

OBJECT COUNTING PADA DATA VIDEO

**Rudy Herteno¹⁾, M. Reza Faisal²⁾, Radityo Adi Nugroho³⁾,
Friska Abadi⁴⁾, Rahmat Ramadhani⁵⁾.**

^{1),2),3),4),5)} Ilmu komputer Universitas Lambung Mangkurat
Jl.Ahmad Yani Km36, Banjarbaru
rudy.herteno@ulm.ac.id

Abstract

One object counting implementation is counting the number of road users from video data sources obtained from CCTV streaming. Video processing on CCTV is usually done on the server side by sending video data. If the need is only to determine the density of traffic, then the method is considered too expensive to be implemented because of the cost of internet connection and bandwidth that must be spent. The solution is to use a small computing device that can process the video first, and the calculation results are sent to the server regularly. In this study, a comparison between the Tensorflow Object Counting learning algorithm and the MOG2 Background Subtractor image processing algorithm with the aim to determine the accuracy of the calculation. The result is known that better accuracy is given by the MOG2 Background Subtractor technique and also the process is carried out using only a small percentage of the amount of memory and processor compared to the Tensorflow Object Counting technique. MOG2 Background Subtractor technique is expected to be used on devices that have small data sources

Keywords : Object Counting, Tensorflow, MOG2 Background Subtractor.

Abstrak

Salah satu implementasi object counting adalah menghitung jumlah pengguna jalan dari sumber data video yang didapat dari streaming CCTV. Pemrosesan video pada CCTV biasanya dilakukan disisi server dengan mengirimkan data video. Jika keperluannya hanya untuk mengetahui kepadatan lalu lintas, maka cara tersebut dinilai terlalu mahal untuk diimplementasikan karena biaya koneksi internet dan bandwidth yang harus dikeluarkan. Pemecahannya adalah menggunakan perangkat komputasi kecil yang dapat memproses video tersebut terlebih dahulu, dan hasil perhitungannya dikirimkan ke server secara berkala. Pada penelitian ini dilakukan perbandingan antara algoritma pembelajaran Tensorflow Object Counting dan algoritma image processing MOG2 Background Subtractor dengan tujuan untuk mengetahui akurasi penghitungan. Hasilnya diketahui akurasi yang lebih baik diberikan oleh teknik MOG2 Background Subtractor dan juga proses yang dilakukan hanya menggunakan prosentase jumlah memori dan prosessor yang kecil dibandingkan teknik Tensorflow Object Counting. Sehingga teknik MOG2 Background Subtractor ini diharapkan dapat digunakan pada perangkat yang memiliki sumber data kecil.

Kata kunci : Object Counting, Tensorflow, MOG2 Background Subtractor.

1. PENDAHULUAN

Pemantauan (monitoring) atau pengawasan (surveillance) telah dilakukan manusia untuk mengetahui kondisi dan keadaan suatu tempat pada waktu tertentu. Untuk melakukan hal tersebut biasanya dengan menempatkan orang pada lokasi-lokasi yang ingin diketahui kondisinya dalam jangka waktu tertentu. Namun dengan perkembangan teknologi di bidang komunikasi maka hal tersebut dapat dilakukan dengan menggunakan kamera seperti Closed-Circuit Television (CCTV) yang dikenal juga sebagai perangkat video pengawas (video surveillance) (Kruegle, 2006). Perangkat ini mengirimkan sinyal berupa data video ke tempat tertentu sehingga hanya dapat diakses oleh perangkat tertentu saja. Hal ini berbeda dengan broadcast televisi pada umumnya dimana sinyal dikirimkan secara terbuka sehingga dapat diakses oleh siapa saja.

Dengan menggunakan perangkat kamera tersebut tetap diperlukan manusia untuk melihat video yang dikirimkan secara streaming dalam jangka waktu tertentu. Hal itu membuat para peneliti melakukan riset untuk membantu mengurangi peran manusia dalam melakukan monitoring atau pengawasan. Sebagai contoh untuk memantau tempat parkir yang kosong secara otomatis dengan menggunakan algoritma image processing (Yusnita, Fariza, & Norazwinawati, 2012). Riset yang lain deteksi kebakaran dan asap dari data video secara real-time dengan metode foreground image accumulation dan optical flow technique (Yu, Mei, & Zhang, 2013). Menghitung mendeteksi buah citrus dan menghitung jumlahnya dengan menggunakan beberapa algoritma image processing seperti convert RGB image to HSV, thresholding, orange color detection, noise removal, watershed segmentation dan kemudian melakukan penghitungan (Dorj, Lee, & Yun, 2017). Contoh-contoh ini dikelompokkan menjadi teknik yang menggunakan algoritma image processing.

Kelompok teknik lain adalah teknik yang menggunakan model hasil pembelajaran algoritma machine learning. Riset berikut menggunakan deep learning untuk membangun sistem peringatan darurat dari data video CCTV (Kang & Choo, 2016). Riset lain adalah mendeteksi kemacetan lalu-lintas dari video CCTV dengan menggunakan Convolutional Neural Network (Kurniawan, Dewa, & Afiahayati, 2018).

Namun jika ingin membuat aplikasi atau sistem untuk menghitung jumlah kendaraan pada jalan raya dengan menggunakan video dari CCTV umumnya video akan dikirim ke komputer dimana manusia akan melihat video tersebut pada layar komputer atau video akan diproses oleh kedua kelompok teknik yang disebutkan di atas untuk menghitung kendaraan. Kedua kelompok teknik di atas dapat digunakan pada komputer karena resource komputer yang memadai. Tetapi cara seperti ini memerlukan koneksi internet yang bagus dan menggunakan data yang besar untuk pengiriman data video. Agar penggunaan data dapat dikurangi maka proses penghitungan kendaraan dapat dilakukan pada perangkat kecil yang dekat CCTV, kemudian perangkat tersebut melakukan proses penghitungan kendaraan dan hasilnya akan dikirim secara berkala ke server. Perangkat kecil seperti ini telah umum digunakan pada era industri 4.0 yang sering dikenal dengan istilah Internet of Things. Perangkat kecil ini adalah komputer kecil dan minimal dengan

memori dan prosesor yang jauh lebih kecil dibanding komputer yang umumnya kita gunakan.

Penelitian ini dilakukan untuk menguji kelompok teknik yang mana yang cocok digunakan pada perangkat kecil tersebut. Pengujian dilakukan dengan menggunakan data video yang sama. Parameter yang dibandingkan adalah akurasi dari teknik yang digunakan untuk mendapatkan akurasi mana yang lebih baik. Kemudian dilakukan pencatatan pemakaian memory dan prosesor serta waktu komputasi.

2. METODOLOGI PENELITIAN

2.1 Dataset

Dataset pada percobaan ini adalah video yang direkam dari sumber streaming CCTV pada website resmi Banjarbaru Kota yaitu <http://cctv.banjarbarukota.go.id/> dengan lokasi-lokasi sebagai berikut:

Tabel 1. Dataset video CCTV lalu-linta di Banjarbaru.

No	Lokasi	Durasi (Menit)	Jumlah Mobil
1.	CCTV Depan Polres Banjarbaru	01:26	49
2.	CCTV Batas Kota	01:26	48
3.	CCTV Pasar Bauntung	01:09	49
4.	CCTV Universitas Lambung Mangkurat Banjarbaru	01:14	25
5.	CCTV Q Mall	01:09	53

File video tersebut juga diambil dengan sudut yang berbeda sehingga setiap video yang digunakan memiliki sudut pandang terhadap object berbeda pula. Berikut adalah gambar dari masing-masing tempat yang telah disebutkan di atas.



Gambar 1. CCTV Depan Polres Banjarbaru.

Pada penelitian ini obyek yang akan dihitung adalah mobil. Obyek-obyek selain mobil akan tidak akan dihitung.

2.1 Prosedur Penelitian

Pada penelitian ini dilakukan beberapa tahapan yaitu:

- a. Pengambilan data video.
Pengambil data video sebagai dataset pada penelitian ini dilakukan dengan cara mengakses website resmi Banjarbaru Kota yaitu <http://cctv.banjarbarukota.go.id/>. Kemudian melakukan perekaman video selama kurang lebih 1 menit. Selanjutnya peneliti melakukan penghitungan jumlah mobil pada setiap video dan membiarkan obyek lainnya.
- b. Penghitungan obyek dengan Tensorflow Object Counting.
Tensorflow Object Counting API merupakan salah satu algoritma yang dapat menghitung jumlah object pada suatu gambar atau video. Dalam menjalankan Tensorflow Object Counting API diperlukan sebuah model pengenalan object yang telah ditraining menggunakan citra object yang ingin dikenali. Pada penelitian ini, model yang digunakan adalah SSD Mobilenet v1. Selanjutnya setiap file video yang telah dijelaskan sebelumnya dijadikan sebagai input video pada Tensorflow Object Counting API dan dieksekusi per video. Secara otomatis tensorflow akan men-generate satu buah file output yang berisi video hasil penghitungan obyek. Untuk setiap file video yang diproses dilakukan pencatatan jumlah mobil yang berhasil dikenali, waktu proses dan penggunaan memori.
- c. Penghitungan obyek dengan Background Subtractor MOG2.
Object counting pada video menggunakan background subtraction method memanfaatkan perubahan foreground yang diekstrak untuk proses perhitungan. Proses pendeteksian object yang bergerak dengan metode background subtraction berdasarkan pada perbedaan antara background referensi dengan foreground. Foreground pada penelitian ini adalah object mobil. Object mobil dikatakan foreground dikarenakan object tersebut memiliki pergerakan dinamis. Berbeda dengan Tensorflow Object Counting API yang outputnya berupa file video baru, background subtraction method secara realtime menampilkan output ketika video dijalankan. Perhitungan object counting akan ditampilkan secara langsung dilayar ketika ada object mobil yang terdeteksi. Untuk setiap file video yang diproses dilakukan pencatatan jumlah mobil yang berhasil dikenali, waktu proses dan penggunaan memori.
- d. Perhitungan dan perbandingan kinerja algoritma.
Hasil penghitungan obyek mobil dari setiap algoritma dibandingkan dengan jumlah obyek dari setiap video. Hasil akan dihitung dengan menggunakan rumus MAE dan MSE.

3. HASIL DAN PEMBAHASAN


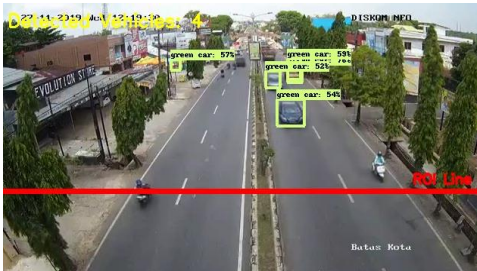

3.1 Hasil

Sesuai dengan tahapan penelitian, maka langkah selanjutnya adalah melakukan penghitungan obyek mobil pada dataset video dengan menggunakan kedua teknik atau algoritma yang telah disebutkan pada bab sebelumnya.

3.1.1 Tensorflow Object Counting

Berikut adalah hasil dari penghitungan obyek mobil dari 5 video tersebut.

Tabel 2. Hasil penghitungan obyek mobil dengan Tensorflow Object Counting.

No	Hasil Pengukuran																										
1	 <p style="text-align: center;">CCTV Depan Polres Banjarbaru</p> <table border="1"> <thead> <tr> <th>Percobaan</th> <th>Mobil Terdeteksi</th> <th>Execution Time</th> <th>Memory Usage</th> </tr> </thead> <tbody> <tr><td>1</td><td>8</td><td>221.8210</td><td>108.500 MiB</td></tr> <tr><td>2</td><td>8</td><td>286.4374</td><td>122.836 MiB</td></tr> <tr><td>3</td><td>8</td><td>289.4634</td><td>124.805 MiB</td></tr> <tr><td>4</td><td>8</td><td>218.2424</td><td>329.383 MiB</td></tr> <tr><td>5</td><td>8</td><td>210.0331</td><td>222.594 MiB</td></tr> </tbody> </table>			Percobaan	Mobil Terdeteksi	Execution Time	Memory Usage	1	8	221.8210	108.500 MiB	2	8	286.4374	122.836 MiB	3	8	289.4634	124.805 MiB	4	8	218.2424	329.383 MiB	5	8	210.0331	222.594 MiB
Percobaan	Mobil Terdeteksi	Execution Time	Memory Usage																								
1	8	221.8210	108.500 MiB																								
2	8	286.4374	122.836 MiB																								
3	8	289.4634	124.805 MiB																								
4	8	218.2424	329.383 MiB																								
5	8	210.0331	222.594 MiB																								
2	 <p style="text-align: center;">CCTV Batas Kota</p> <table border="1"> <thead> <tr> <th>Percobaan</th> <th>Mobil Terdeteksi</th> <th>Execution Time</th> <th>Memory Usage</th> </tr> </thead> <tbody> <tr><td>1</td><td>13</td><td>180.3984</td><td>176.203 MiB</td></tr> <tr><td>2</td><td>13</td><td>185.2826</td><td>238.934 MiB</td></tr> <tr><td>3</td><td>13</td><td>183.6157</td><td>169.438 MiB</td></tr> <tr><td>4</td><td>13</td><td>181.7587</td><td>336.152 MiB</td></tr> <tr><td>5</td><td>13</td><td>181.9414</td><td>283.355 MiB</td></tr> </tbody> </table>			Percobaan	Mobil Terdeteksi	Execution Time	Memory Usage	1	13	180.3984	176.203 MiB	2	13	185.2826	238.934 MiB	3	13	183.6157	169.438 MiB	4	13	181.7587	336.152 MiB	5	13	181.9414	283.355 MiB
Percobaan	Mobil Terdeteksi	Execution Time	Memory Usage																								
1	13	180.3984	176.203 MiB																								
2	13	185.2826	238.934 MiB																								
3	13	183.6157	169.438 MiB																								
4	13	181.7587	336.152 MiB																								
5	13	181.9414	283.355 MiB																								
3	 <p style="text-align: center;">CCTV Pasar Bauntung</p> <table border="1"> <thead> <tr> <th>Percobaan</th> <th>Mobil Terdeteksi</th> <th>Execution Time</th> <th>Memory Usage</th> </tr> </thead> <tbody> <tr><td>1</td><td>27</td><td>214.7075</td><td>153.445 MiB</td></tr> <tr><td>2</td><td>27</td><td>208.5792</td><td>132.516 MiB</td></tr> <tr><td>3</td><td>27</td><td>211.6707</td><td>129.230 MiB</td></tr> <tr><td>4</td><td>27</td><td>211.7493</td><td>117.043 MiB</td></tr> <tr><td>5</td><td>27</td><td>211.8280</td><td>128.891 MiB</td></tr> </tbody> </table>			Percobaan	Mobil Terdeteksi	Execution Time	Memory Usage	1	27	214.7075	153.445 MiB	2	27	208.5792	132.516 MiB	3	27	211.6707	129.230 MiB	4	27	211.7493	117.043 MiB	5	27	211.8280	128.891 MiB
Percobaan	Mobil Terdeteksi	Execution Time	Memory Usage																								
1	27	214.7075	153.445 MiB																								
2	27	208.5792	132.516 MiB																								
3	27	211.6707	129.230 MiB																								
4	27	211.7493	117.043 MiB																								
5	27	211.8280	128.891 MiB																								

No Hasil Pengukuran

4



CCTV Qmall

Percobaan	Mobil Terdeteksi	Execution Time	Memory Usage
1	14	190.7993	109.953 MiB
2	14	181.8311	172.055 MiB
3	14	185.5157	121.645 MiB
4	14	180.5242	285.578 MiB
5	14	179.2511	356.035 MiB

5



CCTV ULM

Percobaan	Mobil Terdeteksi	Execution Time	Memory Usage
1	1	175.9112	233.562 MiB
2	1	173.1342	356.902 MiB
3	1	173.8985	355.871 MiB
4	1	172.3666	335.559 MiB
5	1	186.3937	127.051 MiB

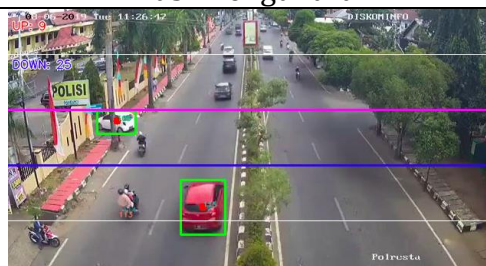
3.1.2 Background Substraction MOG2

Berikut adalah hasil dari penghitungan obyek mobil dari 5 video tersebut.

Tabel 3. Hasil penghitungan obyek mobil dengan Background Substraction MOG2.

No Hasil Pengukuran

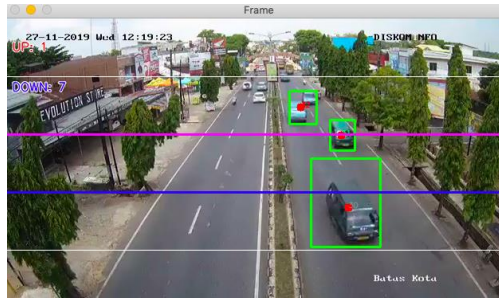
1



CCTV Depan Polres Banjarbaru

No	Hasil Pengukuran			
	Percobaan	Mobil Terdeteksi	Execution Time	Memory Usage
	1	40	77.1702	110.129 MiB
	2	40	78.3529	111.004 MiB
	3	40	77.4529	111.684 MiB
	4	40	78.0650	106.277 MiB
	5	40	77.5878	105.566 MiB

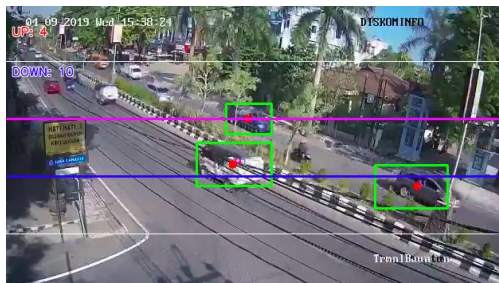
2



CCTV Batas Kota

Percobaan	Mobil Terdeteksi	Execution Time	Memory Usage
1	49	64.5720	91.797 MiB
2	49	64.9877	95.277 MiB
3	49	64.8797	111.684 MiB
4	49	65.0772	91.293 MiB
5	49	65.8996	83.078 MiB

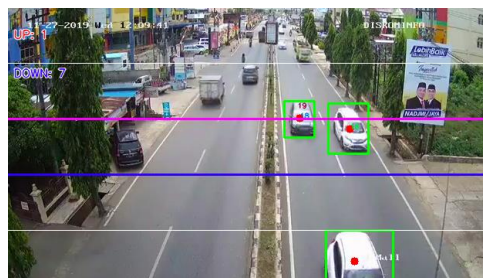
3



CCTV Pasar Bauntung

Percobaan	Mobil Terdeteksi	Execution Time	Memory Usage
1	75	89.1255	95.555 MiB
2	75	90.1921	105.219 MiB
3	75	91.2381	98.965 MiB
4	75	90.9882	92.723 MiB
5	75	89.4388	92.742 MiB

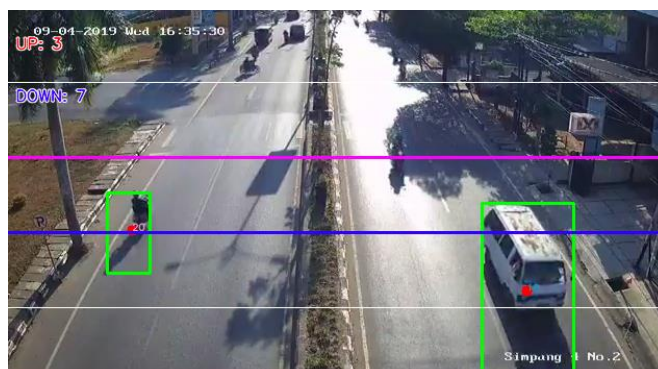
4



CCTV Qmall

No	Hasil Pengukuran			
	Percobaan	Mobil Terdeteksi	Execution Time	Memory Usage
	1	60	68.0856	120.602 MiB
	2	60	67.5644	120.602 MiB
	3	60	67.7372	98.516 MiB
	4	60	67.1039	117.887 MiB
	5	60	89.4388	72.8163 MiB

5



CCTV ULM

Percobaan	Mobil Terdeteksi	Execution Time	Memory Usage
1	79	71.5583	70.852 MiB
2	79	69.8537	97.461 MiB
3	79	67.9696	93.195 MiB
4	79	68.1910	109.457 MiB
5	79	69.1507	110.324 MiB

3.2 Pembahasan

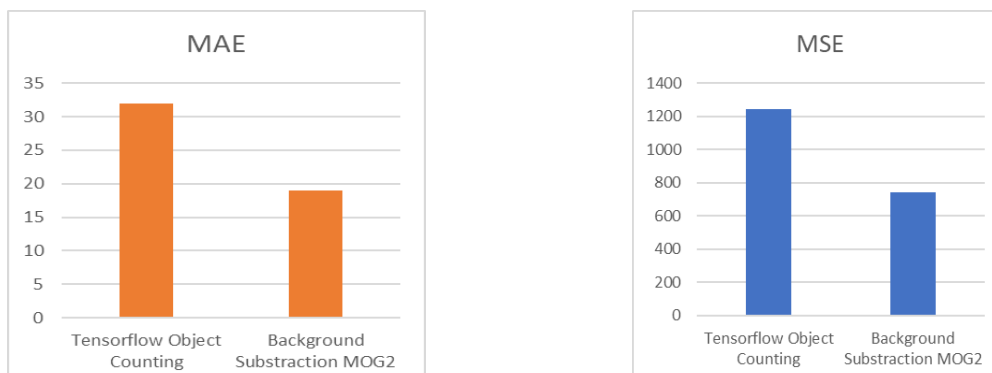
Dari hasil di atas maka dibuat rangkuman kinerja prediksi atau tingkat error prediksi, waktu dan penggunaan memori yang digunakan untuk proses. Untuk perbandingan kinerja prediksi atau tingkat error dapat dilihat pada Tabel 4.

Tabel 4. Perbandingan nilai rata-rata penghitungan obyek mobil.

No	Metode	Lokasi	Rata-Rata Prediksi	Jumlah Mobil
1	Tensorflow Object Counting	CCTV Depan Polres Banjarbaru	8	49
		CCTV Batas Kota	13	48
		CCTV Pasar Bauntung	27	49
		CCTV Qmall	14	53
		CCTV ULM	1	25
2	Background Substraction MOG2	CCTV Depan Polres Banjarbaru	40	49
		CCTV Batas Kota	49	48
		CCTV Pasar Bauntung	75	49
		CCTV Qmall	60	53
		CCTV ULM	79	25

Dari data tersebut di atas maka dapat dihitung MAE dan MSE dari masing-masing teknik atau algoritma yaitu:

- a. Algoritma Tensorflow Object Counting didapat MAE=32 dan MSE=1243.
- b. Algoritma Background Substraction MOG2 didapat MAE=19 dan MSE=744,6.



Gambar 2. Perbandingan MAE dan MSE antara algoritma Tensorflow Object Counting dan Background Substraction MOG2.

Dari Gambar 2 dapat dilihat jika algoritma Background Substraction MOG2 memiliki error atau kesalahan lebih kecil dibandingkan algoritma Tensorflow Object Counting. Algoritma Background Substraction MOG2 lebih dapat mengenali object yang bergerak. Tetapi tidak bisa membedakan apakah obyek yang bergerak itu mobil atau bukan. Oleh karena itu pada lokasi-lokasi yang lebih banyak obyek selain mobil seperti sepeda motor yaitu di CCTV ULM, algoritma ini melakukan banyak kesalahan.

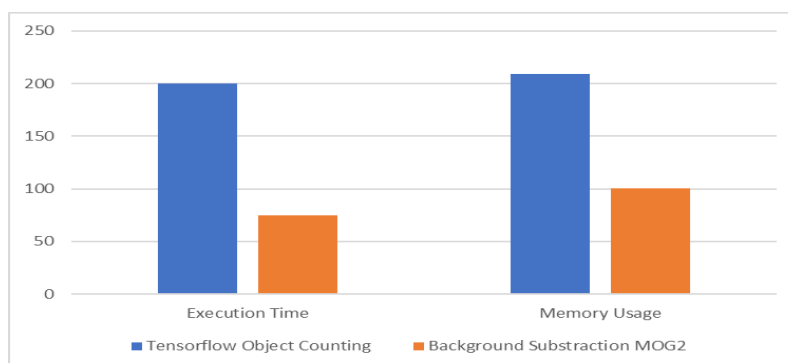
Pada Tabel 5 dan Tabel 6 adalah rata-rata waktu pemrosesan dan penggunaan memori dari kedua algoritma.

Tabel 5. Rata-rata waktu proses dan penggunaan memori oleh algoritma Tensorflow Object Counting.

No	Lokasi	Rata-Rata Execution Time	Rata-Rata Memory Usage
	CCTV Depan Polres Banjarbaru	245.1995	181.6236
	CCTV Batas Kota	182.5994	240.8164
	CCTV Pasar Bauntung	211.7069	132.225
	CCTV Qmall	183.5843	209.0532
	CCTV ULM	176.3408	281.789
	Rata-Rata	199.8862	209.1014

Tabel 6. Rata-rata waktu proses dan penggunaan memori oleh Background Substraction MOG2.

No	Lokasi	Rata-Rata Execution Time	Rata-Rata Memory Usage
	CCTV Depan Polres Banjarbaru	77.72576	108.932
	CCTV Batas Kota	65.08324	94.6258
	CCTV Pasar Bauntung	90.19654	97.0408
	CCTV Qmall	71.98598	106.0847
	CCTV ULM	69.34466	96.2578
	Rata-Rata	74.86724	100.5882



Gambar 3. Perbandingan waktu proses dan penggunaan memori.

Dari hasil perbandingan pada Gambar 3 maka dapat dilihat bahwa algoritma Background Substraction MOG2 menggunakan memori yang lebih kecil dan waktu proses yang lebih singkat dibanding algoritma Tensorflow Object Counting. Hal ini memberikan informasi bahwa algoritma Background Substraction MOG2 memungkinkan untuk digunakan pada perangkat kecil. Selain itu kinerja prediksi juga lebih bagus karena memiliki nilai error yang lebih kecil.

4. KESIMPULAN

Berikut yang dapat diambil kesimpulan pada penelitian yaitu sebagai berikut:

- Tingkat kesalahan prediksi dari perhitungan oleh teknik Tensorflow Object Counting adalah MAE=32 dan MSE=1243 dengan rata-rata waktu proses 199.8862 ms dan penggunaan memori 209.1014 MiB.
- Tingkat kesalahan prediksi dari perhitungan oleh teknik Background Subtractor MOG2 adalah MAE=19 dan MSE=744,6 dengan rata-rata waktu proses 74.86724 ms dan penggunaan memori 100.5882 MiB.
- Algoritma Background Subtractor MOG2 cocok digunakan pada perangkat kecil karena memiliki tingkat kesalahan yang kecil selain itu didukung dengan penggunaan memori dan waktu proses yang kecil juga.

DAFTAR PUSTAKA

- [1] Citrus, Fruit size, Machine vision, Watershed transform, & Yield mapping. (2013). Citrus Yield Mapping System in Natural Outdoor Scenes using the Watershed Transform. <https://doi.org/10.13031/2013.20853>
- [2] Dorj, U. O., Lee, M., & Yun, S. seok. (2017). An yield estimation in citrus orchards via fruit detection and counting using image processing. *Computers and Electronics in Agriculture*. <https://doi.org/10.1016/j.compag.2017.05.019>
- [3] Ian Goodfellow, Yoshua Bengio, A. C. (2015). Deep Learning Book. *Deep Learning*. <https://doi.org/10.1016/B978-0-12-391420-0.09987-X>
- [4] Kang, B., & Choo, H. (2016). A deep-learning-based emergency alert system. *ICT Express*. <https://doi.org/10.1016/j.icte.2016.05.001>
- [5] Kruegle, H. (2006). *CCTV Surveillance: Analog and Digital Video Practices and Technology*. *CCTV Surveillance*.
- [6] Kurniawan, J., Dewa, C. K., & Afiahayati. (2018). Traffic Congestion Detection: Learning from CCTV Monitoring Images using Convolutional Neural Network. In *Procedia Computer Science*. <https://doi.org/10.1016/j.procs.2018.10.530>
- [7] Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*. <https://doi.org/10.1038/nature14539>
- [8] Lehmann, E. L., & Casella, G. (1998). *Theory of Point Estimation , Second Edition Springer Texts in Statistics. Design*. <https://doi.org/10.2307/1270597>
- [9] Miao, Y., Han, J., Gao, Y., & Zhang, B. (2019). ST-CNN: Spatial-Temporal Convolutional Neural Network for crowd counting in videos. *Pattern Recognition Letters*. <https://doi.org/10.1016/j.patrec.2019.04.012>
- [10] Palaniappan Annamalai, & Won Suk Lee. (2013). Citrus Yield Mapping System Using Machine Vision. <https://doi.org/10.13031/2013.13701>
- [11] Schmidhuber, J. (2015). Deep Learning in neural networks: An overview. *Neural Networks*. <https://doi.org/10.1016/j.neunet.2014.09.003>
- [12] Willmott, C. J., & Matsuura, K. (2005). Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate Research*. <https://doi.org/10.3354/cr030079>
- [13] Yu, C., Mei, Z., & Zhang, X. (2013). A real-time video fire flame and smoke detection algorithm. In *Procedia Engineering*. <https://doi.org/10.1016/j.proeng.2013.08.140>
- [14] Yusnita, R., Fariza, N., & Norazwinawati, B. (2012). Intelligent Parking Space Detection System Based on Image Processing. *International Journal of Innovation, Management and Technology*.
- [15] Zivkovic, Z. (2004). Improved adaptive Gaussian mixture model for background subtraction. In *Proceedings - International Conference on Pattern Recognition*.
- [16] Zivkovic, Z., & Van Der Heijden, F. (2006). Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognition Letters*. <https://doi.org/10.1016/j.patrec.2005.11.005>