

PENDETEKSIAN JUMLAH PENUMPANG YANG MASUK BERDASARKAN CCTV PADA PINTU BUS DENGAN METODE YOLO

Vincent Marcellino¹, Vinny Christanti Mawardi², Novario Jaya Pradana³

^{1,2,3}) Teknik Informatika, FTI, Universitas Tarumanagara
Jl. Letjen S Parman no 1, Jakarta 11440 Indonesia

email : vincent.535180047@stu.untar.ac.id, vinym@fti.untar.ac.id, novariojp@fti.untar.ac.id

ABSTRACT

Salah satu cara untuk mengurangi kemacetan pada kota besar adalah dengan mengubah pola pikir masyarakat untuk lebih menggunakan kendaraan umum, kendaraan umum bus merupakan salah bentuk dari kendaraan umum. Namun kelebihan penumpang pada kendaraan umum merupakan permasalahan yang dapat ditemukan. Perancangan pendeteksian jumlah penumpang ini bertujuan untuk membantu melakukan perhitungan jumlah penumpang dari kendaraan umum menggunakan kamera, guna mendeteksi jumlah penumpang dalam kendaraan umum. Perancangan ini menggunakan algoritma YOLO (*You Only Look Once*), algoritma ini digunakan karena memiliki performa pendeteksian yang cepat pada skenario pendeteksian secara *real-time*. Perancangan ini menggunakan data berupa gambar yang telah dipecah dari video untuk kemudian digunakan sebagai data latih, data uji, dan data validasi. Setelah melakukan proses pengujian dengan 50 data video untuk pintu masuk dan pintu keluar, hasil yang didapatkan berupa 82% untuk tingkat akurasi perhitungan penumpang pada data video pintu masuk dan 72% untuk tingkat akurasi perhitungan penumpang pada data video pintu keluar.

Key words

Kamera, Pendeteksian Jumlah Penumpang, *You Only Look Once*.

1. Pendahuluan

Semakin banyak penumpang yang dapat masuk dalam suatu kendaraan umum maka transportasi umum tersebut akan lebih efisien untuk mengurangi kemacetan. Salah satu bentuk transportasi umum yang dinilai efisien adalah bus. Bus sendiri memiliki berbagai macam permasalahan di dalamnya salah satu permasalahan yang ditemukan yaitu sulitnya memonitor keluar masuk penumpang. Permasalahan yang telah di jabarkan juga telah menjadi permasalahan yang cukup besar bagi perusahaan bus[1].

Sistem perhitungan penumpang otomatis atau sistem pendeteksian jumlah penumpang menjadi salah satu opsi solusi yang dapat membantu mengurangi permasalahan yang berada pada transportasi umum bus yaitu kelebihan penumpang. Dengan sistem pendeteksian jumlah penumpang perusahaan transportasi dengan mudah memonitor keadaan kendaraan umum, sistem pendeteksian jumlah penumpang juga berfungsi untuk memberi informasi jika terdeteksi ada kelebihan penumpang. Rancangan ini akan menghasilkan aplikasi untuk menghitung berapa jumlah penumpang yang berada pada kendaraan umum yang difokuskan pada kendaraan umum bus.

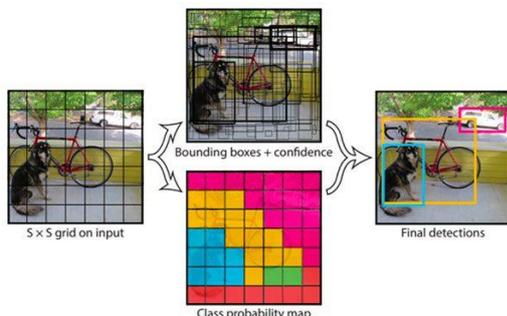
Aplikasi yang akan dibangun menggunakan algoritma YOLO (*You Only Look Once*), algoritma YOLO adalah algoritma *object detection* yang memiliki basis CNN (*Convolutional Neural Network*) untuk melakukan *image classification*. Algoritma YOLO dipilih berdasarkan kelebihan dan kekurangan algoritma YOLO tersebut, kekurangan algoritma YOLO dibandingkan algoritma lain adalah YOLO bukan algoritma yang memiliki tingkat akurasi tertinggi namun kelebihan dari YOLO adalah tingkat deteksi yang cepat dan memberikan keuntungan untuk melakukan *object detection* secara *real-time*. YOLO memiliki tingkat deteksi yang cepat dihasilkan oleh algoritma yang hanya melakukan proses deteksi sebanyak 1 kali, dengan memecah data menjadi beberapa bagian YOLO melakukan deteksi secara efisien dan tidak melakukan perulangan. Pendeteksian yang dilakukan akan mendeteksi lewat kepala[2].

2. Sistem Persamaan Linier

2.1 YOLO

YOLO adalah singkatan dari *You Only Look Once* merupakan suatu algoritma yang digunakan untuk melakukan pendeteksian objek yang dikembangkan oleh Joseph Redmon di tahun 2016[3]. YOLO memiliki dasar yaitu dari CNN namun yang membedakan kedua

algoritma adalah YOLO hanya melakukan CNN sebanyak 1 kali sehingga membuat YOLO memiliki kecepatan deteksi yang baik. YOLO membagi gambar *input* menjadi *grid* berukuran $S \times S$, S adalah sel yang berada di dalam *grid*[4].



Gambar 1 Ilustrasi Algoritma YOLO

Algoritma ini seperti dalam ilustrasi di atas akan melakukan *convolutional network* kemudian akan menghasilkan *boudingbox* beserta *confidence* dan probabilitas *class* dari setiap sel. Setelah itu YOLO akan memberikan hasil yaitu *boudingbox* untuk setiap objek yang terdeteksi. Untuk memperoleh *boudingbox* tersebut YOLO akan melakukan *convolutional network* dengan ukuran $S * S * B * (5 + C)$. Seperti yang telah dijelaskan di atas S mewakili sel, B mewakili berapa *boudingbox* yang akan di prediksi pada 1 *grid* dan C sendiri mewakili berapa banyak *class* yang akan di deteksi. Angka 5 akan mewakili letak dari *boudingbox* dan akan memiliki nilai x , y , w , h dan *confidence*. x mewakili letak x dari *boudingbox*, y mewakili letak y dari *boudingbox*, w mewakili nilai *width* dari *boudingbox*, h mewakili nilai *height* dari *boudingbox* dan *confidence* adalah nilai probabilitas dari objek yang terdeteksi pada *boudingbox* tersebut. Nilai x dan y akan memiliki nilai dalam rasio 0 sampai 1 sedangkan untuk nilai w dan h dapat memiliki nilai lebih dari 1 karena Panjang dan lebar dari suatu *boudingbox* kemungkinan akan lebih dari sel tersebut. Nilai *confidence* akan didefinisikan sebagai berikut $Pr(Class_i) * IoU_{pred}^{truth}$ [5].

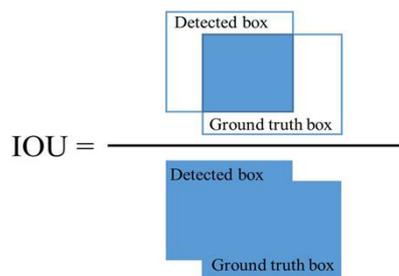
2.2 Intersection Over Union

Intersection Over Union atau biasa disingkat IoU adalah suatu metode untuk mengukur keakuratan dari suatu *boudingbox*[6]. IoU menjadi penting karena beberapa algoritma pengenalan objek atau *object detection* membutuhkan tingkat akurasi yang tinggi, meskipun ada beberapa algoritma yang memperbolehkan adanya beberapa kesalahan dalam melakukan penempatan pada *boudingbox* tersebut. IoU menghitung dengan cara *Area of Overlapping* (area pertemuan dari *boudingbox* prediksi dan *boudingbox* yang sebenarnya) dibagi dengan *Area of Union* (area dari persatuan dari kedua *boudingbox* tersebut).

Berikut adalah rumus dari IoU :

$$IoU = \frac{Area\ of\ Overlapping}{Area\ of\ Union} \dots\dots\dots(1)$$

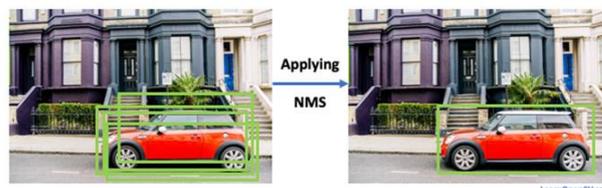
IoU sebesar 0.5 biasanya digunakan untuk melakukan evaluasi terhadap algoritma pendeteksian objek, namun semakin lama IoU sebesar 0.5 mulai dinilai tidak ideal dan kurang maka dari itu nilai IoU lebih besar dari 0.5 mulai digunakan.



Gambar 2 Ilustrasi IoU

2.3 Non-Max Suppression

Non Max Suppression adalah suatu metode yang digunakan untuk menentukan *boudingbox* yang paling baik dan mengurangi *boudingbox* yang terbentuk menjadi hanya satu *boudingbox*, dengan cara ini *boudingbox* yang menumpuk akan diolah menjadi 1 *boudingbox* saja[7].



Gambar 2 Ilustrasi Non-Max Suppression

Cara kerja dari metode ini adalah dengan mengambil prediksi yang memiliki nilai *confidence* yang maksimum dan menghilangkan prediksi yang lain dengan tingkat *confidence* yang rendah.

2.4 Mean Average Precision

Average Precision atau biasa di singkat AP adalah salah satu cara untuk menentukan evaluasi performa dari tiap model. Jika dijabarkan secara ringkas AP melakukan perhitungan untuk nilai rata-rata dari *precision* untuk setiap *recall* dan diukur dengan rentang 0 sampai dengan 1[8]. *Precision* sendiri menilai seberapa akurat suatu prediksi yang dilakukan[9].

$$Precision = \frac{TP}{TP + FP} \dots\dots\dots(2)$$

Recall sendiri menilai seberapa baik nilai positif yang dihasilkan suatu model.

$$Recall = \frac{TP}{TP + FN} \dots\dots\dots(3)$$

Setelah mengetahui Recall dan Precision, AP dinilai dengan rumus sebagai berikut.

$$AP = \sum_{i=0}^n [Recall_{i+1} - Recall_i] * P_{inter}(Recall_{i+1}) \dots\dots\dots(4)$$

$$mAP = \sum_{i=1}^n \frac{AP(i)}{N} x 100\% \dots\dots\dots(5)$$

Terakhir AP akan dirata-rata dengan AP lainnya agar menghasilkan Mean Average Precision.

3. Hasil Percobaan

Proses pengujian dilaksanakan untuk memastikan bahwa aplikasi yang telah di buat, memiliki keluaran yang diharapkan dan memeriksa apakah aplikasi berjalan sesuai dengan yang diharapkan. Pengujian dilakukan dengan model yang telah disiapkan dan dilatih untuk mendeteksi kepala manusia untuk menghitung penumpang kendaraan umum bus.

Pengujian pada model YOLO yang telah dibuat dilakukan untuk mengukur keefektivitasan model dalam mendeteksi kepala manusia yang mewakili penumpang yang ada dalam kendaraan umum bus. Model YOLO akan diuji dengan menggunakan dataset yang telah dipilih. Dataset yang sudah dipilih akan di bagi menjadi data latih, data validasi, dan data uji.

Tabel 1 Pengujian Model YOLO Pintu Masuk

Jenis Data	Jumlah Gambar	Terdeteksi		Tidak	Akurasi
		Benar	Salah	Terdeteksi	
Tidak Menggunakan Atribut	100	98	-	2	98%
Menggunakan Atribut	100	93	1	6	93%
	200	191	1	8	95,5%

Tabel 2 Pengujian Model YOLO Pintu Keluar

Jenis Data	Jumlah Gambar	Terdeteksi		Tidak	Akurasi
		Benar	Salah	Terdeteksi	
Tidak Menggunakan Atribut	100	93	2	5	93%
Menggunakan Atribut	100	85	11	4	85%
	200	178	17	5	89%

Pengujian pada model YOLO bertujuan untuk mengetahui seberapa akurat model dalam mendeteksi kepala manusia. Setelah melakukan pengujian maka didapatkan dari 200 data validasi menghasilkan persentase akurasi sebesar 97.5%. Setelah melakukan pengujian maka didapatkan dari 200 data validasi sehingga menghasilkan persentase akurasi sebesar 95.5%.

Pengujian pendeteksian jumlah penumpang pada pintu masuk dilakukan dengan 10 data video sebagai data validasi sehingga menghasilkan persentase akurasi 100%. Pengujian pendeteksian jumlah penumpang pada pintu keluar dilakukan dengan 10 data video sebagai data validasi sehingga menghasilkan persentase akurasi 90%.

Tabel 3 Pengujian Pendeteksian Jumlah Penumpang Pintu Masuk

Jenis Video	Jumlah Video	Durasi	Jumlah Video Terdeteksi		Akurasi
			Benar	Salah	
Kurang dari Sama Dengan 5 Orang	10	15 – 25 detik	10	0	100%
Lebih dari 5 Orang	10	15 – 25 detik	9	1	90%
Atribut Hitam	10	15 – 25 detik	9	1	90%
Atribut Selain Hitam	10	15 – 25 detik	7	3	70%
Kurang Cahaya	10	15 – 25 detik	6	4	60%
	50		41	9	82%

Tabel 4 Pengujian Pendeteksian Jumlah Penumpang Pintu Keluar

Jenis Video	Jumlah Video	Durasi	Jumlah Video Terdeteksi		Akurasi
			Benar	Salah	
Kurang dari Sama Dengan 5 Orang	10	15 – 25 detik	9	1	90%
Lebih dari 5 Orang	10	15 – 25 detik	10	0	100%
Atribut Hitam	10	15 – 25 detik	9	1	90%
Atribut Selain Hitam	10	15 – 25 detik	5	5	50%
Kurang Cahaya	10	15 – 25 detik	4	6	40%
	50		37	13	74%

Berdasarkan pengujian yang telah dilakukan baik pada model maupun pada pendeteksian jumlah penumpang, dapat disimpulkan bahwa hasil pendeteksian kepala manusia untuk mendeteksi kelebihan penumpang dengan dataset yang telah disiapkan cukup baik untuk mendeteksi kepala manusia dan hasil pendeteksian untuk pintu masuk dengan data testing didapatkan akurasi mAP 88.84% dan memiliki nilai IoU 71.30%, sedangkan hasil pendeteksian

untuk pintu keluar dengan data *testing* didapatkan akurasi mAP 87.32% dan memiliki nilai IoU 61.65%.

Dari hasil pengujian, dapat dilihat bahwa aplikasi kesulitan untuk mendeteksi kepala manusia jika keadaan video memiliki kekurangan cahaya dan kesulitan untuk mendeteksi jika penumpang menggunakan atribut pada kepala yang tidak berwarna gelap.

4. Kesimpulan

Kesimpulan yang dapat ditarik setelah melakukan pengujian terhadap aplikasi Pendeteksian Jumlah Penumpang Yang Masuk Berdasarkan CCTV Pada Pintu Bus Dengan Metode YOLO, sebagai berikut:

1. Aplikasi yang telah dibuat dapat menghitung jumlah penumpang dari data yang berupa gambar telah dimasukkan oleh pengguna untuk kondisi memiliki penumpang lebih dari 5 ataupun kurang dari 5 orang dan kondisi penumpang dengan atribut berwarna hitam untuk video pintu masuk maupun pintu keluar. Total akurasi sebesar 82% untuk pintu masuk dan 76% untuk pintu keluar.
2. Aplikasi mampu mendeteksi kepala manusia dari data video hasil CCTV cukup baik dengan akurasi sebesar 95,5% untuk pintu masuk dan 82% untuk pintu keluar.
3. Berdasarkan hasil pengujian video yang berhasil dideteksi adalah video yang memiliki cahaya yang baik dan penumpang dengan warna rambut atau atribut berwarna gelap.
4. Aplikasi kurang akurat untuk mendeteksi penumpang memiliki rambut berwarna terang ataupun menggunakan atribut berwarna terang dan aplikasi kurang akurat saat mendeteksi penumpang dengan data video yang kurang cahaya.

REFERENSI

- [1] Farhan Destian, Kajian Efektivitas Bus Trans Metro Bandung Trayek Leuwipanjang – Antapani, <http://eprints.itenas.ac.id/454/>, 28 Maret 2022
- [2] Fathur Rahman Haikal, Analisis Perbandingan Akurasi Dan Performa Algoritma Deteksi Objek Pada Yolo V3 Dengan Singl Shot Multibox Detector (SSD), https://repository.pnj.ac.id/id/eprint/1239/1/4617010028_SKRIPSI_FATHUR%20RAHMAN%20HAIKAL%201-5.pdf, 15 Maret 2022
- [3] Liunanda, Christopher Nathanel, Rostianingsih, Silvia dan Purbowo, Anita Nathania, “Implementasi Algoritma YOLO pada Aplikasi Pendeteksi Senjata Tajam di Android”, Jurnal Infra, Vol 8, No 2 (2020), h. 2
- [4] Nazilly, Mada Lazuardi, Rahmat, Basuki dan Puspaningrum, Eva Yulia, “Implementasi Algoritma Yolo (You Only Look Once) Untuk Deteksi Api”, Jurnal Teknologi Informasi dan Ilmu Komputer, Vol 8, No 4 (2020), h. 84
- [5] Nazilly, Mada Lazuardi, Rahmat, Basuki dan Puspaningrum, Eva Yulia, “Implementasi Algoritma Yolo (You Only Look Once) Untuk Deteksi Api”, Jurnal Teknologi Informasi dan Ilmu Komputer, Vol 8, No 4 (2020), h.84

- [6] Pramestya, Ravy Hayu, Deteksi Dan Klasifikasi Kerusakan Jalan Aspal Menggunakan Metode Yolo Berbasis Citra Digital, https://repository.its.ac.id/59044/1/06111650010019-Master_Thesis.pdf, 30 Maret 2022
- [7] Prakash, Jatin, Non Maximum Suppression: Theory and Implementation in PyTorch, [https://learnopencv.com/non-maximum-suppression-theory-and-implementation-in-pytorch/#:~:text=Non%20Maximum%20Suppression%20\(NMS\)%20is,arrive%20at%20the%20desired%20results.,](https://learnopencv.com/non-maximum-suppression-theory-and-implementation-in-pytorch/#:~:text=Non%20Maximum%20Suppression%20(NMS)%20is,arrive%20at%20the%20desired%20results.,) 28 February 2022
- [8] Zulkiflie, Muhammad Amirullah, Implementasi Algoritma Object Detection Yolov4 Dan Euclidean Distance Dalam Mendeteksi Pelanggaran Social Distancing, <http://repository.unhas.ac.id/id/eprint/12039/>, 27 Maret 2022
- [9] Fauzy, Ahmad Rizky, Implementasi Metode Region Convolutional Neural Network Dalam Mendiagnosa Anomali Pneumonia Pada Foto Thorax, https://elibrary.unikom.ac.id/id/eprint/1158/10/14.%20UNIKOM_AHMAD%20RIZKY%20FAUZY_BAB%204.pdf, 20 Maret 2022

Vincent Marcellino, mahasiswa S1, program studi Teknik Informatika Universitas Tarumanagara.