

SISTEM DETEKSI PELANGGARAN *SOCIAL DISTANCING* DI RUANG TERBUKA MENGUNAKAN ALGORITMA *YOU ONLY LOOK ONCE* (YOLO)

OUTDOOR SOCIAL DISTANCING VIOLATION SYSTEM DETECTION USING YOU ONLY LOOK ONCE (YOLO) ALGORITHM

Dwiki Rahman¹, Casi Setianingsih², Fussy Mentari Dirgantara³

^{1,2,3} Universitas Telkom, Bandung

dkrahm@telkomuniversity.ac.id¹, setiacasie@telkomuniversity.ac.id², fussymentari@telkomuniversity.ac.id²

Abstrak

Pemerintah mengeluarkan perintah untuk menerapkan *social distancing* atau *physical distancing*. *Social distancing* merupakan metode menjaga jarak setidaknya satu meter dari orang lain. Hal ini berguna untuk mengurangi / mencegah penularan penyakit (virus) dan mengurangi rantai penyebaran covid-19. Sehingga rumah sakit bisa memberikan pelayanan secara optimal. Untuk itu, Tugas Akhir ini disusun untuk membuat sebuah sistem yang dapat mendeteksi pelanggaran *social distancing* pada suatu tempat terbuka. Sistem menggunakan algoritma *You Only Look Once* (YOLO). Sistem yang dikembangkan menggunakan model Yolov4 yang telah di *pre-trained* untuk mendeteksi 80 kelas objek. Pengujian sistem ini dilakukan berdasarkan beberapa skenario. Sistem diprogram menggunakan bahasa pemrograman Python dengan *tools* yang digunakan untuk *coding* adalah Microsoft Visual Studio Code dan Anaconda. Hasil terbaik dari pembuatan model deteksi didapat dari rasio dataset yaitu 90% data *train* dan 10% data *test* dengan hasil *mean average precision* yang didapatkan adalah 54,11%, dan dengan hasil uji akurasi deteksi pada saat pengujian sebesar 100%.

Kata Kunci : akurasi, *mean average precision*, python, *social distancing*, yolov4

Abstract

The government issued orders to implement *social distancing* or *physical distancing*. *Social Distancing* is a method of maintaining a distance of at least one meter from other people. This is useful for reducing / preventing disease transmission (virus) and reduce chain of spread of covid-19. So the hospital can provide optimal service. For this reason, this final project is structured to create a system that can detect violations of *social distancing* in an open place. This system uses the *You Only Look Once* (YOLO) algorithm. The developed system uses a *pre-trained* Yolov4 model to detect 80 object classes. Testing of this system is carried out based on several scenarios. The system is programmed using Python programming language with tools used for coding are Microsoft Visual Studio Code and Anaconda. The best results from the creation of the detection model are obtained from dataset ratio of 90% train data and 10% test data with the *mean average precision* results obtained is 54.11%, and the detection accuracy at the time of testing was 100%.

Keywords: accuracy, *mean average precision*, python, *social distancing*, yolov4

1. Pendahuluan

Social Distancing atau bisa disebut juga dengan *Physical Distancing* artinya menjaga jarak aman antara diri sendiri dengan orang lain terutama orang yang tidak satu rumah [1]. Jarak aman yang disebutkan adalah jarak satu hingga dua meter. Cara ini juga harus diikuti dengan protokol lainnya seperti harus selalu memakai masker, mencuci tangan, dan protokol lainnya.

Program pendeteksian pelanggaran *Social Distancing* sudah pernah diteliti oleh beberapa peneliti pendahulu. Salah satunya di penelitian yang berjudul “*Social Distancing Detection with Deep Learning Model*” [2]. Pada penelitian ini menggunakan algoritma YOLO V3 yang merupakan pengembangan dari YOLO V2 dalam pendeteksian manusianya. Dalam mengukur jarak antar orang, di dalam penelitian ini menggunakan *top-down view* untuk *perspective view* dari kamera yang digunakan untuk meningkatkan akurasi dari pengukuran jarak. Kelemahan dari penelitian ini adalah, bisa terjadinya *overlay* dimana terdapat dua orang yang berdekatan sehingga terlihat berdempetan sehingga dianggap tidak melanggar *Social Distancing*.

Maka dari itu, pada Tugas Akhir ini penulis mengusulkan untuk menggunakan algoritma *You Only Look Once* (YOLO) yang memiliki performansi yang lebih tinggi dari versi sebelumnya yaitu YOLOv4. Tujuan dari penelitian ini adalah membuat sistem yang mendeteksi pelanggaran secara *real-time* dengan efektif dan akurat di lingkungan *outdoor*, serta dapat diakses menggunakan *website* untuk mempermudah penggunaan.

2. Dasar Teori

2.1 Citra Digital

Citra adalah suatu gambaran atau kemiripan dari suatu objek [3]. Citra merupakan kombinasi dari titik, garis, bidang, dan warna untuk merepresentasikan suatu objek dari data yang ada.

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ \vdots & \vdots & \vdots & \vdots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1,N-1) \end{bmatrix}$$

Gambar 2.1 Matriks citra digital

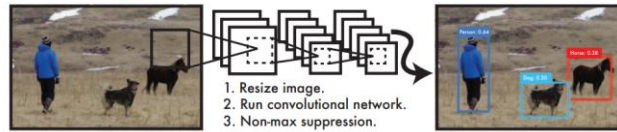
2.2 Deep Learning

Deep Learning merupakan cabang dari *Machine Learning* dimana pemodelan data bersifat abstrak tingkat tinggi dan memiliki *layer* yang sangat banyak dan juga kompleks. Pembelajaran di dalam *Deep Learning* bersifat *supervised*, *semi-supervised*, dan *unsupervised* [4].

2.3 You Only Look Once (YOLO)

You Only Look Once (YOLO) merupakan algoritma *Deep Learning* yang terus dikembangkan sampai sekarang. YOLO juga merupakan pengembangan dari CNN dan diciptakan oleh *Joseph Redmon* pada tahun 2015. Algoritma YOLO menggunakan *single neural network* untuk mendeteksi *bounding box* dan *class probabilities* secara langsung dari keseluruhan gambar dalam sekali evaluasi [5]. Algoritma ini juga didukung dengan bantuan GPU untuk dapat memproses gambar secara *real-time* hingga 45 fps [5]. Pada gambar di bawah, dijelaskan secara ringkas bagaimana proses algoritma ini dilakukan. Dimulai dengan mengubah atau

resize gambar dan proses *convolutional network* dilakukan. Terakhir dengan metode *non-max suppression* akan menemukan objek dengan *bounding boxes* yang memiliki *confidence score* tertinggi.



Gambar 2.2 Sistem Deteksi YOLO [5]

2.4 Transfer Learning

Transfer Learning merupakan sebuah metode yang menggunakan suatu model yang telah terbentuk dan dilatih oleh suatu *dataset* dan kemudian model tersebut digunakan kembali untuk permasalahan yang berbeda dengan mengganti atau memodifikasi parameter atau konfigurasinya sesuai dengan *dataset* yang baru [6].

2.5 Euclidean Distance

Dalam menghitung jarak antar orang berdasarkan yang terdeteksi, akan menggunakan metode *Euclidean Distance*. Metode ini akan menghitung jarak antar objek dengan mengukur jarak dua titik di dalam *Euclidean Space*, dan dapat berupa dua dimensi ataupun tiga dimensi [7].

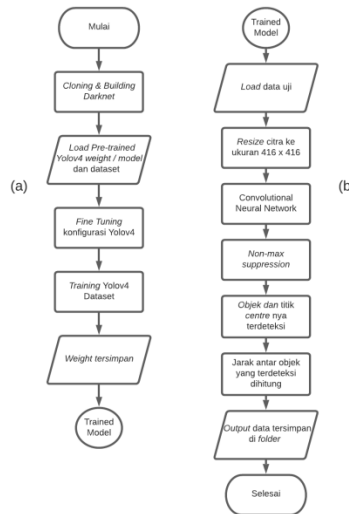
Untuk mengukur tingkat kemiripan data, digunakan rumus sebagai berikut :

$$d(x,y) = |x-y| = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{2.1}$$

3. Pembahasan

3.1 Desain Sistem

Pada perancangan sistem ini, memiliki dua proses utama, yaitu *training* model dan pengujian model (*testing*).

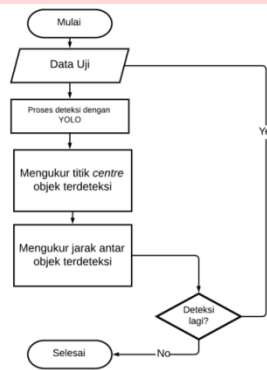


Gambar 3.1 Flowchart Training dan Testing dari sistem

Dapat dilihat dari gambar *flowchart* diatas, bagian (a) menunjukkan proses *training* yang dilakukan terhadap dataset. Proses dimulai dengan *building Darknet* sebagai arsitektur dari YOLOv4. Kemudian akan dilakukan penyesuaian atau *fine-tuning* terhadap *configuration file* dari *pre-trained YOLOv4 model* untuk data latih baru, dan setelah itu *training* dapat dilakukan. *Output* yang dihasilkan akan berupa *weight file* yang akan tersimpan, untuk digunakan pada sistem.

Sementara itu, bagian (b) menunjukkan alur bagaimana sistem mendeteksi objek hingga dapat menghitung jarak antar objek untuk menentukan pelanggaran dari *social distancing*. *Output* hasil *training* akan mempengaruhi tingkat akurasi yang diberikan saat dilakukan *testing*. Sistem akan membaca *weight file* yang telah dilatih untuk melakukan deteksi objek beserta perhitungan jarak antar objek, dan kemudian akan menghasilkan rekaman pendeteksian.

Secara umum, alur dari cara kerja sistem bisa dilihat melalui *flowchart* di bawah ini.



Gambar 3.2 Alur Kerja Sistem

Dari gambar di atas, sistem dimulai dengan memasukkan data uji yang akan digunakan yaitu berupa *file* video yang kemudian sistem akan menjalankan proses deteksi dengan algoritma YOLO kemudian dari objek yang terdeteksi, dihitung titik *centre*-nya dan jarak antar titik *centre* tersebut.

3.2 Hasil Training

Training untuk dataset menggunakan 3 rasio berbeda yaitu 90%:10%, 80%:20%, 70%:30% dengan keterangan data *train* : data *test*. Hasil terbaik dari 3 rasio tersebut kemudian akan dilatih lagi dengan parameter yang berbeda. Hasil *training* keseluruhan dapat dilihat pada tabel 1 dan 2 di bawah ini.

Tabel 3.1 Hasil *Training* rasio berbeda

Rasio Dataset	Presisi	Recall	F1 Score	Average IoU	Average Loss	mAP
90% : 10%	61%	63%	62%	44.61%	1	51.93%
80% : 20%	35%	59 %	44%	26.47%	1	32.34%
70% : 30%	39%	54%	45%	29.58%	1	26.18%

Tabel 3.2 Hasil *training* parameter berbeda

Max Batches	Learning Rate	Presisi	Recall	F1 Score	Average IoU	mAP
1000	0.001	51%	61%	55%	37.10%	47.20%

	0.002	52%	66%	58%	37.10%	53.97%
2000	0.001	62%	64%	63%	45.83%	54.11%
	0.002	50%	63%	56%	36.31%	51.14%

3.3 Hasil Pengujian

Pengujian sistem ini menerapkan dua skenario yaitu, skenario pengambilan video CCTV dan skenario pengambilan video sejajar dengan objek. Hasil yang didapatkan dapat dilihat pada tabel 3 dan 4 di bawah ini.

Tabel 3.3 Tabel hasil pengujian skenario CCTV

No	Sudut	Jarak (cm)	Hasil Pengujian		Keterangan Gambar
			Berhasil	Tidak berhasil	
1	0°	150	√		Gambar 4.17
2	30° kanan	150	√		Gambar 4.18
3	30° kiri	150	√		Gambar 4.19
4	60° kanan	150	√		Gambar 4.20
5	60° kiri	150		√	Gambar 4.21

Tabel 3.4 Hasil pengujian skenario sejajar

No	Sudut	Jarak (cm)	Hasil Pengujian		Keterangan Gambar
			Berhasil	Tidak Berhasil	
3	0°	150	√		Lampiran A
6	30° kanan	150	√		Lampiran A
9	30° kiri	150	√		Lampiran A
12	60° kanan	150		√	Lampiran A
15	60° kiri	150		√	Lampiran A

Dari hasil pengujian, dapat disimpulkan bahwa di skenario sejajar, sudut pengambilan gambar sangat mempengaruhi tingkat keberhasilan deteksi. Semakin besar sudutnya, maka akan semakin besar kemungkinan gagal dalam mendeteksi pelanggaran. Keterangan gambar dapat dilihat pada *link* berikut : <https://bit.ly/LampiranJurnalDR>

4. Kesimpulan

Berdasarkan hasil pengujian dan analisis yang dilakukan pada sistem ini, maka dapat ditarik beberapa kesimpulan sebagai berikut :

1. Hasil *training* terbaik yang didapatkan adalah dengan rasio 90%:10% dan nilai *max batches* 2000 dan *learning rate* 0.001 dengan nilai mAP 54.11%.
2. Hasil akurasi pengujian sistem deteksi pelanggaran *social distancing* adalah 73% dengan jarak optimal pendeteksian adalah 12 m dengan sudut pengambilan 0° dan 30° kiri.
3. Sistem pendeteksi pelanggaran *social distancing* menggunakan algoritma *You Only Look Once* (YOLO) dapat mendeteksi secara *real-time* dengan baik.

4. Sistem deteksi dapat berjalan secara optimal dan akurat melalui *website* yang dibuat dengan akurasi pengujian fungsionalitas *website* 100%.

REFERENSI

- [1] Centers for Disease Control and Prevention, "Prevent Getting Sick", *cdc.gov*, Available: cdc.gov/coronavirus/2019-ncov/prevent-getting-sick/. [Diakses 20 Juni 2021, 20.22 WIB].
- [2] Yew Cheong Hou, Mohd Zafri Baharuddin, Salman Yussof, Sumayyah Dzulkifly, "Social Distancing Detection with Deep Learning Model", *8th International Conference on Information Technology and Multimedia (ICIMU)*, 2020.
- [3] Pulung Nurtantio, Andono, T.Sutojo, Muljono. 2017. *Pengolahan Citra Digital*. Yogyakarta: ANDI.
- [4] Xiaoyue Jiang, Abdenour Hadid, Yanwei Pang, Eric Granger, Xiaoyi Feng, "Deep Learning in Object Detection and Recognition", pp. 1-3, 2019.
- [5] Joseph Redmond, "You Only Look Once: Unified, Real-time Object Detection", proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition, pp.1-10, 2016.
- [6] S. J. Pan and Q. Yang, "A Survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol.22, no.10, pp. 1345-1359, 2009.
- [7] M. Nishom, "Perbandingan Akurasi Euclidean Distance, Minkowski Distance, dan Manhattan Distance pada Algoritma K-Means Clustering berbasis Chi-Square", *Jurnal Informatik: Jurnal Pengembangan IT (JPIT)*, Vol.04, No.01, Januari 2019, pp. 20-21, 2019.