oleh:

Lutfiyah Nur Fadhilah
NRP. 3120640018

Proyek Akhir ini Diajukan sebagai Salah Satu Syarat untuk
Memperoleh Gelar Sarjana Terapan Komputer (S.Tr.Kom.)
di

Program Studi D4 Teknik Informatika
Departemen Teknik Informatika dan Komputer
Politeknik Elektronika Negeri Surabaya

Disetujui dan disahkan pada tanggal 20 Agustus 2021 oleh:

Dosen Pembimbing:

1. Dr.. Tita Karlita, S.Kom, M.Kom
NIP. 197910142002122002
2. M. Udin Harun Al Rasyid, Ph.D
NIP. 198108082005011001

Dosen Penguji:

1. Dr. Arna Fariza, S.Kom., M.Kom.
NIP. 197107081999032001
2. Desy Intan Permatasari, S.Kom., M.Kom.
NIP. 198603232015042004
3. Yanuar Risah Prayogi, S.Kom., M.Kom.
NIP. 198801062019031009



Mengetahui,
Ketua Program Studi D4 Teknik Informatika

Ahmad Syauqi Ahsan, S.Kom., M.T.
NIP. 197505302003121001

KATA PENGANTAR

Syukur alhamdulillah atas kehadiran Allah SWT atas segala rahmat dan karunia-Nya sehingga penulis bisa menyelesaikan proyek akhir ini yang berjudul Social Distancing Detector dengan Metode YOLO v4.

Pembuatan dan penyusunan proyek akhir ini diajukan sebagai salah satu syarat untuk menyelesaikan studi Diploma-4 (D4) dan memperoleh gelar Sarjana Terapan Komputer (S.Tr.Kom) di Program Studi Teknik Informatika Departemen Teknik Komputer dan Teknik Informatika Politeknik Elektronika Negeri Surabaya. Proses penyelesaian proyek akhir ini berdasar pada teori-teori yang telah diperoleh dalam perkuliahan, study literature dan bimbingan dari dosen pembimbing.

Penyelesaian proyek akhir ini tidak akan terwujud tanpa adanya pihak lain. Oleh karena itu dalam kesempatan ini penulis menyampaikan ucapan terima kasih kepada pihak-pihak yang telah membantu dan membimbing pelaksanaan dan penyelesaian proyek akhir ini, yaitu kepada :

1. Allah SWT yang telah mencurahkan segenap kenikmatan baik berupa nikmat iman dan islam serta barokah, hidayah dan inayah nya kepada penulis.
2. Bapak M. Udin Harun Al Rasyid, S.Kom, Ph.D. selaku Kepala Departemen Teknik Informatika dan Komputer PENS dan juga Dosen Pembimbing.
3. Bapak Ahmad Syauqi Ahsan, S.Kom, M.T. selaku Ketua Prodi Sarjana Terapan Teknik Informatika PENS.
4. Keluarga tercinta, kedua orang tua dan adik yang telah memberikan do'a, kasih sayang, semangat, nasihat dan nafkah yang tak bisa penulis balas dengan apapun dan entah sampai kapanpun.
5. Ibu Dr. Tita Karlita S.Kom, M.Kom. selaku dosen pembimbing yang membimbing, menyemangati dan bersedia menerima segala macam keluh kesah penulis hingga Proyek Akhir ini selesai.
6. Teman – teman kelas D4 Lanjut Jenjang yang memberikan semangat dan kenangan selama menjalani perkuliahan ditengah pandemi.
7. Teman – teman angkatan 2017 khususnya Dio Dzaky dan Muhammad Irzam Kamil mahasiswa Jurusan lain yang telah mendukung dan memberikan bantuan kepada penulis selama pengerjaan Proyek Akhir.

8. PT Velacom Indonesia yang telah mendukung saya melanjutkan Pendidikan ditengah kewajiban saya untuk bekerja.
9. Komunitas Surabaya.py yang telah memberikan penulis kesempatan untuk belajar lebih dalam mengenai *machine learning* dan Bahasa pemograman Python yang kemudian menerapkannya pada Proyek Akhir ini sehingga mendapatkan hasil yang cukup memuaskan.

Segala ucapan terima kasih tentunya belum cukup, semoga Allah SWT senantiasa membalas segala kebaikan Anda semua. Aamiin yaa robbal'alamin. Penulis menyadari bahwasanya masih banyak kekurangan dan kelemahan dalam buku ini. Oleh karena itu kritik dan saran yang membangun serta koreksi yang konstruktif sangat diharapkan untuk perkembangan lebih lanjut. Semoga dengan adanya buku ini memberikan manfaat bagi pengembangan ilmu pengetahuan dan bagi semua pihak pada umumnya serta bagi penulis sendiri pada khususnya.

Surabaya, 31 Desember 2021

Lutfiyah Nur fadhilah

ABSTRAK

Coronavirus disease 2019 atau COVID-19 telah menginfeksi ratusan juta orang dan merenggut ribuan juta korban jiwa di seluruh dunia. Penyebaran virus COVID-19 ini juga dapat dikatakan cukup cepat sehingga berdampak pada tatanan kehidupan di masyarakat. Masyarakat harus tetap beraktifitas untuk memenuhi kebutuhan sehari-hari seperti makanan, perawatan kesehatan, dan pekerjaan yang diperlukan. Oleh karena itu Social Distancing Detector merupakan sebuah solusi yang diajukan dengan memberi tanda peringatan kepada masyarakat untuk tetap menjaga jarak aman antar individu sesuai anjuran yang telah ditentukan oleh World Health Organization (WHO) yaitu 1,83m ~6feet. Penelitian ini menggunakan YOLOv4 untuk melakukan deteksi objek, dalam hal ini yaitu objek manusia. Penggunaan YOLOv4 ini dikarenakan YOLOv4 dapat memproses pengolahan citra dengan cepat dan akurat dibandingkan versi YOLO sebelumnya. Penelitian ini terdiri dari lima tahapan, yaitu data preparation, hyperparameter tuning, model training, model evaluation, dan model testing. Proses training memanfaatkan teknik transfer learning dari pre-trained model convolutional neural network (CNN). Training menggunakan dataset Google OpenImage yang telah berlabel untuk menghasilkan model yang dapat dijadikan solusi penelitian. Ujicoba yang dilakukan menggunakan jumlah dataset sebanyak 2000 gambar, 1 kelas, 64 batch, dengan learning rate 0,001. Proses training ini dilakukan dengan jupyternotebook (MI Azure). Bounding box pada pemrograman ini digunakan untuk mendeteksi dan mengklasifikasikan object secara tepat dan dilanjutkan untuk perhitungan jarak objek. Persentase keberhasilan dari hasil pengujian deteksi objek ini didapatkan sebesar 93,8% dengan nilai error deteksi jarak citra dengan jarak nyata sebesar 21,6%.

Kata kunci

COVID-19, *Realtime, Social Distancing Detector, YouOnlyLookOnce-v4, Convolutional Neural Network (CNN)*.

ABSTRACT

Coronavirus disease 2019 or COVID-19 has infected hundreds of millions of people and claimed thousands of millions of lives worldwide. The spread of the COVID-19 virus can also be said to be quite fast so that it has an impact on the order of life in society. People must continue to do activities to meet their daily needs such as food, health care, and necessary work. Therefore, the Social Distancing Detector is a solution proposed by giving a warning sign to the public to maintain a safe distance between individuals according to the recommendations set by the World Health Organization (WHO) which is 1.83m ~ 6feet. This study uses YOLOv4 to detect objects, in this case human objects. The use of YOLOv4 is because YOLOv4 can process image processing quickly and accurately compared to the previous YOLO version. This research consists of five stages, namely data preparation, hyperparameter tuning, training model, evaluation model, and testing model. The training process utilizes transfer learning techniques from the pre-trained convolutional neural network (CNN) model. The training uses a labeled Google OpenImage dataset to produce a model that can be used as a research solution. The Experiment was carried out using a dataset of 2000 images, 1 class, 64 batches, with a learning rate of 0.001. This training process is carried out with jupyternotebook (ML Azure). Bounding boxes in this programming are used to detect and classify objects appropriately and continue to calculate object distances. The percentage of success from the object detection test results was obtained at 93.8% with an error value of image distance detection with a real distance of 21.6%.

Keyword

COVID-19, Realtime, Social Distancing Detector, YouOnlyLookOnce-v4, Convolutional Neural Network (CNN)

DAFTAR ISI

HALAMAN JUDUL	
HALAMAN PENGESAHAN	
KATA PENGANTAR.....	i
ABSTRAK	iii
ABSTRACT	iv
DAFTAR ISI.....	v
DAFTAR GAMBAR.....	viii
DAFTAR TABEL	ix
BAB 1 PENDAHULUAN	1
1.1. LATAR BELAKANG.....	1
1.2. PERMASALAHAN	2
1.3. TUJUAN	2
1.4. MANFAAT	3
1.5. SISTEMATIKA PENULISAN	3
BAB 2 KAJIAN PUSTAKA	5
2.1. DESKRIPSI PERMASALAHAN	5
2.2. TEORI PENUNJANG.....	6
2.2.1 Computer Vision	6
2.2.2 Convolutional Neural Network	7
2.2.3 YOLO v4	10
2.3. PENELITIAN TERKAIT	12
2.3.1 Narinder Singh Pun, Sanjay Kumar Sonbhadra dan Sonali Agarwa dalam jurnalnya yang berjudul <i>Monitoring COVID-19 social distancing with person detection and tracking via fine-tuned YOLO v3 and Deepsort technique</i> [3]	12
2.3.2 Mahdi Rezael dan MohsenAzarmi dalam jurnalnya yang berjudul <i>DeepSOCIAL: Social Distancing Monitoring and Infection Risk Assessment in COVID-19 Pandemic</i> . [4]	12
2.3.3 Imran Ahmed dan Misbah Ahmad dalam jurnalnya yang berjudul <i>A deep learning-based social distance monitoring framework for COVID-19</i> . [9]	13
2.3.4 Dongfang Yang, Ekim Yurtsever, Vishnu Renganathan, Keith A. Redmill, dan Umit Ozguner dalam jurnalnya yang berjudul <i>A Deep Convolutional Neural Network Model for Vehicle Recognition and Face Recognition</i> [8]	13

2.3.5.	Vinitha. V dan Velantina dalam jurnalnya yang berjudul <i>Social Distancing Detection System With Artificial Intelligence Using Computer Vision And Deep Learning</i> [6].....	14
BAB 3	DESKRIPSI SISTEM.....	15
3.1.	DESKRIPSI SOLUSI.....	15
3.2.	DESAIN SISTEM	16
3.2.1.	Training Dataset.....	17
3.2.2.	Hasil Training	18
3.2.3.	Implementasi.....	19
BAB 4	EKSPERIMEN DAN ANALISA	24
4.1.	PARAMETER EKSPERIMEN	24
4.2.	KARAKTERISTIK DATA	24
4.3.	SPESIFIKASI PERALATAN UJI COBA	25
4.4.	HASIL EKSPERIMEN	26
4.4.1.	Antarmuka sistem deteksi jarak aman	26
4.4.2.	Hasil Evaluasi Proses Training	27
4.4.3.	Lingkungan Terkontrol.....	29
4.4.2.1.	Pengujian Deteksi Manusia	29
4.4.2.2.	Pengujian Deteksi Pelanggaran Jarak.....	30
4.5.	ANALISIS HASIL EKSPERIMEN	31
4.5.1.	Analisis hasil evaluasi model.....	31
4.5.2.	Analisis hasil deteksi manusia	32
4.5.3.	Analisis hasil deteksi jarak.....	33
4.5.4.	Analisis keseluruhan sistem.....	33
BAB 5	PENUTUP.....	34
5.1.	KESIMPULAN	34
5.2.	SARAN	34
DAFTAR PUSTAKA	Error! Bookmark not defined.	
LAMPIRAN.....	Error! Bookmark not defined.	

DAFTAR GAMBAR

Gambar 2.1 Computer Vision.....	6
Gambar 2.2 Proses CNN	8
Gambar 2.3 Proses <i>Convolutional Layer</i>	9
Gambar 2.4 <i>Pooling layer</i>	9
Gambar 2.5 <i>Yolo One-Stage Detector</i>	11
Gambar 3.1. Alur kerja sistem.....	15
Gambar 3.2. Bagan Desain Sistem	16
Gambar 3.3. Bagian proses training dataset	17
Gambar 3.4. Tampilan Dataset	17
Gambar 3.5. Hasil Training	19
Gambar 3.6. Bagian tahap implementasi.....	19
Gambar 3.7. Ilustrasi deteksi objek	20
Gambar 3.8. Ilustrasi perhitungan jarak objek.....	21
Gambar 3.9. Ilustrasi pembagian area untuk deteksi jarak	22
Gambar 4.1. Dataset	25
Gambar 4.2. Antarmuka Social Distancing Detector	26
Gambar 4.3. Dataset yang digunakan	27
Gambar 4.4. Grafik total loss dan map	29
Gambar 4.5 Elapsed time saat deteksi	32

DAFTAR TABEL

Tabel 4.1. Spesifikasi kamera untuk uji coba	25
Tabel 4.2. Spesifikasi laptop untuk uji coba.....	26
Tabel 4.3. Hyperparameter	28
Tabel 4.4. Hasil pengujian deteksi objek.....	30
Tabel 4.5. Hasil pengujian deteksi jarak.....	31

BAB 1

PENDAHULUAN

1.1. LATAR BELAKANG

Seperti yang telah diketahui bersama bahwa Pandemi COVID-19 telah menyebabkan dampak yang luar biasa bagi masyarakat dunia. Virus ini disebabkan oleh SARS-CoV-2 yang dapat menginfeksi manusia dengan jangka waktu yang cepat dengan media udara. Berdasarkan data yang dihimpun oleh Our World Data, pada 18 Juli 2021 sejumlah 2,88 juta orang masyarakat Indonesia telah terjangkit Virus COVID-19 dimana 73.582 orang yang meninggal dunia disebabkan oleh virus ini. Pada Desember 2019, ketika COVID-19 muncul di Wuhan, Cina, Dalam satu bulan, wabah di Cina mencapai puncaknya pada minggu pertama bulan Februari dengan 2.000 hingga 4.000 kasus terkonfirmasi baru per hari. Kemudian, untuk pertama kalinya setelah wabah ini, ada tanda kelegaan tanpa kasus baru yang dikonfirmasi selama lima hari berturut-turut hingga 23 Maret 2020. Banyak negara maju, termasuk India dan Korea Selatan, memanfaatkan GPS untuk melacak pergerakan orang yang dicurigai atau terinfeksi untuk memantau kemungkinan terpapar di antara orang sehat. Di India, pemerintah menggunakan Aplikasi Arogya Setu, yang bekerja dengan bantuan GPS dan bluetooth untuk menemukan keberadaan pasien COVID-19 di sekitar wilayah tersebut. Di sisi lain, beberapa departemen penegakan hukum telah menggunakan drone dan kamera pengintai lainnya untuk mendeteksi pertemuan massal orang, dan mengambil tindakan regulasi untuk membubarkan massa. Social distancing dipilih sebagai tindakan yang dapat mencegah penyebaran. Hal ini membuktikan bahwa langkah-langkah social distancing yang diberlakukan di China pada awalnya, kemudian diadopsi di seluruh dunia digunakan untuk mengendalikan COVID-19. Pemerintah Indonesia juga telah berupaya keras untuk dapat mengendalikan virus ini mulai dengan program Pembatasan Sosial Berskala Besar (PSBB), Pemberlakuan Pembatasan Kegiatan Masyarakat (PPKM) hingga level darurat pada saat ini, dan program vaksinasi besar-besaran yang dilakukan oleh pemerintah pusat saat ini.

1.2. PERMASALAHAN

Selama pandemi COVID-19, pemerintah telah mencoba menerapkan berbagai praktik jarak sosial, seperti membatasi perjalanan antar kota, mengontrol perbatasan wilayah, menutup sekolah dan tempat hiburan, serta menyiagakan masyarakat untuk tetap menjaga jarak 1,6 m hingga 2 m dengan satu sama lain. Namun, memantau jumlah penyebaran infeksi dan efisiensi hambatan bukanlah tugas mudah. Masyarakat harus keluar dan memenuhi kebutuhan penting seperti makanan, perawatan kesehatan, dan pekerjaan yang diperlukan. Dalam situasi seperti itu, Kecerdasan buatan dapat memainkan peran penting dalam memfasilitasi pemantauan jarak sosial. Computer Vision, sebagai sub-bidang dari Kecerdasan Buatan yang telah sangat berhasil dalam memecahkan berbagai masalah perawatan kesehatan yang kompleks dan telah menunjukkan potensinya dalam pengenalan COVID-19 berbasis CT-Scan atau sinar-X dan juga dapat berkontribusi pada pemantauan jarak sosial. Selain itu, jaringan neural dalam memungkinkan untuk mengekstrak fitur kompleks dari data sehingga dapat memberikan pemahaman yang lebih akurat tentang gambar dengan menganalisis dan mengklasifikasikan fitur ini. Contohnya termasuk diagnosis, manajemen dan pengobatan klinis, serta pencegahan dan pengendalian COVID-19.

Oleh karena itu Social Distancing Detector merupakan sebuah solusi yang diajukan dengan memberi tanda peringatan kepada masyarakat untuk tetap menjaga jarak aman antar individu sesuai persyaratan yang telah ditentukan oleh World Health Organization (WHO) yaitu 6 kaki ~ 182,88cm[10].

1.3. TUJUAN

Penelitian proyek akhir ini dibuat untuk menerapkan Convolutional Neural Network (CNN) untuk membuat sebuah sistem pemantau atau *surveillance* yang mampu mendeteksi objek manusia melalui kamera pemantauan yang kemudian dianalisis dengan metode YouOnlyLookOnce v4. Objek manusia akan dipantau apakah telah memenuhi jarak aman sesuai syarat WHO. Saat sistem mendeteksi adanya pelanggaran *Social Distancing* maka akan terlihat jumlah manusia yang beresiko pada layar. Dengan pemantauan ini maka penyebaran COVID-19 diharapkan berkurang dengan penerapan disiplin protokol kesehatan *social-distancing*.

1.4. MANFAAT

Kontribusi utama penelitian ini dapat disoroti sebagai berikut :

1. Bagi Masyarakat

Penelitian ini bertujuan untuk mendukung pengurangan penyebaran virus corona dengan memberikan solusi berbasis AI untuk memantau dan mendeteksi pelanggaran jarak social antar individu secara otomatis.

2. Bagi Univeristas

Penelitian ini diharapkan dapat digunakan untuk menambah referensi sebagai bahan penelitian lanjutan yang lebih mendalam.

3. Bagi Peneliti

Penelitian ini diharapkan dapat memperdalam ilmu khususnya dalam bidang iinformatika.

1.5. SISTEMATIKA PENULISAN

Sistematika buku laporan proyek akhir ini terbagi dalam bab-bab yang akan dibahas sebagai berikut:

Bab 1 Pendahuluan

Bab pendahuluan berisi latar belakang lahirnya ide untuk membuat sebuah sistem untuk pelanggaran jarak social distancing sebagai proyek akhir penulis. Dari latar belakang tersebut penulis dapat menguraikan masalah yang terjadi, tujuan dari pembuatan proyek akhir, kontribusi dari proyek akhir apabila proyek akhir telah selesai dibuat, dan sistematika dari penulisan buku proyek akhir ini.

Bab 2 Kajian Pustaka

Bab ini membahas teori-teori dasar yang terkait dengan deteksi manusia dan pelanggaran jarak social distancing sehingga dapat menunjang pengerjaan Proyek Akhir ini. Selain teori-teori dasar, penulis juga menyertakan penelitian-penelitian terdahulu yang berhubungan dengan proyek akhir ini untuk dapat membuktikan keunikan dari proyek akhir ini.

Bab 3 Desain Sistem

Bab ini menjelaskan rancangan sistem, perancangan pengambilan data serta pengelolaan data.

Bab 4 Eksperimen dan Analisis

Bab ini meliputi parameter eksperimen, karakteristik data, scenario ujicoba, tempat dan waktu eksperimen, spesifikasi peralatan ujicoba, cara penafsiran dan penyimpulan hasil proyek akhir.

Bab 5 Kesimpulan dan Saran

Bab yang berisi kesimpulan dari proyek tugas akhir, saran untuk pengembangan, perbaikan, maupun penyempurnaan pada aplikasi yang telah dibuat.

BAB 2

KAJIAN PUSTAKA

2.1. DESKRIPSI PERMASALAHAN

Banyak peneliti di bidang medis dan farmasi meneliti pengobatan penyakit menular COVID-19, namun hingga saat ini belum ada solusi pasti yang ditemukan. Disisi lain, mengendalikan penyebaran virus di tempat umum adalah masalah yang dapat ditangani dengan kecerdasan buatan atau *Artificial Intelligence* (AI) yang salah satunya adalah dengan menggunakan *Computer Vision*.

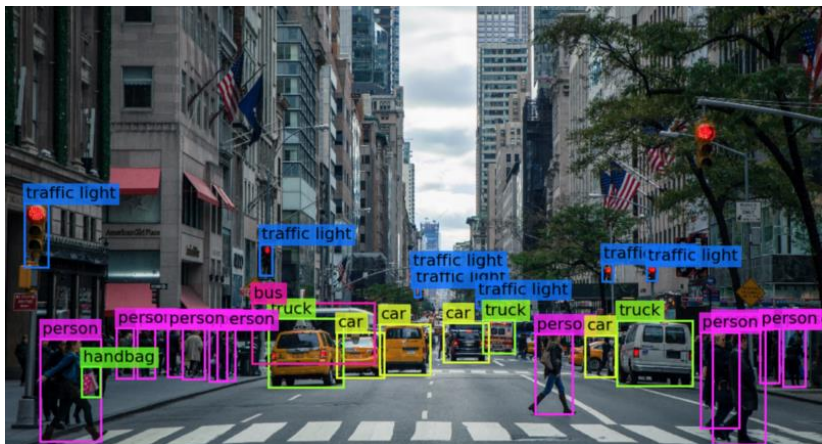
Berbagai studi telah dilakukan dengan menggunakan strategi implementasi yang berbeda telah membuktikan bahwa pengendalian prevalensi merupakan salah satu cara, dan jarak sosial merupakan cara yang efektif untuk mengurangi penularan dan mencegah penyebaran virus di masyarakat. Beberapa peneliti menggunakan model *Susceptible, Infectious*, atau *Recovered* (SIR). SIR adalah sistem pemodelan epidemiologi yang menghitung jumlah teoritis orang yang terinfeksi penyakit menular dalam populasi tertentu dari waktu ke waktu. Baru baru ini Eksin et al. memperkenalkan model baru SIR yang dimodifikasi dengan memasukkan parameter jarak sosial, yang dapat digunakan untuk menentukan jumlah individu yang terinfeksi dan pulih. Di sisi lain, kemajuan terbaru dalam *computer vision, deep learning*, dan pengenalan pola sebagai sub kategori AI memungkinkan computer untuk memahami dan menafsirkan data visual dari gambar atau video digital. Ini juga memungkinkan computer untuk mengidentifikasi dan mengklasifikasikan berbagai jenis objek. Kemampuan tersebut dapat memainkan peran penting dalam memberdayakan , mendorong, dan melakukan pengawasan dan pengukuran jarak sosial juga, misalnya computer vision dapat mengubah kamera CCTV dalam kapasitas infrastruktur saat ini menjadi kamera “pintar” yang tidak hanya memantau orang tetapi juga dapat menentukan apakah orang tersebut mengikuti pedoman jarak sosial atau tidak. Sistem semacam itu membutuhkan algoritma deteksi manusia yang sangat tepat.[3].

2.2. TEORI PENUNJANG

Dalam pengerjaan proyek akhir ini, terdapat teori-teori pendukung yang dapat diimplementasikan pada sistem deteksi pelanggaran jarak social distancing ini. Teori-teori pendukung tersebut akan diuraikan pada sub-bab berikut.

2.2.1 *Computer Vision*

Computer vision merupakan sebuah bidang dalam *artificial intelligent* yang bertujuan melatih sebuah komputer untuk memahami hal visual. Bidang *computer vision* memiliki fokus untuk mereplikasi bagian yang paling kompleks pada manusia yaitu sistem penglihatan manusia dan memungkinkan sebuah komputer dapat mengidentifikasi dan memproses objek pada gambar dan video sama seperti yang manusia lihat. Model *deep learning* biasa dimanfaatkan pada teknologi *computer vision* ini, sebuah mesin yang dimasuki model *deep learning* dapat secara akurat mengidentifikasi dan mengklasifikasikan objek pada sebuah gambar, kemudian bereaksi terhadap apa yang mesin itu lihat.



Gambar 2.1 Computer Vision

Sumber : Everything You Ever Wanted To Know About Computer Vision. by Ilija Mihajlovic Towards Data Science

Computer vision dapat dimanfaatkan ke dalam berbagai hal di berbagai bidang salah satunya dengan membuat sistem deteksi objek seperti pada

Gambar 2.1 dengan memanfaatkan perkembangan teknologi yang sangat pesat.

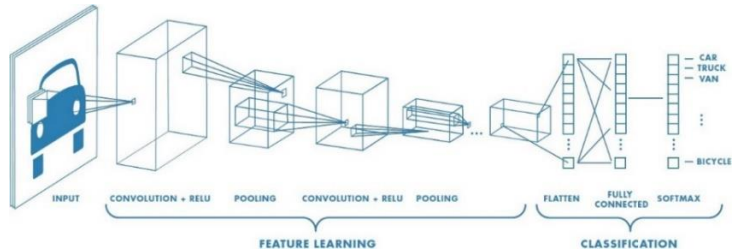
Computer Vision telah diperluas ke bidang yang luas mulai dari merekam data mentah ke dalam ekstraksi pola citra dan interpretasi informasi . computer vision memiliki kombinasi konsep, teknik, dan ide dari pengolahan citra digital, pengenalan pola, kecerdasan buatan dan komputer grafis . Sebagian besar tugas dalam computer vision terkait dengan proses memperoleh informasi tentang peristiwa atau deskripsi, dari adegan input (gambar digital) dan ekstraksi fitur. Metode yang digunakan untuk memecahkan masalah dalam computer vision bergantung pada domain aplikasi dan sifat data yang dianalisis.

Computer vision adalah kombinasi dari pemrosesan gambar dan pengenalan pola. Hasil dari proses Computer Vision adalah pemahaman gambar. Pengembangan bidang ini dilakukan dengan mengadaptasi kemampuan penglihatan manusia dalam mengambil informasi. Computer Vision adalah disiplin ilmu mengekstraksi informasi dari gambar, berbeda dengan Grafik Komputer . Pengembangan dari visi komputer tergantung pada sistem teknologi komputer, apakah tentang kualitas gambar , perbaikan atau pengenalan citra. Ada tumpang tindih dengan Image Processing pada teknik dasar, dan beberapa penulis menggunakan kedua istilah tersebut secara bergantian. Tujuan utama Computer Vision adalah untuk membuat model dan ekstrak data dan informasi dari gambar, sedangkan Image Processing adalah tentang menerapkan transformasi komputasi untuk gambar, seperti penajaman, kontras, antara lain [4]. Itu juga memiliki arti yang sama dan terkadang tumpang tindih dengan In Human and Computer Interaction (HCI). Cakupan HCI fokus pada desain penuh, antarmuka dan semua aspek teknologi yang terkait dengan interaksi antara manusia dan komputer. HCI kemudian dikembangkan sebagai disiplin ilmu tersendiri (yang merupakan bidang ilmu interdisipliner) yang membahas keterkaitan antara manusia-komputer yang dimediasi oleh perkembangan teknologi termasuk aspek manusia.

2.2.2. Convolutional Neural Network

Pada saat ini pendekatan yang paling populer dan sering digunakan dalam mendeteksi sebuah objek ialah menggunakan metode *convolutional neural network* (CNN) dimana pada metode ini dapat mempelajari sebuah objek pada *input* gambar. CNN adalah pengembangan dari *Multilayer*

Perceptron (MLP) yang didesain untuk mengolah data dua dimensi. CNN termasuk dalam jenis *Deep Neural Network* karena kedalaman jaringan yang tinggi dan banyak diaplikasikan pada data citra. Pada kasus klasifikasi citra, MLP kurang sesuai untuk digunakan karena tidak menyimpan informasi spasial dari data citra dan menganggap setiap piksel adalah fitur yang independen sehingga menghasilkan hasil yang kurang baik[3].



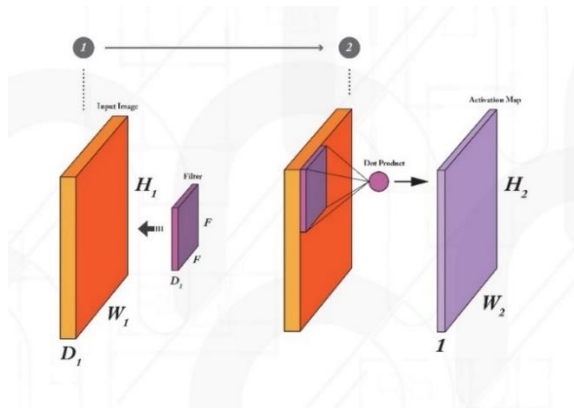
Gambar 2.2 Proses CNN

Sumber : <https://www.mathworks.com/solutions/deep-learning/convolutional-neural-network.html>

Gambar 2.2 merupakan Contoh jaringan dengan banyak convolutional layer. Filter diterapkan ke setiap gambar training pada resolusi yang berbeda, dan output dari setiap gambar yang dikonvolusi digunakan sebagai input ke lapisan berikutnya. CNN Secara umum algoritma CNN memiliki tiga lapisan atau *layer*, yaitu *convolutional layer*, *pooling layer* / *subsampling layer* dan *fully connected layer*.

a. Convolutional Layer

Pada *convolutional layer* dan *subsampling layer* dilakukan ekstraksi dari sebuah gambar menjadi *features* yang berupa angka-angka yang merepresentasikan image tersebut.



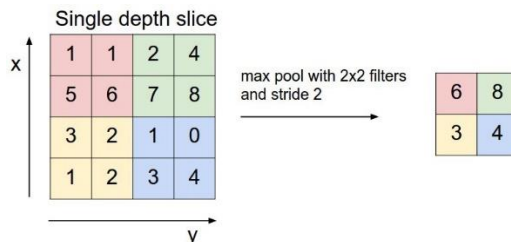
Gambar 2.3 Proses *Convolutional Layer*

Sumber : <https://medium.com/nodeflux/mengenal-convolutional-layer-dan-pooling-layer-3c6f5c393ab2>

Convolutional layer merupakan proses yang memanfaatkan sebuah *filter* yang diinisiasi dengan nilai tertentu. Seperti *input* gambar *filter* tersebut memiliki ukuran panjang dan tinggi. Seperti pada gambar 2.3 filter ini nantinya akan digeser keseluruhan bagian dari input gambar, setiap pergeserannya dilakukan operasi “dot” antara nilai *input* gambar dan filter tersebut. Hasil dari proses operasi “dot” dari setiap pergeseran input gambar dan filter akan menghasilkan *activation map*.

b. *Pooling Layer / Subsampling Layer*

Secara khusus *subsampling layer* merupakan proses mengurangi atau mereduksi ukuran pada sebuah input gambar.



Gambar 2.4 *Pooling layer*

Sumber : <https://medium.com/@samuelsena/pengenalan-deep-learning-part-7-convolutional-neural-network-cnn-b003b477dc94>

Dalam layer ini metode yang biasa digunakan ialah *max pooling* dengan mengambil nilai terbesar pada area yang di *filter* seperti ditunjukkan pada Gambar 2.4.

c. *Fully Connected Layer*

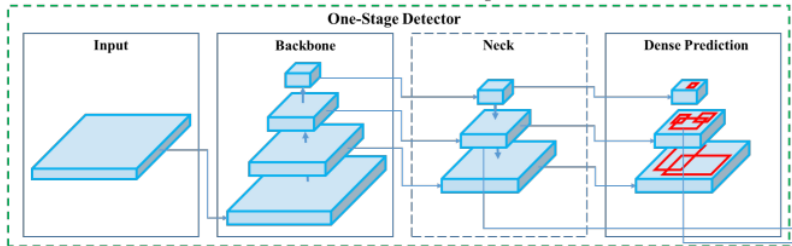
Activation map yang didapatkan dari *layer* sebelumnya masih berbentuk *array multidimensional*. Pada *layer* ini akan dilakukan transformasi pada dimensi *activation map* agar bisa diklasifikasikan secara linear. Setiap neuron pada *activation map* harus ditransformasi menjadi data satu dimensi agar bisa menjadi *input* pada *fully connected layer*. Akibatnya *input* gambar akan kehilangan informasi dan tidak *reversible*, oleh karena itu *fully connecter layer* diimplementasikan pada akhir jaringan.

2.2.3. YOLO v4

Pengenalan objek merupakan salah satu permasalahan dalam computer vision yang berhubungan dengan pengenalan suatu objek dalam citra dan video digital. Computer vision merupakan salah satu cabang dari kecerdasan buatan (artificial intelligence) yang difokuskan pada pengembangan algoritma untuk menganalisis informasi dari suatu citra ke dalam bentuk informasi yang sebenarnya di dunia nyata. Saat ini deep learning banyak digunakan pada penelitian mengenai computer vision. Terdapat dua pendekatan dalam pengenalan objek menggunakan deep learning yaitu pendekatan berdasarkan region proposal atau metode dua tahap dan pendekatan berdasarkan metode regresi atau metode satu tahap. Metode pengenalan objek yang menggunakan dua tahap antara lain region-based convolutional neural networks (R-CNN) , Spatial Pyramid Pooling Network (SPP-Net) , Fast R-CNN , and Faster R-CNN . Metode tersebut menggunakan dua tahap karena mengklasifikasikan ulang dengan menerapkan model ke citra di beberapa lokasi dan skala lalu memberi nilai pada citra sebagai bahan evaluasi untuk pengenalan objek. Metode pengenalan objek dalam satu tahap tanpa melakukan klasifikasi ulang sehingga menjadi lebih cepat yaitu You OnlyLookOnce . Seperti yang sebelumnya dijelaskan, YOLO (You Only Look Once) merupakan salah satu dari *pretrained model* dengan versi yang paling terbarunya adalah YOLO versi empat. YOLO merupakan model untuk menyelesaikan masalah *object detection*. Klasifikasi dan deteksi objek memiliki perbedaan, dimana klasifikasi hanya untuk mengklasifikasi atau menggolongkan objek yang berbeda sedangkan pada deteksi objek juga

mengklasifikasi suatu objek dan memberitahukan dimana objek itu berada pada sebuah gambar.

Terdapat dua tipe dari model deteksi objek yaitu, model *one stage* dan *two stage*. model *two stage* memiliki akurasi yang sangat baik namun sangat lambat, sedangkan model *one stage* bukan berarti akurasi yang dimiliki tidak baik, namun tak sebaik model *two stage* namun memiliki kecepatan mendeteksi objek jauh lebih cepat, YOLOv4 merupakan model *object detection one stage*[7].



Gambar 2.5 Yolo One-Stage Detector

Sumber : YOLOv4: Optimal Speed And Accuracy of Object Detection

Secara umum pada arsitektur deteksi objek memiliki beberapa bagian yaitu *backbone*, *neck*, dan *head* seperti yang terlihat pada Gambar 2.5. Berbeda dengan pendahulunya, YOLOv4 memiliki banyak perkembangan terutama pada arsitektur *backbone* dan *neck* yang memiliki nilai mAP 10% lebih baik serta *frame rate* yang dihasilkan 12% lebih cepat, hal ini menyebabkan YOLOv4 lebih ringan proses trainingnya. *Backbone* merupakan *convolution layer* utama, pada YOLOv4 arsitektur *backbone* terdapat metode atau *plugin* tambahan untuk meningkatkan akurasi yaitu, BoF (*Bag of Freebies*), BoS (*Bag of Specials*). Algoritma YOLO merupakan algoritma deep learning untuk deteksi objek yang menggunakan pendekatan berbeda dari algoritma lain, yaitu menerapkan sebuah jaringan syaraf tunggal pada keseluruhan citra. YOLO mendeteksi sebuah objek dalam beberapa tahap yaitu

1. Membagi citra dalam region/grid berukuran $s \times s$. Grid-grid tersebut bertanggung jawab untuk mendeteksi objek. Pada tiap grid juga akan diprediksi bounding box beserta nilai confidence.
2. Tiap bounding box memiliki 5 nilai informasi yaitu x, y, w, h dan c . Nilai x dan y adalah koordinat titik tengah bounding box yang terprediksi, nilai w dan h adalah rasio ukuran lebar dan tinggi

relatif terhadap grid, dan c adalah nilai confidence bounding box tersebut.

3. Pada algoritma YOLO, tiap grid akan memprediksi nilai class probabilitas jika diprediksi terdapat objek di dalamnya. Saat pengujian, YOLO akan mengkalikan nilai class probability dengan nilai confidence dari bounding box. Sehingga menghasilkan nilai confidence kelas secara spesifik pada tiap bounding box. Nilai ini menunjukkan class probability yang muncul pada bounding box dan seberapa akurat bounding box memprediksi sesuai dengan objek.

2.3. PENELITIAN TERKAIT

Sehubungan dengan proyek akhir yang dilakukan penulis, maka sangatlah penting untuk melakukan tinjauan studi terkait penelitian terdahulu dengan topik yang serupa yang nantinya mendukung pengerjaan proyek akhir ini. Berikut beberapa ulasan tentang penelitian terdahulu yang pernah dilakukan sebelumnya, antara lain:

2.3.1. Narinder Singh Pun, Sanjay Kumar Sonbhadra dan Sonali Agarwa dalam jurnalnya yang berjudul *Monitoring COVID-19 social distancing with person detection and tracking via fine-tuned YOLO v3 and Deepsort technique*[3]

Penelitian mengenai deteksi kendaraan dan estimasi jarak ini dilakukan oleh Jong Bae Kim pada tahun 2019[13]. Penelitian ini menggunakan metode Aggregated Channel Features (ACFs) dan Inverse Perspective Mapping untuk mengubah gambar input dua dimensi menjadi tiga dimensi. Dengan menghasilkan gambar yang diproyeksi dalam tiga dimensi, kendaraan otonom dan jarak antar kendaraan pun terdeteksi. Dengan metode tersebut menunjukkan hasil akurasi sebesar 87,5 persen untuk mendeteksi kendaraan dan 92,8 persen untuk estimasi jarak di waktu yang nyata.

2.3.2. Mahdi Rezael dan MohsenAzarmi dalam jurnalnya yang berjudul *DeepSOCIAL: Social Distancing Monitoring and Infection Risk Assessment in COVID-19 Pandemic*.[4]

Mahdi Rezael dan MohsenAzarmi dalam jurnalnya yang berjudul *DeepSOCIAL: Social Distancing Monitoring and Infection Risk Assessment in COVID-19 Pandemic* yang diterbitkan pada tanggal 26 Oktober 2020 telah

melakukan penelitian mengenai *social distancing detector* dengan mengusulkan tiga tahap termasuk mendeteksi orang, pelacakan, dan estimasi antar jarak sebagai solusi total untuk pemantauan jarak social dan analisis resiko infeksi berbasis zona interaksi. [4].

2.3.3. Imran Ahmed dan Misbah Ahmad dalam jurnalnya yang berjudul *A deep learning-based social distance monitoring framework for COVID-19*. [9]

Imran Ahmed dan Misbah Ahmad dalam jurnalnya yang berjudul *A deep learning-based social distance monitoring framework for COVID-19* yang diterbitkan pada tanggal November 2021 telah melakukan penelitian mengenai *social distancing detector* dengan mengusulkan YOLOv3 digunakan untuk pendeteksian manusia. Sebagai penampilan seseorang, visibilitas, skala, ukuran, bentuk, dan pose sangat bervariasi dari karya ini adalah upaya pertama yang memanfaatkan transfer learning untuk model deteksi berbasis deeplearning. [9].

2.3.4. Dongfang Yang, Ekim Yurtsever, Vishnu Renganathan, Keith A. Redmill, dan Umit Ozguner dalam jurnalnya yang berjudul *A Deep Convolutional Neural Network Model for Vehicle Recognition and Face Recognition* [8]

Dongfang Yang, Ekim Yurtsever, Vishnu Renganathan, Keith A. Redmill, dan Umit Ozguner dalam jurnalnya yang berjudul *A Vision-based Social Distancing and Critical Density Detection System for COVID-19 Pandemic* yang diterbitkan pada tanggal 8 Juli 2020 telah melakukan penelitian mengenai *social distancing detector* dengan menggunakan kamera fixed monocular untuk mendeteksi individu di *Region of Interest* (ROI) dan mengukur jarak interpersonal secara real time tanpa perekaman data. Usulan ini berupa pengiriman signal audio-visual non-intrusif untuk peringatan kepada publik jika pelanggaran jarak sosial terdeteksi. Selain itu, kami mendefinisikan critical *social density metric* baru dan mengusulkan tidak memasuki *Region Of Interest* (ROI) jika kepadatan lebih tinggi dari nilai critical social density metric. [8]

2.3.5. Vinitha. V dan Velantina dalam jurnalnya yang berjudul *Social Distancing Detection System With Artificial Intelligence Using Computer Vision And Deep Learning* [6]

Vinitha. V dan Velantina dalam jurnalnya yang berjudul *Social Distancing Detection System With Artificial Intelligence Using Computer Vision And Deep Learning* yang diterbitkan pada tanggal 8 Agustus 2020 telah melakukan penelitian mengenai pengembangan pendeteksi social distancing baru yang mungkin dapat berkontribusi pada kesehatan publik. Model ini mengusulkan kerangka kerja berbasis *real-time deep learning* yang efisien untuk mengotomatiskan proses pemantauan jarak sosial melalui deteksi objek dan pendekatan pelacakan, di mana setiap individu diidentifikasi secara realtime dengan bantuan kotak pembatas (*bounding box*). Jumlah pelanggaran dikonfirmasi dengan menghitung jumlah orang yang terbentuk dan istilah indeks pelanggaran dihitung sebagai rasio jumlah orang terhadap jumlah cluster. Uji coba ekstensif dilakukan dengan model deteksi objek canggih yang sedang populer yaitu faster RCNN, SSD, dan YOLO v3, karena pendekatan ini sangat sensitif terhadap lokasi spasial kamera, pendekatan yang sama dapat disesuaikan untuk penyesuaian yang lebih baik dengan bidang pandang yang sesuai. [6]

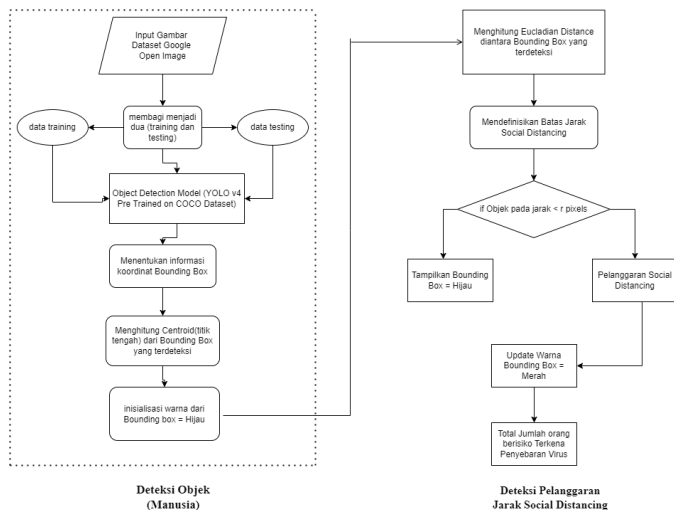
BAB 3

DESKRIPSI SISTEM

Pada bab ini menjelaskan solusi yang digunakan untuk menyelesaikan permasalahan-permasalahan yang telah diuraikan pada bab sebelumnya. Pada bab ini juga mendeskripsikan mengenai desain sistem yang dibangun pada proyek akhir ini.

3.1. DESKRIPSI SOLUSI

Proyek ini mengusulkan sistem yang menggunakan metode *deep learning* yang efisien dan berbasis kerangka kerja untuk mengotomatiskan proses pemantauan jarak sosial melalui deteksi objek dan pendekatan pelacakan, Dimana setiap individu manusia diidentifikasi dengan bantuan *bounding box*. Bantuan *bounding box* yang dihasilkan dalam mengidentifikasi kelompok orang dihitung dengan bantuan *closeness property*. Jumlah pelanggaran dikonfirmasi dengan menghitung jumlah orang yang terdeteksi memiliki *closeness property*. Adapun alur kerja sistem deteksi pelanggaran jarak pada proyek akhir ini ditunjukkan pada **Gambar 3.1.**

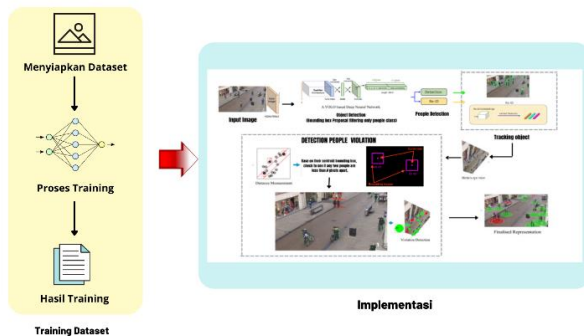


Gambar 3.1. Alur kerja sistem

Pada **Gambar 3.1**, sistem berjalan dengan memproses image yang telah diinput, image tersebut akan dibagi kedalam region atau grid yang nantinya akan diproses probabilitas class yang diprediksi sesuai algoritma yolo. Kemudian setelah objek terdeteksi kedalam class yang memiliki probabilitas tertinggi, maka objek tersebut, dalam hal ini manusia, akan memiliki bounding box atau kotak bujur sangkar dengan id masing masing yang nantinya berfungsi untuk melakukan tracking. Sistem dilanjutkan dengan pendeteksian jarak, dimulai dengan citra yang dideteksi hanya fokus pada Region of Interest(ROI) atau garis lurus untuk penentuan jarak, hal ini dilakukan untuk menyesuaikan posisi citra saat pengambilan gambar dengan arah pandang bird eye view. Pada proses ini, setiap koordinat centroid dari bounding box akan diproses untuk menghitung jarak antar objek dnegan menggunakan rumus Euclidean distance. Nilai pixel matrix dari perhitungan Euclidean distance akan dibandingkan dengan r atau nilai batas pelanggaran jarak. Jika memang jarak tersebut kurang dari r maka bounding box akan berubah menjadi berwarna merah yang mengindikasikan pelanggaran jarak.

3.2. DESAIN SISTEM

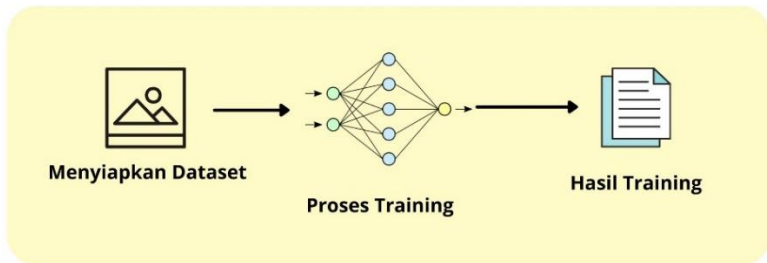
Perancangan sistem yang dilakukan pada proyek akhir ini ditunjukkan pada **Gambar 3.2**. Penjelasan mengenai desain sistem akan diuraikan pada sub-bab berikutnya.



Gambar 3.2. Bagan Desain Sistem

3.2.1. Training Dataset

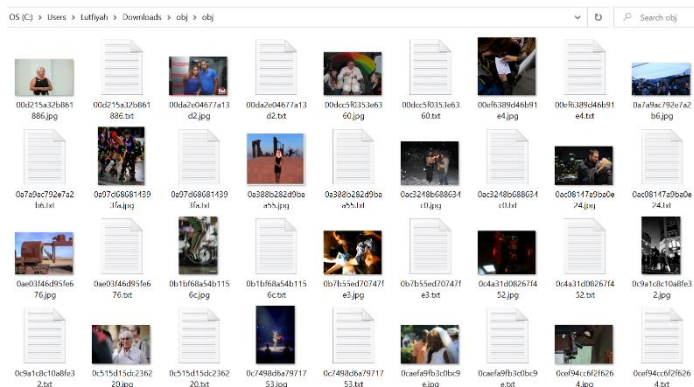
Training data merupakan proses melatih dataset sehingga sistem dapat mengenali objek pada gambar. Berikut adalah rincian proses dari tahap *training* data yang ditunjukkan pada **Gambar 3.3**.



Gambar 3.3. Bagian proses training dataset

Menyiapkan Dataset

Dataset yang digunakan adalah dataset Google open image. Dataset di download melalui website Open Images Dataset V6. Dataset yang tersedia pada OpenImage telah memiliki label sehingga penulis tidak perlu melakukan proses anotasi mandiri. Proses download dilakukan dengan memilih terlebih dahulu class apa saja yang akan di download. Pada proses ini penulis hanya mendownload gambar dataset class Human Body dengan jumlah data training 1000 gambar dan data testing 1000 gambar. Gambar 3.4



Gambar 3.4. Tampilan Dataset Google Open Image merupakan tampilan data yang diunduh dari Google OpenImage Dataset.





Proses Training

Pada tahap ini, hasil dari tahap menganotasi gambar kemudian dilatih menggunakan metode *transfer learning*. *Transfer learning* merupakan suatu teknik atau metode yang memanfaatkan model yang sudah dilatih terhadap suatu dataset untuk menyelesaikan permasalahan lain yang serupa dengan cara menggunakannya sebagai starting point, memodifikasi dan memperbarui parameternya sehingga sesuai dengan dataset yang baru[7]. Model yang telah dilatih sebelumnya disebut dengan *pre-trained network*. *Pre-trained network* tersebut akan diadaptasi dengan dataset baru, proses adaptasi ini disebut dengan *fine tuning*. AlexeyAB menyediakan berbagai *pre-trained network* salah satunya adalah YOLOv4 yang digunakan pada tahap ini. Sebelumnya, pretrained Yolo V4 tersebut telah dilatih menggunakan COCO Dataset. Pretrained YOLOv4 berperan sebagai base model yang berfungsi untuk mengatur pendeteksian objek dengan membuat *bounding box* dan menentukan posisi objek, sedangkan YOLOv4 berperan sebagai *network model* yang akan bekerja untuk mengekstraksi fitur yang akan diklasifikasikan ke dalam beberapa kelas, sehingga keduanya saling melengkapi untuk pembuatan aplikasi deteksi objek. YouOnlyLookOnce(YOLO) merupakan model yang biasanya digunakan untuk deteksi objek yang akan diimplementasikan pada *Social Distancing Detector*. Selain Pretrained network, file yang dibutuhkan adalah YOLOv4-obj.cfg , obj.names, dan obj.data. pada YOLOv4-obj.cfg penulis mengubah max_batches dan learning rate. Sedangkan obj.names dan obj.data untuk menginisiasikan nama dan jumlah class yang digunakan, pada hal ini yaitu hanya satu class yaitu class Human Body.

Proses *training* dilakukan dengan menggunakan Jupyter Notebook (ML Azure). Machine Learning Azure adalah salah satu produk Microsoft berbasis *cloud* yang merupakan *environment* bahasa pemrograman Python dengan format *notebook*.

3.2.2. Hasil Training

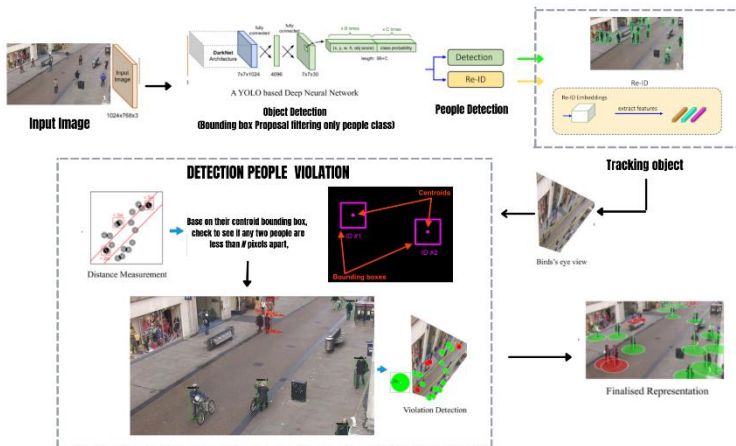
Setelah data training selesai dibuat, file .weights dapat disimpan. Kemudian menyimpan kedalam folder program. Langkah berikutnya adalah membuat kode program untuk mendeteksi jarak. Setelah objek data benar maka berikutnya dapat dilihat hasil dari program.

Name	Type
 obj.data	DATA File
 obj.names	NAMES File
 yolov4-obj.cfg	CFG File
 yolov4-obj_best.weights	WEIGHTS File

Gambar 3.5. Hasil Training

3.2.3. Implementasi

Tahap selanjutnya adalah implementasi pada android yang mana detail dari tahap ini ditunjukkan pada **Gambar 3.6**.

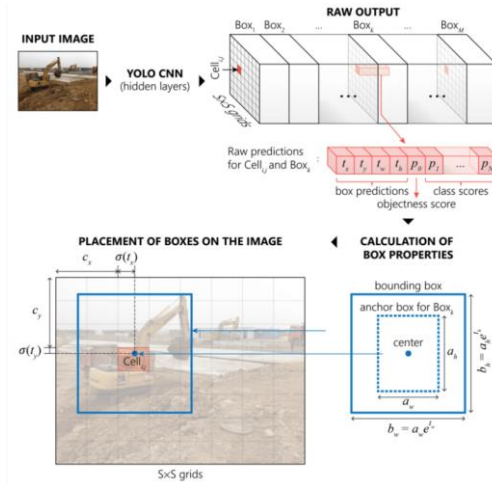


Gambar 3.6. Bagian tahap implementasi

People Detection

Saat gambar telah diinput , maka YOLO akan membagi citra dalam region/grid berukuran $s \times s$, Grid-grid tersebut bertanggung jawab untuk mendeteksi objek. Pada tiap grid juga akan diprediksi bounding box beserta nilai confidence. Tiap bounding box memiliki 5 nilai informasi yaitu x, y, w, h dan c , tiap grid akan memprediksi nilai class probabilitas jika diprediksi terdapat objek di dalamnya. Saat pengujian, YOLO akan mengkalikan nilai class probability dengan nilai confidence dari bounding box. Sehingga

menghasilkan nilai confidence kelas secara spesifik pada tiap bounding box. Nilai ini menunjukkan class probability yang muncul pada bounding box dan seberapa akurat bounding box memprediksi sesuai dengan objek, proses ini dapat diilustrasikan seperti pada **Gambar 3.7**



Gambar 3.7. Ilustrasi deteksi objek

Sumber : https://www.researchgate.net/figure/Schematic-diagram-of-the-YOLO-algorithm_fig1_343662936

Menghitung Jarak Objek

Setelah setiap objek manusia memiliki *bounding box* dan id nya masing masing, maka akan didapatkan titik tengah dari bounding box yang disebut dengan centroid. Sistem akan menghitung jarak antar centroid objek satu dengan lainnya, dalam hal ini dicari centroid dengan objek terdekat. Kemudian ditetapkanlah konsep *bird-eye view* pada citra sehingga dapat dihitung panjang horizontal dan vertical dari unit yang dideteksi. konsep ini merupakan perhitungan menggunakan rumus L2Norm atau biasa disebut dengan Euclidean distance.

$$d(\mathbf{p}, \mathbf{q}) = \left(\sum_{k=1}^n |p_k - q_k|^r \right)^{1/r} \quad (3.1)$$

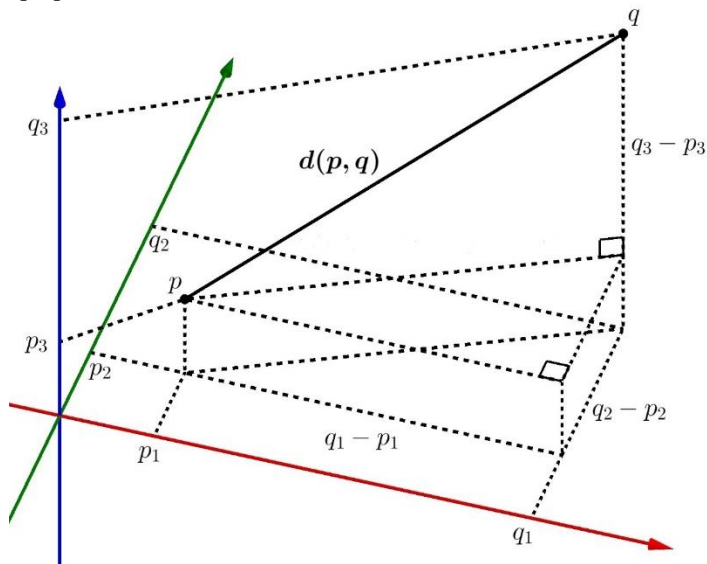
$r = 1$ hamming distance, untuk mencari jarak 2 titik dan hanya memiliki atribut binary

$r = 2$ euclidean distance (l_2 norm)

$r = \infty$ supremum (l_{\max} atau l_{∞} norm) distance. jarak maksimum antar dua titik objek.

Seperti yang terlihat pada persamaa 3.1 , L_2 Norm (Euclidean Distance) adalah metode yang digunakan untuk mengukur jarak antara 2 titik poin yang berbeda dan dapat digunakan untuk menghitung satu dimensi, dua dimensi, dan dimensi maksimal atau tidak terhingga. Pada penelitian ini diperlukan 3 vector berbeda dimana 2 titik centroid tersebut berada pada 3 sumbu x,y, dan z maka digunakan rumus dengan nilai $r=2$. Hasil dari perhitungan Euclidean distance ini nantinya berupa satuan pixel yang akan dikonversikan kedalam jarak real yaitu centimeter(cm) sehingga nantinya dapat divalidasi apakah sistem berjalan sesuai tujuan.

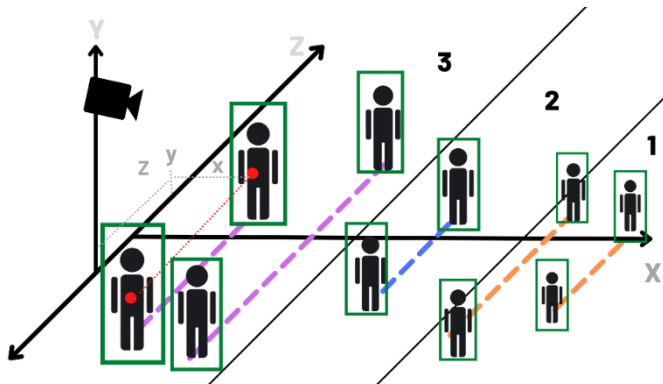
Pada **Gambar 3.8** terlihat dua titik P dan Q yang diilustrasikan sebagai centroid kedua objek. Tujuan sistem adalah mencari Panjang *distance* (d) sebagai jarak antar objek. Titik P dan Q berada pada tiga vector yang terdiri dari vector x berwarna merah, vector y berwarna biru dan vector z berwarna hijau. Maka dapat diketahui bahwa posisi titik P pada setiap vector yaitu $P(p_1, p_2, \text{dan } p_3)$ dan titik Q berada pada 3 vector yang sama dengan posisi $Q(q_1, q_2, q_3)$. [12]



Gambar 3.8. Ilustrasi perhitungan jarak objek

Kalibrasi Kamera

Distorsi yang dihasilkan oleh lensa kamera menyebabkan pengukuran jarak terukur dengan satuan pixel dari citra tidak linear terhadap jarak sesungguhnya yang terukur terhadap kamera cctv. Metode CNN digunakan untuk mendapatkan karakteristik dari jarak yang diperoleh dengan satuan pixel pada citra terhadap jarak sesungguhnya di lapangan. Maka diperlukan kalibrasi kamera yang bertujuan untuk menentukan nilai konstanta. Nilai *konstanta* tersebut digunakan untuk perhitungan konversi jarak objek kedunia nyata. Kalibrasi kamera ini dilakukan menggunakan kamera cctv milik penulis.



Gambar 3.9. Ilustrasi pembagian area untuk deteksi jarak

Gambar 3.9 merupakan ilustrasi pembagian area untuk perhitungan jarak objek pada kamera. Pada metode ini dilakukan kalibrasi kamera secara sederhana terlebih dahulu untuk menentukan nilai K (*konstanta*) masing masing area dari citra hasil tangkapan kamera cctv yang digunakan pada proyek akhir ini. Adapun untuk menentukan nilai Konstanta tersebut menggunakan Persamaan (3.2)[11].

$$\text{Konstanta} = \frac{\text{Hasil Pengukuran Pixel}}{\text{Hasil Pengukuran Centimeter}} \quad (3.2)$$

Pembagian area kedalam tiga bagian ini dilakukan untuk mendapatkan nilai konstanta yang sesuai dengan jarak kamera dengan objek.. Area 1 merupakan jarak terjauh posisi objek dengan kamera seperti yang diilustrasikan pada Gambar 3.9 . Sistem akan menghasilkan bounding box dengan luas paling kecil untuk jarak objek yang paling jauh dengan kamera. Sedangkan untuk objek pada posisi area 3 akan memiliki luas bounding box yang lebih besar. Hal ini yang mendasari pembagian area perhitungan konstanta dengan kalibrasi kamera.

Setelah didapatkan nilai kalibrasi untuk 1cm pada setiap area, maka nilai pixel tersebut dapat dikonversikan kedalam satuan centimeter(cm). proses konversi dapat dilihat dari persamaan 3.3 sebagai berikut.

$$\text{Dalam cm} = \frac{\text{Hasil Pengukuran Pixel (px)}}{\text{Hasil kalibrasi (px} \div \text{cm)}} \quad (3.3)$$

Hasil dari penggunaan persamaan 3.3 adalah jarak antar objek sebenarnya dari eksperimen dan dalam satuan cm tidak lagi dalam pixel karena sudah dilakukan kalibrasi. Kalibrasi yang dilakukan sesuai dengan data hasil pencitraan dan juga ukuran sebenarnya dari jarak antar objek, yang dibandingkan untuk mendapat jarak sebenarnya dari data eksperimen (Bakti dkk,2016).

Pelanggaran Jarak Sosial

Nilai dari hasil perhitungan jarak (d) dari perhitungan euclidean distance nantinya akan dibandingkan dengan nilai r yaitu batas maksimal jarak terdekat antar kedua objek.sesuai dengan peraturan WHO yaitu 1,83meter ~ 6 feet[10]. Kondisi 0 akan menghasilkan bounding box berwarna merah dimana mengindikasikan pelanggaran jarak . Sebaliknya jika hasil perhitungan jarak memiliki berada pada kondisi 1 maka boundingbox akan tetap berwarna hijau seperti yang terlihat pada persamaan 3.4

$$\Lambda_{\xi}(P_i, P_j, r) = \begin{cases} 1 & \text{if } \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \leq r \\ 0 & \text{if } \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} > r \end{cases} \quad (3.4)$$

BAB 4

EKSPERIMEN DAN ANALISA

Pada bab ini membahas mengenai tahapan-tahapan uji coba sistem deteksi pelanggaran jarak social distancing ini. Uji coba ini dilakukan untuk memastikan sistem berjalan sesuai dengan harapan dan dapat menyelesaikan permasalahan yang ada.

4.1. PARAMETER EKSPERIMEN

Parameter-parameter yang digunakan pada eksperimen pengujian sistem adalah:

Evaluasi deteksi manusia

Pengujian hasil deteksi manusia berfungsi untuk mengetahui performa model *deep learning* dalam mendeteksi class Human Body, sehingga parameter yang digunakan antara lain:

1. Kegiatan atau Perilaku Objek

Parameter jarak berfungsi untuk mengetahui sejauh mana sistem mampu mendeteksi objek manusia jika melakukan kegiatan yang berbeda

Evaluasi perhitungan pelanggaran jarak

Pengujian hasil perhitungan pelanggaran jarak berfungsi untuk mengetahui tingkat keakuratan sistem dalam mendeteksi pelanggaran jarak antar manusia, sehingga parameter yang dibutuhkan antara lain:

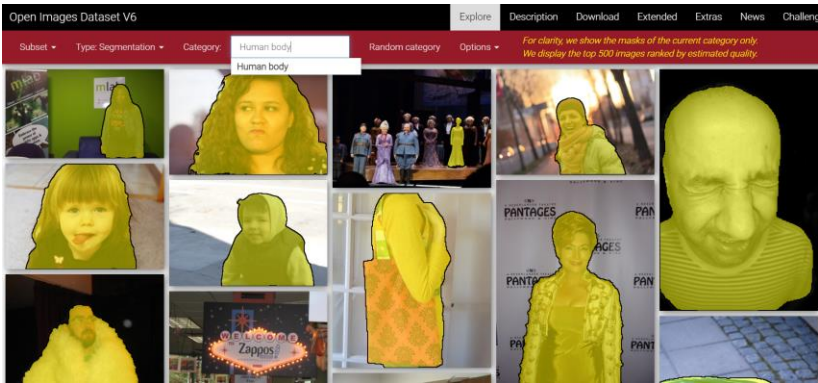
1. Jarak

Parameter jarak berfungsi untuk mengetahui sejauh mana sistem mampu mendeteksi jarak antar objek

4.2. KARAKTERISTIK DATA

Karakteristik data yang digunakan adalah data berupa gambar dataset yang diambil dari Google Open Image. Dataset Open image adalah kumpulan data dari ~9 juta gambar yang dianotasi dengan image-level labels, object

bounding boxes, object segmentation masks, visual relationships. Berisi total 16 juta bounding box untuk 600 class objek pada 1,9 juta gambar, menjadikannya dataset terbesar yang ada dengan anotasi object local. Kotak-kotak tersebut sebagian besar telah digambar secara manual oleh annotator profesional untuk memastikan akurasi dan konsistensi.



Gambar 4.1. Dataset

Gambar 4.1 merupakan gambar dataset yang digunakan. Pada gambar tersebut menampilkan berbagai macam variasi gambar manusia dengan berbagai macam kegiatan dan angle yang berbeda. Selanjutnya dataset yang telah diunduh dibagi menjadi dua, yaitu 50% untuk dataset *training* dan 50% untuk dataset testing dengan jumlah data untuk training 1000 gambar dan untuk testing 1000 gambar. Dataset tersebut memiliki format gambar .Jpg dan label format .txt dengan berbagai macam ukuran dimensi gambar.

4.3. SPESIFIKASI PERALATAN UJI COBA

Uji coba dilakukan dengan menggunakan perangkat laptop dan ponsel dengan spesifikasi yang ditunjukkan pada **Tabel 4.1** dan **Tabel 4.2**.

Tabel 4.1. Spesifikasi kamera untuk uji coba

No.	Parameter	Nilai
1.	Merk kamera	Logitech C310 HD Webcam
2.	Resolusi	720p
3.	Frame rate	30 Fps

Tabel 4.2. Spesifikasi laptop untuk uji coba

No.	Parameter	Nilai
1.	CPU	Intel Core i5 10th Gen i5-1035G1 or i7
2.	RAM	8 GB
3.	Sistem Operasi	Windows 10
4.	Tipe Sistem	64-bit
6.	Storage	512 GB SSD
7.	Graphic Cards	Intel Iris Xe Graphics

4.4. HASIL EKSPERIMEN

Pada sub-bab ini menerangkan mengenai hasil eksperimen yang telah dilakukan selama pengerjaan proyek akhir ini. Berikut adalah hasil dari eksperimen.

4.4.1. Antarmuka sistem deteksi pelanggaran jarak social

Pada sub-bab ini menerangkan mengenai hasil antarmuka beserta hasil deteksi jarak pelanggaran jarak pada proyek akhir ini.



Gambar 4.2. Antarmuka Social Distancing Detector

Pada **Gambar 4.2** menampilkan hasil deteksi dari tangkapan kamera cctv. Pada sisi pojok atas, terdapat informasi mengenai jumlah orang yang

beresiko terkena penyebaran virus akibat jarak batas aman yang dilanggar. Sesuai dengan algoritma sistem, objek yang melanggar akan memiliki bounding box berwarna merah dan saat teap berada dijarak aman akan memiliki bounding box berwarna hijau..Angka yang tercetak pada tengah boundingbox merupakan luas dari boundingbox yang merepresentasikan besaran objek dalam satuan luas pixel boundingbox ($w \times h$). Hal ini bertujuan untuk membandingkan luas untuk mendapatkan informasi area deteksi jarak.Video juga menghasilkan tampilan jarak antar objek saat dideteksi melanggar. Setiap centroid objek terdeteksi akan menarik garis dan memunculkan angka jarak dalam centimeter(cm).

4.4.2. Hasil Evaluasi Proses Training

Hasil evaluasi saat proses *training* bertujuan untuk menemukan model yang paling baik saat dilakukan proses *training*. Hal ini dikarena setiap nilai *hyperparameter* yang digunakan berbeda menghasilkan hasil evaluasi model yang berbeda pula. Dataset yang digunakan pada proses training merupakan dataset yang diunduh pada website Google OpenImage dataset seperti yang ditunjukkan pada **Gambar 4.3**. Dataset yang digunakan terdapat 2000 gambar yang terdapat objek manusia yang dibagi menjadi 1000 gambar untuk *training data* dan 1000 gambar untuk *testing data*.



Gambar 4.3. Dataset yang digunakan

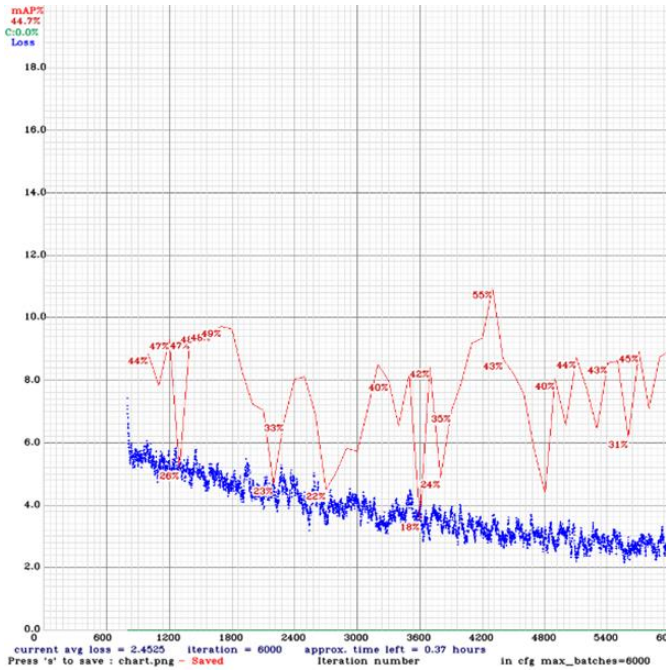
Pada proses *training* dilakukan percobaan dengan mengubah *hyperparameter* *max_batch_size* dan *learning rate*. **Tabel 4.3** merupakan

hyperparameter yang dimodifikasi untuk mendapatkan model terbaik yang kemudian digunakan pada implementasi.

Tabel 4.3. Hyperparameter

No	Parameter	YOLOv4
1	batch	64
2	Mini batch	1
3	Max Batch	6000
4	Learning Rate	0.001
5	class	1

Pada **Gambar 4.4** merupakan hasil evaluasi model pada proses *training* dengan menggunakan *pre-trained model* dan jumlah data gambar yang sama. mAP (*Mean Average Precision*) merupakan perhitungan standard yang umum digunakan untuk menghitung akurasi dari pendeteksi objek seperti Faster R-CNN, SSD, YOLO, dan lain sebagainya. Pada percobaan yang telah dilakukan, dapat diketahui bahwa dengan nilai Batch Size sebesar 64 , max_batch 6000 dan learning rate 0,001 mendapatkan hasil mAP sebesar 44,7% dan nilai total loss yaitu 2,4525. Hasil ini dinilai lebih baik jika dibandingkan dengan percobaan sebelumnya tanpa dilakukan training. nilai loss semakin mendekati nilai 0 menunjukkan performa model tersebut semakin baik dalam mendeteksi suatu objek.



Gambar 4.4. Grafik total loss dan map

4.4.3. Lingkungan Terkontrol

Pengujian dengan lingkungan terkontrol merupakan pengujian sistem yang dilakukan dengan memberikan perlakuan yang telah ditentukan terhadap variabel-variabel yang akan diteliti. Pada pengujian ini dilakukan untuk mengetahui performa sistem dalam mendeteksi pelanggaran social distancing.

4.4.2.1. Pengujian Deteksi Manusia

Pengujian ini dilakukan untuk mengetahui performa sistem dalam mendeteksi objek manusia ketika objek dalam keadaan berjalan dengan kecepatan yang tidak ditentukan (*uncontrolled*). Pengujian ini dilakukan dengan video yang diambil langsung oleh penulis dengan menggunakan kamera cctv pengujian. Video tersebut berdurasi 43 detik yang kemudian diekstrak ke dalam bentuk *frame* sebagai data masukan menggunakan aplikasi *video converter*. Hasil ekstrak video ditunjukkan pada **Gambar 4.9**.

Hasil ekstraksi video yang ditunjukkan pada **Gambar 4.9** mendapatkan sebanyak 97 gambar. Terdapat 97 gambar yang memiliki objek manusia dan 0 gambar tidak memiliki objek manusia. Kemudian sistem melakukan pendeteksian jarak pada setiap gambar. Adapun hasil pengujian ini ditunjukkan pada **Tabel 4.7**.

Tabel 4.4. Hasil pengujian deteksi objek manusia

N = 97 frame		Hasil Deteksi	
		Ter-bounding box	Tidak ter-bounding box
Kondisi Aktual	Ada objek manusia	91 (True Positive)	6 (False Negative)
	Tidak Ada objek manusia	15 (False Positive)	0 (True Negative)

Pada **Tabel 4.7**, parameter *true positive* menunjukkan sistem dapat mendeteksi objek manusia yang ada pada gambar dengan letak *bounding box* yang tepat. Dari keseluruhan gambar, sebanyak 91 gambar yang terdeteksi benar sehingga mendapatkan nilai akurasi sebesar 0,938 yang didapatkan dari **Persamaan (4.1)**.

$$Akurasi = \frac{(True\ Positive + True\ Negative)}{N} \quad (4.1)$$

4.4.2.2. Pengujian Deteksi Pelanggaran Jarak

Pengujian ini dilakukan untuk melihat persen Error dari pengukuran Eksperimen dengan Ukuran Sebenarnya

Tabel 4.5. Hasil pengujian deteksi error

no	Jarak Sebenarnya (cm)	Jarak Terdeteksi (cm)	Error (%)
1	70	70,7	1%
2	125	98	21,6%
3	65	64,5	0,77%
4	220	220	0
5	115	100	13,04%

$$Error = \left| \frac{Sebenarnya - Hasil Eksperimen}{Volume Sebenarnya} \times 100\% \right| \quad (4.2)$$

Pada **Tabel 4.5**, untuk mendapatkan nilai error dihitung menggunakan persamaan **Persamaan (4.2)**. Perbedaan hasil pada nilai error dipengaruhi oleh perhitungan konstanta dengan melakukan kalibrasi kamera untuk menghitung jarak. Dari hasil perhitungan *error* dapat dilihat bahwa nilai kesalahan pembacaan hasil pengolahan citra dan hasil sebenarnya masih masuk dalam rentang kesalahan yang kecil yaitu nilai kesalahan terbesarnya berada pada 21,6% Nilai ini merupakan nilai perbandingan yang sangat jauh rentangnya dari 100%. Sehingga dapat dikatakan bahwa hasil deteksi jarak pada penelitian ini memiliki tingkat kebenaran yang relatif tinggi.

4.5. ANALISIS HASIL EKSPERIMEN

Pada sub-bab ini menerangkan lebih terperinci mengenai analisis hasil eksperimen yang telah dipaparkan pada sub-bab sebelumnya.

4.5.1. Analisis hasil evaluasi model

Pada hasil percobaan untuk menemukan model deteksi objek didapatkan model dengan ukuran batch sebesar 64 dan learning rate 0,001 memiliki hasil evaluasi model dengan nilai mAP keseluruhan yaitu 44,7% dan nilai kerugian (Total Loss) yaitu 2,452. Dari perbandingan tersebut dapat diketahui bahwa jumlah epoch dan learning ratesangat berpengaruh pada tingkat akurasi. Semakin besar jumlah epoch maka semakin baik tingkat

akurasi, sedangkan jika semakin besar learning rate maka semakin buruk hasil akurasi dan dalam proses pelatihannya tidak stabil. Seperti yang dijelaskan oleh Skapura (1991, h. 104) bahwa semakin besar learning rate, maka ketelitian jaringan akan semakin berkurang, dan berlaku sebaliknya. Apabila semakin kecil learning rate, maka ketelitian jaringan semakin besar atau bertambah. Namun konsekuensinya adalah proses training akan semakin lama (Alfina, 2012). Hasil tersebut merupakan model terbaik walaupun nilai mAP yang didapat lebih kecil, namun hal ini berpengaruh dengan kecepatan deteksi objek. Hal ini dapat dibuktikan dengan nilai elapsed time yang tercetak saat deteksi.

```

Command Prompt - python Ap >
0.18719761118526865
0.18781665691761093
0.18890587647459986
0.18695833275995357
0.18662888276983253
0.18439690857465684
0.18693492662489117
0.18749592474735494
0.18213810258828038
0.18841470133695112
0.17168462600289977
0.18339144024559284
0.1842327197496189
0.18849425309464998
0.18710663611835984
0.1869963246338343
0.1741978808457396
0.17295986998929585
0.1847556697467031
0.17552477329178257
0.181319408292571413
0.1869963246338343
159 conv 1824 3 x 3/ 1 13 x 13 x 512 -> 13 x 13
160 conv 18 1 x 1/ 1 13 x 13 x1024 -> 13 x 13
161 yolo
[yolo] params: iou loss: ciou (4), iou_norm: 0.07, cls_norm: 1.0
max_kinds: greedy (1), beta = 0.600000
total BFLOPS: 59.563
avg_outputs = 489778
Try to load weights: backup_training\yolov4-obj_best.weights
Loading weights from backup_training\yolov4-obj_best.weights...
seen 64, trained: 275 K-images (4 Kilo-batches, 64)
Done! Loaded 162 layers from weights-file
Loaded - names_list: backup_training\obj.names, classes = 1
Used AVX
Used FMA & AVX2
0.5961001898741374
0.5969489211577863
0.6884378493349942
0.6474851640315796
0.641659369352996
0.7029083448143989
0.6928946442002694
0.7183227881919992
0.6359419111050484
0.639313842839495
0.6883688810905839
0.6637709095401186

```

Gambar 4.5 Elapsed time saat deteksi

Pada gambar 4.5 tercetak output waktu yaitu elapsed time (T) dalam *milisecond* dimana untuk mendapatkan waktu yang diperlukan untuk memproses satu frame nya adalah yaitu $1/T$. pada sisi kiri merupakan deteksi tanpa dilakukan training ulang menggunakan dataset baru dengan nilai elapsed time yang didapat $\sim 0,187$, sedangkan untuk gambar di sisi kanan merupakan hasil yang tercetak menggunakan model yang telah dilakukan training dengan nilai yang didapat yaitu $\sim 0,506$. Semakin tinggi atau semakin banyak jumlah frekuensi yang dihasilkan, gambar benda maka lebih sedikit efek blur yang menjadikan proses deteksi semakin baik (Ariningsih, 2016)

4.5.2. Analisis hasil deteksi manusia

Pada hasil percobaan deteksi objek yang ditunjukkan pada **Tabel 4.4** dapat diketahui bahwa sistem mampu mendeteksi objek manusia. Beberapa objek bukan manusia terdeteksi dikarenakan resolusi gambar yang perlu ditingkatkan sehingga dapat meminimalisir kesalahan deteksi. Jarak pandang

kamera juga perlu diatur sehingga keseluruhan badan objek terdeteksi dan hal ini dapat menghindari terjadinya oklusi

4.5.3. Analisis hasil deteksi jarak

Pada hasil percobaan yang ditunjukkan pada **Tabel 4.5** dapat diketahui bahwa sistem mampu mendeteksi jarak antar objek dan mengkonversikan objek pada citra berupa metrix pixel kedalam jarak nyata yaitu centimeter. Proses Kalibrasi kamera sangat diperlukan dalam mengkonversikan jarak citra pada jarak nyata. Hal ini dikarenakan distorsi kamera yang menyebabkan pengukuran jarak terukur dengan satuan pixel dari citra tidak linear terhadap jarak sesungguhnya yang terukur terhadap kamera cctv.

4.5.4. Analisis keseluruhan sistem

Dari keseluruhan rangkaian percobaan yang telah dilakukan, sistem pada proyek akhir ini memiliki kelemahan pada pendeteksian pelanggaran jarak. Sistem mampu mendeteksi objek dengan akurasi yang baik namun saat dilanjutkan kedalam proses perhitungan jarak, dalam beberapa Kondisi terjadi selisih nilai dari jarak nyata. Metode yang dilakukan dalam perhitungan jarak dapat dikembangkan dengan melakukan learning pada konstanta perhitungan konversi jarak pixel pada jarak nyata saat dilakukan kalibrasi kamera. Sehingga area deteksi dapat dibagi dengan jarak yang rapat untuk menghasilkan akurasi yang lebih baik.

BAB 5

PENUTUP

Pada bab ini memuat penjelasan ringkas mengenai hasil uji coba yang telah dilakukan pada bab sebelumnya. Pada bab ini juga memuat saran-saran yang dapat membangun proyek akhir ini di kemudian hari.

5.1. KESIMPULAN

Pada sub-bab ini memuat penjelasan singkat mengenai hasil uji coba. Adapun kesimpulan dari proyek akhir ini adalah sebagai berikut:

1. Hasil proses *training* menggunakan *pre-trained model* YOLOV4 mendapatkan model paling baik dengan ukuran batch 64 dan learning rate 0,001 menghabiskan waktu pembelajaran selama \pm 26 jam dan mendapatkan mAP rata-rata sebesar 44,7% dan nilai total loss yaitu 2,452.
2. Hasil deteksi objek manusia menggunakan Yolov4 dapat mendeteksi objek manusia dengan arah pandang bird eye view. Selain itu, sistem mampu mendeteksi objek manusia dengan citra objek walaupun objek yang tertangkap layar melewati batas frame kamera.
3. Hasil deteksi objek manusia dengan kondisi objek bergerak mendapatkan akurasi sebesar 0,938. Nilai tersebut cukup baik untuk deteksi manusia yang kemudia dilanjutkan untuk deteksi jarak.
4. Sistem memiliki besar kesalahan (error) rata-rata pada hasil perhitungan jarak pada citra dengan jarak yang sebenarnya yaitu 21,6%. Nilai ini belum cukup baik untuk menghitung jarak aman untuk objek manusia dalam usaha protocol Kesehatan social dirtancing.

5.2. SARAN

Selama tahap pengujian banyak sekali kekurangan yang ada pada sistem, untuk meningkatkan performa sistem, maka perlu diperhatikan hal-hal berikut:

1. Meningkatkan hasil evaluasi model deteksi objek pada proses *training* dengan menambahkan ukuran batch, learning rate dan hyperparameter

lainnya agar sistem mampu mendeteksi objek manusia dengan akurasi yang lebih tinggi.

2. Menambahkan dataset objek manusia yang diambil dari berbagai jarak dan dataset objek manusia dengan kegiatan yang lebih bervariasi.
3. Melakukan pendeteksian jarak dengan melakukan kalibrasi kamera sebelumnya dengan pembagian area yang lebih rapat untuk mendapatkan konstanta konversi jarak pada citra dengan jarak nyata yang lebih tepat.

DAFTAR PUSTAKA

- [1] Wikipedia Indonesia, "Penyakit koronavirus 2019," 10 December 2020. [Online]. Available: https://id.wikipedia.org/wiki/Penyakit_koronavirus_2019.
- [2] World Health Organization, "WHO corona-virus (COVID-19)," 2 May 2020. [Online]. Available: <https://www.who.int/emergencies/diseases/novel-coronavirus-2019>.
- [3] M. Rezael and M. Azarmi, "DeepSOCIAL: Social Distancing Monitoring and Infection Risk Assessment in COVID-19 Pandemic," *arXiv*, 2020.
- [4] D. Yang, E. Yurtsever, V. Renganathan, K. A. Redmill and U. Ozguner, "A Vision-based Social Distancing and Critical Density Detection System for COVID-19," *arXiv*, 2020.
- [5] V. V and Velantina, "Social Distancing Detection System With Artificial Intelligence Using Computer Vision And Deep Learning," *arXiv*, 2020.
- [6] A. Bochkovskiy, C.-Y. Wang and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," *arXiv*, 2020.
- [7] N. S. Pun, S. K. Sonbhadra and S. Agarwa, "Monitoring COVID-19 social distancing with person detection and tracking via fine-tuned YOLO v3 and Deepsort techniques," *arXiv*, 2020.
- [8] X. Luo, R. Shen and J. Hu, "A Deep Convolution Neural Network Model for Vehicle Recognition and Face Recognition," *arXiv*, 2017.
- [9] I. Ahmed and M. Ahmad, "A deep learning-based social distance monitoring framework for COVID-19," *arXiv*, 2021.
- [10] DC Health, "Travel Guidance," 10 December 2020. [Online]. Available: https://coronavirus.dc.gov/sites/default/coronavirus/page_content/attachment/Travel_Guidance_DCHealth_COVID-19_2021.12.16%5B25%5D.pdf.

- [11] D. Ayunita, D. S. Arief and M. Mirdanies, "PENCITRAAN DAN PEMROGRAMAN BERDASARKAN PERHITUNGAN APLIKASI VOLUME PAKET LOGISTIK," *Jom FTEKNIK Volume 6 Edisi 1*, 2019.
- [12] Y. Miftahuddin, S. Umaroh and F. R. Karim, "PERBANDINGAN METODE PERHITUNGAN JARAK EUCLIDEAN, HAVERSINE, DAN MANHATTAN DALAM PENENTUAN POSISI KARYAWAN," *Jurnal Tekno Insentif Vol. 14 No. 2*, pp. 69-77, 2020.

Biodata Penulis

Biodata Penulis



Nama : Lutfiyah Nur Fadhilah
Tempat / Tanggal Lahir : Surabaya, 4 Desember 1998
Alamat : Green Semanggi Mangrove , Cluster
Osbornia Blok E1 no 17 , Kel.
Wonorejo, Kec Rungkut , Kota
Surabaya
No. HP : 082143046608
Email : dhilasby@gmail.com
Github : <https://github.com/dhilxa>
LinkedIn : linkedin.com/in/dhilxa/
Riwayat Pendidikan :

Tahun	Lembaga	Program Studi
2005 – 2011	SD Muhammadiyah 4 Pucang Surabaya	
2011 – 2014	SMPN 19 Surabaya	
2014 – 2017	SMAN 6 Surabaya	

2017 – 2020	Politeknik Elektronika Negeri Surabaya	D3 Teknik Informatika
2020 - 2021	Politeknik Elektronika Negeri Surabaya	D4 Lanjut Jenjang Teknik Informatika