Edu Komputika 8 (1) (2021)

Edu Komputika Journal

http://journal.unnes.ac.id/sju/index.php/edukom

Aplikasi Penghitung Jarak dan Jumlah Orang Berbasis YOLO Sebagai Protokol Kesehatan Covid-19

Faizal Indaryanto[1)]✉, Anan Nugroho[2)], dan Alfa Faridh Suni[2)]

[1] Jurusan Teknik Elektro, Fakultas Teknik, Universitas Negeri Malang, Indonesia

[2] Jurusan Teknik Elektro, Fakultas Teknik, Universitas Negeri Semarang, Indonesia

Info Artikel

Sejarah Artikel:
Diterima: Juni 2021
Direvisi: Juni 2021
Disetujui: Juni 2021

Keywords:
Social Distancning, Computer Vision, Crowd Analisis, People Counting, Distance Counting

Abstrak

Pandemi Covid-19 saat ini cukup memberikan dampak pada masyarakat. Skala penyebaran dari Covid-19 sangatlah cepat, sehingga membutuhkan penanganan yang benar. Salah satu cara untuk mengurangi penyebaran Covid-19 adalah dengan melakukan *Social Distancing*. Namun masyarakat cenderung lalai dalam melaksanakan protokol kesehatan tersebut. Salah satu cara untuk mengatasi masalah ini adalah dengan aplikasi *social distancing detector* yaitu aplikasi yang digunakan untuk mendeteksi jumlah dan jarak dari objek manusia yang ada pada satu area. Penelitian ini bertujuan untuk mengembangkan aplikasi *social distancing detector* menggunakan bahasa pemrograman Python dengan library YOLOv3. YOLOv3 memiliki kelebihan dalam *object detection* dengan akurasi yang tinggi yaitu diatas 90%. Pengujian metode pada penelitian ini menggunakan lima dataset pejalan kaki dari kamera pengawas jalan yang didapatkan dari dataset uji coba beberapa peneliti melalui Github yang memiliki resolusi yang baik dan memiliki objek manusia yang heterogen. Hasil akurasi dari deteksi citra pertama adalah 83,32%. Hasil akurasi dari deteksi citra kedua adalah 76,92%. Hasil akurasi dari deteksi citra ketiga adalah 89,99%. Hasil akurasi dari deteksi citra keempat dan kelima adalah 100%. Hasil Rata-rata tingkat keberhasilan dari semua hasil analisa adalah 90,04% yang diukur dari rata-rata perbandingan jumlah data percobaan berhasil dan jumlah data pengamatan untuk tiap-tiap citra.

Abstract

The current Covid-19 pandemic has had quite an impact on society. The scale of the spread of Covid-19 is very fast, so it requires proper handling. One way to reduce the spread of Covid-19 is to practice social distancing. However, people tend to be negligent in implementing these health protocols. One way to overcome this problem is with a social distancing detector application, which is an application that is used to detect the number and distance of human objects in one area. This study aims to develop a social distancing detector application using the Python programming language with the YOLOv3 library. YOLOv3 has advantages in object detection with high accuracy, which is above 90%. Testing the method in this study used five pedestrian datasets from road surveillance cameras obtained from trial datasets by several researchers through Github which have good resolution and have heterogeneous human objects. The result of the accuracy of the first image detection is 83.32%. The result of the accuracy of the second image detection is 76.92%. The result of the accuracy of the third image detection is 89.99%. The accuracy result of the fourth and fifth image detection is 100%. Results The average success rate of all analysis results is 90.04% as measured by the average comparison of the number of successful experimental data and the number of observational data for each image.

✉ Alamat korespondensi:
Gedung E11 Lantai 2, Teknik Elektro UNNES
Kampus Sekaran, Gunungpati, Semarang, 50229
E-mail: faizal.indaryanto.1805336@students.um.ac.id

ISSN 2252-6811
E-ISSN 2599-297X

PENDAHULUAN

Berdasarkan data dari JHU CSSE COVID-19 Data Saat ini kasus baru dari covid-19 berjumlah 13.737. Angka tersebut merupakan peningkatan yang cukup tinggi dibandingkan dengan bulan sebelumnya. Hal tersebut terjadi karena perilaku masyarakat yang semakin mengabaikan protokol kesehatan. Salah satu protokol kesehatan tersebut adalah menghindari keramaian dengan menerapkan *social distancing*. Orang cenderung sulit dalam mengikuti protokol kesehatan yang satu ini. Faktor penularan yang paling penting adalah karena kontak fisik satu sama lain. Saat ini kita sedang memasuki era Revolusi Industri 4.0. Di era ini, digitalisasi dan efisiensi sangat diperlukan untuk perkembangan teknologi (Nugroho et al.) Salah satu cara untuk mengetahui batasan keramaian dan *social distancing* adalah dengan *social distancing detector* yaitu aplikasi untuk menghitung jumlah orang dan menganalisis jarak antar orang di area tertentu dengan memberikan pemberitahuan jika orang di area tersebut telah melebihi batas dan tidak menerapkan protokol kesehatan *social distancing*. Dengan memungkinkan komputer untuk mendeteksi jarak sosial, maka akan memudahkan pekerjaan manusia dengan menghitung jumlah orang dalam citra digital yang terdiri dari banyak orang, seperti di jalan raya (Rohman 2018). Pengembangan aplikasi ini menggunakan bahasa pemrograman Python dan library YOLOv3. Alasan menggunakan library YOLOv3 adalah karena YOLOv3 mampu mendeteksi objek dengan *frame rate* yang lebih tinggi (Hammam et al. 2020).

Algoritma YOLO (*You Only Look Once*) adalah algoritma *deep learning* yang memanfaatkan jaringan syaraf konvolusional (CNN) dalam mendeteksi objek. Pada proses CNN, terdapat 3 tahapan yaitu *pre-processing*, *processing*, dan *classifying* (Munadhif et al., 2020). Algoritme membagi gambar ke dalam kisi-kisi ukuran s x s, dan kemudian memprediksi kotak pembatas dan peta kelas dari setiap kisi di setiap kisi. Apabila pada satu *grid* terprediksi objek, maka pada *grid* tersebut akan diprediksi *bounding box* yang mengelilingi objek tersebut. Nilai *confidence* akan dihitung pada masing-masing *bounding box* yang kemudian akan diseleksi berdasarkan nilai yang didapat. (Nazilly et al. 2020). Sistem deteksi yang diterapkan adalah melakukan deteksi dengan menggunakan reuse classifier atau locator. (Rachmawati & Widhyaestoeti 2020). Penelitian ini bertujuan untuk mengembangkan aplikasi *social distancing detector* menggunakan bahasa pemrograman Python dengan library YOLOv3. Aplikasi ini digunakan untuk mendeteksi jumlah dan jarak objek manusia agar dapat meminimalisir penyebaran Covid-19. Aplikasi ini dapat diimplementasikan dengan cara menggunakan sistem peringatan pengeras suara ketika terdapat jumlah manusia yang berlebihan dan manusia yang saling berdekatan.

METODE PENELITIAN

Dalam melakukan *object detection*, dibutuhkan algoritma yang dapat membaca fitur dari gambar agar dapat mengenali objek pada sebuah gambar tersebut. Algoritma yang sudah teruji cepat dan akurat pada penelitian sebelumnya adalah Faster R-CNN (*Faster Region-based Convolutional Neural Network).* Meskipun cukup akurat, akurasi yang dihasilkan Faster R-CNN belum maksimal karena region yang dihasilkan oleh RPN (*Region Proposal Network*) terkadang terdeteksi sebagai *background* padahal sebenarnya mengandung objek. Metode lain yang dikembangkan untuk *Real Time Object Detection* adalah Yolo. Dimana Yolo melakukan deteksi lebih cepat dibandingkan dengan Faster R-CNN (Shianto et al., 2019). Penelitian ini menggunakan algoritma YOLO karena memiliki banyak kelebihan dibanding Faster R-CNN. Pertama, YOLO sangat cepat karena menjadikan pendeteksian objek sebagai masalah *regresi* sehingga tidak memerlukan alur yang kompleks. Kedua, YOLO mempertimbangkan secara global tentang citra saat membuat prediksi. Ketiga, YOLO dapat mempelajari generalisasi representasi objek (Pramestya, 2018).

Pengembangan aplikasi *social distancing detector* dengan menggunakan bahasa pemrograman Python dengan library YOLOv3 ini digunakan untuk mengolah dan menganalisis citra digital dan mendeteksi jarak dan menghitung jumlah orang pada citra digital yang diinputkan. Aplikasi ini akan diimplementasikan dalam bentuk sistem aplikasi *personal computer* yang terintegrasi dengan kamera atau CCTV, yang akan memberikan output sistem peringatan ketika terdapat jumlah manusia yang berlebihan dan manusia yang saling berdekatan.

A. Landasan Teori

You Only Look Once atau YOLO adalah Jaringan saraf cerdas untuk deteksi waktu nyata. YOLO menerapkan jaringan syaraf tunggal pada keseluruhan gambar. Jaringan ini akan dibagi menjadi daerah-daerah, kemudian akan memprediksi kotak pembatas dan probabilitas.

Untuk setiap kotak di wilayah pembatas, timbang probabilitas untuk mengklasifikasikannya sebagai objek atau bukan. YOLO membagi citra masukan menjadi kisi-kisi berukuran [11], dimana nilai S adalah 7 dan ukuran citra masukan adalah 448 × 448. Untuk mendapatkan kotak pembatas, kita akan menggulung gambar input. Sebuah kotak pembatas memiliki 5 nilai yang harus disimpan: koordinat, koordinat, lebar (*width*), tinggi (*height*), skor kepercayaan (kotak pembatas yang dimaksud memiliki nilai probabilitas objek) (Hermawan et al., 2021). YOLOv3 memiliki 53 lapisan konvolutional, yang disebut Darknet 53, yang terdiri dari struktur konvolutional dan residual. Di YOLOv3, lapisan konvolusi selalu diikuti dengan normalisasi *batch* dan kebocoran ReLu. Sambungan blok residual atau pintasan di YOLOv3 diselesaikan dengan menambahkan entri di atas blok residu lapisan konvolusi ke hasil filter lapisan konvolusi 1x1, lalu melakukan normalisasi *batch* dan penyaringan ReLu, dilanjutkan dengan *convolutional layer filter* 3x3 dengan *batch normalization*, dan pada bagian akhir dilakukan *leaky* ReLu. *Residual layer* pada YOLOv3 didefinisikan sebagai operasi 18 penjumlahan antara input awal dengan output hasil sesudah dilakukan *convolutional layer* pada *residual block* (Putra et al., 2021).

OpenCV (*Open Source Computer Vision Library*) adalah perpustakaan perangkat lunak untuk pemrosesan gambar waktu nyata yang dinamis, dibuat oleh Intel, dan sekarang didukung oleh Willow Garage dan Itseez. OpenCV dirilis di bawah lisensi BSD yang lebih liberal daripada GPL, dan memberikan kebebasan penuh untuk penggunaan komersial tanpa mengungkapkan kode sumbernya. Ini juga memiliki antarmuka yang mendukung bahasa pemrograman C ++, C, Python, dan Java, termasuk sistem operasi Windows, Linux, Mac OS, iOS, dan Android. OpenCV dirancang untuk efisiensi komputasi, dengan fokus pada aplikasi waktu nyata (Zein 2018).

Object Detection atau deteksi objek merupakan bagian dari *Computer Vision*. *Object Detection* mengacu pada kemampuan komputer untuk mendeteksi sejumlah objek pada suatu gambar. Hal ini dapat dilakukan dengan cara mengambil *image feature* seperti garis, sudut, kontur dan warna dari sebuah gambar. Deteksi objek merupakan bagian dari *Object Recognition* atau identifikasi objek. Oleh karena itu dapat disimpulkan bahwa untuk deteksi objek pasti harus diidentifikasi terlebih dahulu objek tersebut. Sedangkan pada penelitian ini akan dilakukan deteksi objek kemudian di implementasikan deteksi jarak dan perhitungan jumlah objek (Rahmad et al., 2018).

B. Alat & Bahan

Pada penelitian ini digunakan lima dataset pejalan kaki dari kamera pengawas jalan yang didapatkan dari dataset uji coba beberapa peneliti melalui Github yang memiliki resolusi yang baik dan memiliki objek manusia yang heterogen. Dari dataset tersebut akan diuji coba menggunakan aplikasi yang telah dikembangkan menggunakan Jupiter Notebook. Uji coba penelitian ini dilakukan menggunakan laptop dengan spesifikasi Intel core i5, Nvidia Geforce 920MX dengan RAM 8 GB.

C. Usulan Metode

Dalam implementasi aplikasi ini dibutuhkan sekumpulan data yang terkait, data yang dibutuhkan di penelitian ini adalah citra orang yang didapatkan dari dataset uji coba beberapa peneliti melalui Github yang memiliki resolusi yang baik dan memiliki objek manusia yang heterogen. Aplikasi ini menggunakan dataset COCO sebagai *trained* data untuk melakukan *object detection* yang telah disediakan YOLOv3. Perhitungan jarak menggunakan *Non-maxima suppression* untuk mengurangi kotak pembatas yang tumpang tindih menjadi hanya satu kotak pembatas sehingga mewakili deteksi objek sebenarnya. Dalam algoritma deteksi objek, ada kemungkinan terdapat lebih dari satu *bounding box* mendeteksi objek yang sama. Maka, diperlukan *Non-Max Suppression* yang memiliki peran penting dalam memilih *bounding box* dengan nilai *confidence score* yang lebih tinggi. Pada *Non-Max Suppression*, sistem pada awalnya menyeleksi nilai *confident* score yang tertinggi di antara tiga *bounding box* tersebut sebelum kemudian menghapus *bounding box* lain yang memiliki nilai *confidence score* yang lebih rendah (Kusuma et al., 2021).

Kemudian jarak dihitung antara *centroid* dari setiap orang. Cara kerja aplikasi ini ditunjukkan oleh Gambar 1.

Gambar 1. *Flowchart* cara kerja aplikasi

Pada *flowchart* cara kerja aplikasi di Gambar 1. Gambar input akan diproses oleh program untuk menggunakan YOLOv3 untuk deteksi. Program akan mendeteksi orang yang berada pada citra tersebut. Setelah program mendeteksi orang proses selanjutnya adalah dengan menghitung jarak antar orang melalui *centroid* dari setiap orang yang terdeteksi. Kemudian akan dihitung jarak antar *centroid* dari setiap orang, ketika ada jarak antar *centroid* yang berdekatan 50px akan ditampilkan sebagai *high risk*. Lalu citra yang sudah diberi *bounding box* dan *text* ditampilkan sebagai keluaran. Setelah itu keluaran tersebut akan disimpan dalam bentuk *file*. Dalam penelitian ini, skema pengujian dengan beberapa gambar yang berbeda digunakan. Jumlah gambar yang digunakan adalah lima gambar. Citra digital yang didapat dari proses deteksi objek dan perhitungan jarak dari aplikasi akan dibandingkan dengan hasil pengamatan citra digital yang dilakukan oleh penulis.

Algoritma YOLO adalah algoritma *deep learning* untuk pendeteksian objek, menggunakan metode yang berbeda dari algoritma lain, yaitu jaringan saraf tunggal diterapkan ke seluruh gambar. (Nazilly et al. 2020). Bagilah gambar menjadi area/kisi berukuran sxs. *Grid* bertanggung jawab untuk mendeteksi objek. Di setiap *grid*, kotak pembatas juga akan diprediksi bersama dengan nilai kepercayaannya. Nilai kepercayaan ini menunjukkan seberapa aman kotak pembatas berisi objek dan seberapa akurat prediksinya. Setiap kotak pembatas memiliki 5 nilai informasi, yaitu x, y, w, h, dan c. Nilai x dan y adalah koordinat titik tengah yang diharapkan dari kotak pembatas, nilai w dan h adalah rasio aspek relatif terhadap kisi, dan c adalah nilai kotak pembatas kepercayaan. Dalam algoritma YOLO, jika ada objek prediksi di setiap *grid*, nilai kelas probabilitas akan diprediksi. Saat pengujian, YOLO akan mengalikan nilai kelas probabilitas dengan nilai kepercayaan kotak pembatas.

Paradigma paralel dalam GPU sangat penting untuk penyelesaian operasi grafis yang cepat. Umumnya, GPU melakukan perhitungan data dalam bentuk angka dalam gambar (grafik komputer), atau sebaliknya (penglihatan komputer). Ini termasuk dalam GPU tujuan umum. GPU sendiri memiliki arsitektur yang unik, yaitu terdapat prosesor *multi-threaded* yang dapat menjalankan beberapa proses secara bersamaan. Pada saat yang sama, proses paralel menggunakan beberapa komponen komputasi untuk memecahkan masalah besar dan kompleks dalam komputasi. (Jupiyandi et al. 2019).

HASIL DAN PEMBAHASAN

Analisis yang dilakukan dalam penelitian ini adalah untuk mengamati bagaimana hasil proses pendeteksian citra digital dari aplikasi yang dikembangkan sesuai dengan hasil yang diamati oleh penulis. Berdasarkan hal tersebut kita dapat mengetahui tingkat keberhasilan aplikasi ini dalam menentukan jumlah orang dan menentukan jarak antar orang.

Persentase tingkat keberhasilan dihitung dengan menggunakan rumus berikut.

$$k = \frac{n}{m} \times 100\%$$

Dimana:
k = persentase tingkat keberhasilan
n = jumlah data percobaan yang berhasil
m = jumlah data pengamatan

Objek gambar yang digunakan dalam penelitian ini adalah dataset uji coba beberapa peneliti melalui Github yang memiliki resolusi yang baik dan memiliki objek manusia yang heterogen. Citra tersebut akan diproses untuk mendeteksi jumlah dan jarak dari objek manusia. Objek manusia yang terdeteksi akan ditampilkan pada *bounding box* warna hijau. Objek manusia yang berjarak kurang dari 50px akan ditampilkan sebagai *high risk* pada *bounding box* warna merah. Dilakukan lima analisa terhadap lima citra digital yang telah dideteksi. Hasil pengujian data citra digital 1 ditunjukkan pada Gambar 2 dan Gambar 3.

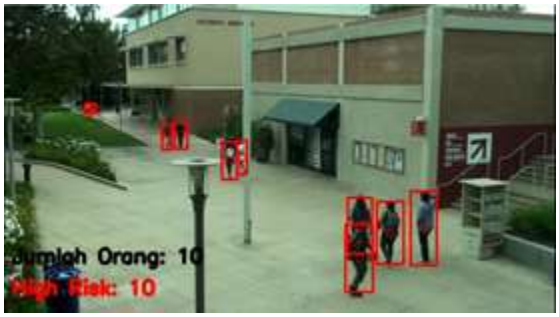
Gambar 2. Citra 1

Gambar 3. Deteksi citra 1

Pada Gambar 2 dan Gambar 3, Anda dapat melihat hasil pengujian pada gambar digital pertama. Tingkat keberhasilan dari deteksi ini untuk perhitungan jumlah orang adalah 88,88%. Dilihat dari tingkat keberhasilannya, aplikasi ini akan berfungsi dengan baik. Dan tingkat kegagalan adalah 11,11% dikarenakan objek yang saling timpang tindih dan juga terdapat objek manusia yang sangat jauh, sehingga program tidak dapat mendeteksi objek tersebut.

Tingkat keberhasilan dari deteksi ini untuk perhitungan jarak orang adalah 77,77%. Dilihat dari tingkat keberhasilannya, hal ini menunjukkan bahwa aplikasi dapat berjalan dengan cukup baik. Dan tingkat kegagalannya adalah 22,22% karena program masih sulit untuk mendeteksi object yang jauh. Hasil pengujian data citra digital 2 ditunjukkan pada Gambar 4 dan Gambar 5.

Gambar 4. Citra 2



Gambar 5. Deteksi citra 2

Pada Gambar 4 dan Gambar 5, Anda dapat melihat hasil pengujian pada gambar digital kedua. Tingkat keberhasilan dari deteksi ini untuk perhitungan jumlah orang adalah 76,92%. Dari persentase tingkat keberhasilan ini, aplikasi dapat berjalan dengan baik. Dan tingkat kegagalannya adalah 23,07% dikarenakan ada beberapa objek manusia yang cukup jauh dan juga objek tersebut kurang memiliki kontras dengan lingkungan sekitar sehingga program tidak dapat mendeteksi objek tersebut.

Tingkat keberhasilan dari deteksi ini untuk perhitungan jarak orang adalah 76,92%. Dilihat dari tingkat keberhasilan ini, aplikasi ini dapat berkinerja cukup baik. Dan tingkat kegagalannya adalah 23,07% karena program masih belum dapat mendeteksi beberapa objek manusia yang tidak berdekatan berkaitan dengan perhitungan jumlah orang. Hasil pengujian data citra digital 3 ditunjukkan pada Gambar 6 dan Gambar 7.

Gambar 6. Citra 3

Gambar 7. Deteksi citra 3

Pada Gambar 6 dan Gambar 7, Anda dapat melihat hasil pengujian pada gambar digital ketiga. Tingkat keberhasilan dari deteksi ini untuk perhitungan jumlah orang adalah 93,33%. Dilihat dari tingkat keberhasilan ini, aplikasi ini mungkin berkinerja baik. Dan tingkat kegagalannya adalah 6,66% dikarenakan objek yang terpotong sehingga program mendeteksi objek yang lain.

Tingkat keberhasilan dari deteksi ini untuk perhitungan jarak orang adalah 86,66%. Dilihat dari persentase tingkat keberhasilannya, aplikasi ini dapat berjalan dengan baik. Dan tingkat kegagalannya adalah 13,33% karena program mendeteksi objek yang lain sehingga program mendeteksi ada objek dengan jarak yang dekat. Hasil pengujian data citra digital 4 ditunjukkan pada Gambar 8 dan Gambar 9.

Gambar 8. Citra 4

Gambar 9. Deteksi citra 4

Pada Gambar 8 dan Gambar 9, Anda dapat melihat hasil pengujian pada gambar digital keempat. Tingkat keberhasilan dari deteksi ini untuk perhitungan jumlah orang adalah 100%. Dilihat dari persentase tingkat keberhasilan ini, aplikasi dapat berjalan dengan baik.

Tingkat keberhasilan dari deteksi ini untuk perhitungan jarak orang adalah 100%. Dilihat dari tingkat keberhasilan ini, aplikasi ini akan berkinerja sangat baik. Hasil pengujian data citra digital 5 ditunjukkan pada Gambar 10 dan Gambar 11.

Gambar 10. Citra 5

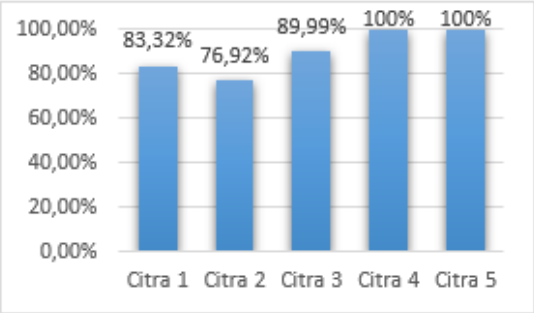
Gambar 11. Deteksi citra 5

Pada Gambar 10 dan Gambar 11 dapat dilihat hasil pengujian dari kelima citra digital tersebut. Tingkat keberhasilan dari deteksi ini untuk perhitungan jumlah orang adalah 100%. Dari persentase tingkat keberhasilan ini, aplikasi dapat berjalan dengan sangat baik.

Tingkat keberhasilan dari deteksi ini untuk perhitungan jarak orang adalah 100%. Dilihat dari tingkat keberhasilannya, aplikasi ini akan berfungsi dengan sangat baik. Perbandingan seluruh hasil Analisis yang dilakukan pada percobaan sebelumnya ditunjukkan pada Tabel 1.

Tabel 1. Perbandingan Hasil Analisa

Citra	Deteksi Jumlah Orang	Deteksi *High Risk*	Akurasi
1	8	5	83,32%
2	10	*10*	76,92%
3	15	9	89,99%
4	5	2	100%
5	3	0	100%



Gambar 12. Grafik rata-rata akurasi

Dari Tabel 1 dan Gambar 12 diatas dapat dilihat rata-rata akurasi perbandingan dari hasil analisa pengujian yang telah dilakukan adalah sebesar 90,04%. Besarnya nilai akurasi ini menunjukkan bahwa aplikasi pendeteksi jarak sosial ini dapat bekerja dengan baik. 9,94% dari hasil tes global gagal karena input yang diterima memiliki informasi yang luas. Dalam artian banyak objek yang berada pada daerah yang jauh dan kurangnya kontras antara objek dengan latar belakang. Akibatnya program akan kesulitan dalam mendeteksi objek manusia.

Berdasarkan hasil rata-rata tersebut dapat dilihat bahwa performa dari metode YOLO sangat baik dengan persentase akurasi yang melebihi 90%. Jika dibandingkan dengan metode Faster R-CNN yang masih memiliki rata-rata akurasi dibawah 90% (Ma'ali & A, 2019), YOLO dapat menjadi pilihan yang lebih baik jika dilihat dari akurasi dan kecepatan dalam proses deteksi objek.

SIMPULAN

Berdasarkan hasil pengujian yang telah dilakukan dapat disimpulkan bahwa aplikasi dapat digunakan untuk mendeteksi jumlah orang dan mendeteksi jarak seseorang. Dapat dikatakan bahwa tujuan dari penelitian ini tercapai karena aplikasi yang dikembangkan memiliki hasil akurasi yang baik dengan rata-rata akurasi hasil analisa sebesar 90,04%. Berdasarkan analisa tersebut dapat disimpulkan bahwa aplikasi dapat diterapkan pada lingkungan masyarakat untuk meredakan gejolak Covid-19 yang semakin meningkat. Ada beberapa faktor yang mempengaruhi akurasi dari aplikasi ini. Semakin jelas input yang digunakan maka akurasi yang akan didapatkan juga akan lebih baik. Faktor yang dapat mengurangi akurasi dalam program ini yaitu objek manusia yang berada pada kejauhan dan kurangnya kontras antara objek manusia dengan latar belakang. Penelitian ini masih kurang dalam penggunaan dataset karena keterbatasan sumber daya yang ada. Penerapan aplikasi ini diharapkan dapat meminimalisir penyebaran Covid19, dan dapat dilakukan lebih banyak penelitian dengan kumpulan data yang lebih banyak dan lebih baik untuk meningkatkan akurasi, sehingga aplikasi dapat segera diimplementasikan.

UCAPAN TERIMAKASIH

Peneliti mengucapkan terima kasih kepada dosen mata kuliah *computer vision* Universitas Negeri Semarang dan semua pihak yang telah mendukung dan membantu terselenggaranya penelitian ini.

DAFTAR PUSTAKA

Anan Nugroho, Risanuri Hidayat, Hanung Nugroho, Johan Debayle. Ultrasound object detection using morphological region-based active contour: an application system. International Journal of Innovation and Learning, Inderscience, In press.

Dong E, Du H, Gardner L. An interactive web-based dashboard to track COVID-19 in real time. Lancet Inf Dis. 20(5):533-534. doi: 10.1016/S1473-3099(20)30120-1

Hammam, H., Asyhar, A., Wibowo, S. A., & Budiman, G. (2020). Implementasi Dan Analisis Performansi Metode You Only Look Once (Yolo) Sebagai Sensor Pornografi Pada Video Implementation and Performance Analysis of You Only Look Once (Yolo) Method As Porn Censorship in Video. 7(2), 3631–3638.

Hermawan, M. I., Tritoasmoro, I. I., Ibrahim, N., Elektro, F. T., Telkom, U., Only, Y., Once, L., Online, S., & Tracking, R. (2021). PENGATURAN LAMPU LALU LINTAS BERDASARKAN KEPADATAN KENDARAAN MENGGUNAKAN METODE YOLO. 8(1), 198–205.

Jupiyandi, S., Saniputra, F. R., Pratama, Y., Dharmawan, M. R., & Cholissodin, I. (2019). Pengembangan Deteksi Citra Mobil untuk Mengetahui Jumlah Tempat Parkir Menggunakan CUDA dan Modified YOLO. Jurnal Teknologi Informasi Dan Ilmu Komputer, 6(4), 413. https://doi.org/10.25126/jtiik.2019641275

Kusuma, T. A. A., Usman, K., & Saidah, S. (2021). PEOPLE COUNTING FOR PUBLIC TRANSPORTATIONS USING YOU ONLY LOOK ONCE METHOD. 2(1), 57–66.

Ma'ali, A. M., & A, M. S. H. (2019). Rancang Bangun Sistem Pengendali Lampu Lalu Lintas Berdasarkan Pengenalan Citra Digital Kendaraan Menggunakan Metode Faster R-Cnn. http://eprints.uty.ac.id/3335/

Munadhif, I., Fathoni, D. H., & Jamiin, A. (2020). PENGENDALIAN CCTV MENGGUNAKAN YOU ONLY LOOK ONCE (YOLO). 6(1), 958–965.

Nazilly, M. L., Rahmat, B., & Puspaningrum, E. Y. (2020). Implementasi Algoritma Yolo (You Only Look Once) Untuk Deteksi Api. Jurnal Informatika Dan Sistem Informasi, 1(1), 81–91.

Pramestya, R. H. (2018). Deteksi dan klasifikasi kerusakan jalan aspal menggunakan metode yolo berbasis citra digital. Master Thesis. http://repository.its.ac.id/59044/1/06111650010019-Master_Thesis.pdf

Putra, B., Pamungkas, G., Nugroho, B., & Anggraeny, F. (2021). DETEKSI DAN MENGHITUNG MANUSIA MENGGUNAKAN YOLO-CNN. 02(1), 67–76.

Rachmawati, F., & Widhyaestoeti, D. (2020). Deteksi Jumlah Kendaraan di Jalur SSA Kota Bogor Menggunakan Algoritma Deep Learning YOLO. Prosiding LPPM UIKA Bogor, 360–370.

Rahmad, E. C., Kom, S. T. M., Rawansyah, D., Pd, M., & Rochastu, T. K. (2018). Object Detection System Sebagai Alat Bantu Mendeteksi Objek Sekitar untuk Penyandang Tunanetra. 81–88.

Rohman, Q. A. (2018). Implementasi Fitur Haar-like dalam Mendeteksi dan Menghitung Jumlah Orang pada Noised Digital Image. *Journal of Informatic, Information System, Software Engineering and Application*, *1*(1), 40-48. doi: 0.20895/INISTA.V1I1

Shianto, K. A., Gunadi, K., & Setyati, E. (2019). Deteksi Jenis Mobil Menggunakan Metode YOLO Dan Faster R-CNN. Jurnal Infra, 7(1), 157–163.

Zein, A. (2018). Pendeteksian Kantuk Secara Real Time Menggunakan Pustaka OPENCV dan DLIB PYTHON. Sainstech: Jurnal Penelitian Dan Pengkajian Sains Dan Teknologi, 28(2), 22–26. https://doi.org/10.37277/stch.v28i2.238