

Deteksi Sampah pada Real-time Video Menggunakan Metode Faster R- CNN

M Fadhilur Rahman^{1*}, Bambang²

¹Program Studi Teknologi Informasi, Universitas Nurul Jadid, Probolinggo

*Corresponding Author

E-mail: fadilurrahman88@gmail.com*, bambangb19@gmail.com

Abstrak

Sampah menjadi permasalahan yang tidak kunjung selesai pada kehidupan manusia. Banyak permasalahan yang disebabkan oleh sampah sebenarnya berasal dari sikap manusia yang tidak peduli terhadap lingkungan. Beberapa solusi sudah diajukan untuk menyelesaikan dan menghindari adanya masalah dari sampah tersebut, salah satunya membuat deteksi sampah yang dapat diaplikasikan secara langsung ke dalam device tertentu. Penelitian ini bertujuan untuk menerapkan sebuah metode deteksi objek berupa Faster R-CNN untuk melakukan deteksi dan klasifikasi dengan kecepatan yang memungkinkan komputer untuk mendeteksi objek sampah secara langsung melalui video real-time. Hasil uji coba pada penelitian ini menunjukkan metode yang digunakan dapat melakukan deteksi objek sampah pada 100 gambar dengan nilai akurasi sebesar 74%, dan untuk melakukan deteksi pada video real-time dengan kecepatan frame video pada kisaran 1 frame per second (fps).

Kata kunci: *Real-time, Deteksi, Faster R-CNN*

Abstract

Garbage is a never-ending problem in human life. Many of the problems caused by waste actually stem from human ignorance of the environment. Several solutions have been proposed to solve and avoid problems from the waste, one of which is making waste detection that can be applied directly to certain devices. This study aims to apply an object detection method in the form of Faster R-CNN to detect and classify at a speed that allows computers to detect trash objects directly through real-time video. The test results in this study indicate the method used can detect trash objects in 100 images with an accuracy value of 74%, and to detect real-time video with video frame rates in the range of 1 frame per second (fps).

Keywords: *Real-time, Detection, Faster R-CNN*

1. PENDAHULUAN

Sampah adalah segala sesuatu yang sudah digunakan dan tidak lagi dibutuhkan oleh manusia. Apabila mengacu pada pasal 1 UU No. 18 Tahun 2008 tentang Pengelolaan Sampah, disebutkan pengertian sampah yaitu sisa kegiatan sehari-hari manusia dan /atau proses alam yang berbentuk padat. Sampah adalah suatu benda yang sudah tidak digunakan lagi oleh manusia sehingga dibuang[1]. Dari beberapa pengertian tersebut dapat disimpulkan bahwa sampah adalah sesuatu yang sudah digunakan dan tidak lagi dibutuhkan sehingga dibuang dan ditinggalkan. Penelitian terkait sampah dan penanggulangannya juga telah banyak diteliti baik dari penggunaan teknologi IoT maupun pembuatan platform untuk menanggulangi sampah yang penuh dan harus dibuang serta dapat dikontrol dan dimonitoring oleh pemiliknya seperti yang dilakukan Rizqi dkk [2]

hingga dengan adanya teknologi artificial intelligent, penelitian deteksi sampah dilakukan yang kedepannya bisa digunakan untuk pemilihan pengkategorian sampah.

Membuang sampah tidak pada tempatnya dapat menyebabkan banyak masalah, selain membuat lingkungan kotor sampah yang menyangkut di jalur aliran air dapat menyebabkan masalah yang lebih besar lagi, contohnya banjir dan kerusakan lingkungan lainnya. Selain menyebabkan musibah banjir, sampah organik dan non-organik yang tidak dipisah bisa menjadi sumber penyakit karena sampah organik tidak terurai dengan baik apabila menyatu dengan sampah non-organik sehingga menjadi sarang bakteri dan kuman yang menjadi akar penyebab penyakit. Beberapa sampah non-organik seperti plastic tidak dapat teruraikan dengan sendirinya selama 20 sampai 100 tahun[3].

Persoalan di atas dapat terjadi pada seluruh negara yang ada di dunia, khususnya sering terjadi pada negara berkembang. Beberapa solusi terus diajukan untuk menemukan cara menyelesaikan problematika yang disebabkan oleh sampah, mulai dari dibuatnya tempat sampah pintar sampai robot pendeteksi sampah. Deteksi sampah menjadi persoalan tersendiri mengingat bentuk sampah yang tidak tetap.

Deteksi objek merupakan suatu proses dalam computer vision yang menggunakan komputer untuk melakukan deteksi pada suatu objek dalam gambar yang dijadikan input dan menandainya pada proses sehingga dapat dilihat pada output. Beberapa metode diajukan untuk melakukan proses deteksi, salah satunya yaitu Faster R-CNN. Faster R-CNN adalah suatu metode algoritma yang dikembangkan berdasarkan metode pendahulunya yaitu Fast R-CNN dengan menggunakan Convolutional Neural Network (CNN) sebagai tulang punggung dari metode tersebut[4]. Faster R-CNN diajukan oleh Shaoqing Ren untuk membuat suatu algoritma yang dapat mendeteksi objek dan melakukan klasifikasi dengan cepat. Faster R-CNN dapat melakukan deteksi pada sebuah video dengan kecepatan 5fps[5].

Tujuan dari penelitian ini adalah untuk melakukan deteksi objek sampah menggunakan metode Faster R-CNN dengan harapan dapat melakukan deteksi secara cepat pada video real-time.

Penelitian tentang deteksi juga sudah dilakukan oleh beberapa penelitian sebelumnya. Irfan Salimi dkk. melakukan penelitian terhadap deteksi objek sampah menggunakan metode Viola & Jones atau biasa disebut Haar-Cascade [6]. Selain melakukan deteksi penelitian tersebut juga membuat suatu teknik untuk melakukan klasifikasi terhadap sampah organik dan sampah non-organik menggunakan Support Vector Machine (SVM). Dari kedua metode tersebut sama-sama tidak menggunakan metode neural network untuk melakukan deteksi maupun klasifikasi terhadap objek sampah sehingga diperlukan proses pelatihan yang dilakukan secara manual.

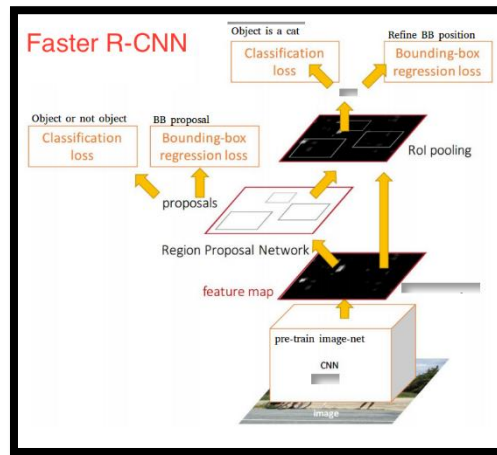
Sebuah penelitian dilakukan oleh Huaizu Jiang dkk. untuk melakukan deteksi objek wajah pada suatu citra menggunakan metode Faster R-CNN guna mendeteksi objek yang memiliki ukuran berbeda pada suatu citra [7]. Dalam penelitian tersebut objek wajah dideteksi dengan baik dengan menampilkan Bounding box pada objek wajah dengan ukuran yang serupa namun dalam skala yang berbeda-beda.

2. METODOLOGI

2.1. Faster R-CNN

Dalam penelitian ini menggunakan metode Faster R-CNN yang memakai CNN sebagai backbone dari metode yang diajukan. Faster R-CNN menjadi penerus dari metode sebelumnya yakni R-CNN dan Fast R-CNN yang dapat melakukan deteksi objek dengan tambahan Selective Search (SS) sebagai bentuk input yang akan diproses dalam CNN, Selective Search dapat

menentukan Regions of Interest (RoI). Dalam penggunaan Selective Search ditemukan bottleneck yang dapat membebankan proses komputasi sehingga memakan waktu yang lebih banyak untuk melakukan deteksi sehingga pada Faster R-CNN algoritma Selective Search diganti menggunakan Regional Proposal Network (RPN). Adapun arsitektur dari Faster R-CNN dapat dilihat pada Gambar 1.



Gambar 1. Arsitektur Faster R-CNN

2.2. Regional Proposal Network (RPN)

Regional Proposal Network merupakan metode baru yang diajukan sebagai pengganti Selective Search pada Fast R-CNN. Faster R-CNN menggunakan gambar seutuhnya sebagai input dan diproses menggunakan CNN dan menghasilkan feature map yang akan dikirim ke Regional Proposal Network untuk menentukan Region of Interest dari gambar. Kelebihan dari RPN yang lain yaitu dapat melakukan deteksi objek secara fleksibel dengan perbedaan ukuran menggunakan skala dan rasio gambar sebesar 3x3. RPN memiliki 2 loss function berupa object loss untuk menentukan apakah objek yang terdeteksi merupakan objek atau non objek (L_{cls}), dan bounding-box regression loss (L_{reg}) untuk menghitung kesalahan yang dapat terjadi dari bonding-box yang ditentukan. Adapun loss function dari RPN dapat dilihat sebagai berikut:

$$L(\{p_i\}, \{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*). \quad (1)$$

Digunakan predicted probability (p_i) dan 4 ribu coordinate (t_i) untuk setiap i-anchor dalam mini batch. Di sisi yang lain, p_i^* dan t_i^* merupakan ground truth dari label dan bounding-box. Kedua persamaan dilakukan normalisasi menggunakan N_{cls} , N_{reg} dan nilai lamda sebesar 10.

2.3. Convolutional Neural Network (CNN)

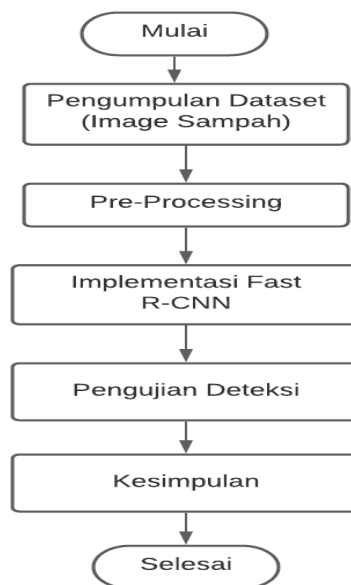
CNN menjadi backbone atau tulang belakang dari metode Faster R-CNN, fungsi dari CNN adalah untuk membuat feature map dari input gambar sehingga gambar lebih dapat dikenali oleh komputer. Feature map tersebut nantinya akan diproses dalam RPN untuk mengenali objek dan memberikan nilai bounding-box yang selanjutnya digunakan untuk melakukan klasifikasi pada objek dalam bounding-box. Banyak metode CNN yang bisa diaplikasikan ke dalam Faster R-CNN, dalam jurnal Shaoqing Ren [5] digunakan CNN arsitektur ZFNet dan VGG16 untuk menghasilkan feature map dari input gambar. Dalam penelitian ini digunakan arsitektur CNN Inception v2 dengan harapan dapat meringankan proses komputasi yang terlalu berat[8]. Inception v2 merupakan kelanjutan dari arsitektur inception dengan memperluas proses komputasi[9]. Teori Inception terbukti dapat mengurangi proses komputasi apabila dibandingkan dengan arsitektur lain yang memperdalam proses konvolusi.

2.4. Video Real-Time

Video adalah teknologi sinyal elektronik berupa gambar bergerak. Pengaplikasian teknologi video pada umumnya terdapat pada televisi, tetapi dapat juga digunakan dalam aplikasi teknik, saintifik produksi dan keamanan [10]. *Realtime* atau internet *broadcasting live* atau yang lebih dikenal *livecasting* adalah mekanisme pengontrolan, perekaman data, pemrosesan yang sangat cepat sehingga output yang dihasilkan dapat diterima dalam waktu yang relatif sama. *Realtime* menyiarkan suatu file media saat kejadian tengah berlangsung. Stasiun televisi di Indonesia jarang melakukan *livecasting* kecuali untuk satu-dua acara tertentu saja yang sifatnya insidental.

2.5. Metode

Metodologi yang digunakan dalam penelitian ini meliputi pengumpulan data image sampah, pre-processing, implementasi penggunaan Fast R-CNN, pengujian deteksi, kesimpulan dari percobaan pengujian deteksi yang telah dilakukan. Berikut merupakan metodologi penelitian dapat dilihat pada Gambar 2.



Gambar 2. Metodologi Penelitian

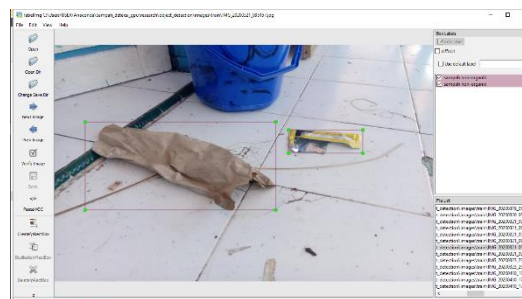
3. HASIL DAN PEMBAHASAN

Dalam bagian ini terdapat uji coba dari deteksi sampah menggunakan metode Faster R-CNN serta pembahasan terhadap hasil dari uji coba yang dilakukan. Adapun seluruh proses pelatihan dan uji coba menggunakan satu device PC dengan spesifikasi berupa processor Intel Core i5 gen. 7, RAM 8GB dan GPU NVIDIA GeForce 930MX dengan 2GB GDDR3 V-RAM.

3.1 Training model

Untuk mengurangi tingkat error dari deteksi, peneliti mengupulkan dataset secara manual menggunakan objek sampah yang ada di sekitar lokasi penelitian sehingga gambar sampah yang ingin diklasifikasikan lebih terlihat nyata dan sesuai untuk diaplikasikan pada deteksi real-time.

Setelah pengumpulan gambar untuk dataset, diperlukan membuat annotations dari gambar-gambar yang sudah dikumpulkan. Annotations tersebut digunakan untuk menentukan ground-truth yang digunakan untuk menghitung loss regression pada penentuan lokasi bounding-box.



Gambar 3. Annotation menggunakan LabelImg.

Proses training dilakukan menggunakan dataset yang sudah diberi annotation menunjukkan penurunan nilai loss yang tinggi pada awalnya dan turun secara bertahap dari waktu ke waktu proses pelatihan berjalan.

```
Anaconda Prompt (Anaconda3) - python train.py --logtostderr --train_dir=training/ --pipeline_config_path=training/faster_rcnn_inception_v2_pets.co...  
ry trying to allocate 2.34GiB with freed_by.count=0. The caller indicates that this is not a failure, but may mean that  
there could be performance gains if more memory were available.  
2020-05-08 13:33:50.560259: W tensorflow/core/common_runtime/bfc_allocator.cc:237] Allocator (GPU_0_bfc) ran out of memo  
ry trying to allocate 3.50GiB with freed_by.count=0. The caller indicates that this is not a failure, but may mean that  
there could be performance gains if more memory were available.  
INFO:tensorflow:Recording summary at step 0.  
I0508 13:33:54.889194 9580 supervisor.py:1050] Recording summary at step 0.  
INFO:tensorflow:global step 1: loss = 1.6358 (22.214 sec/step)  
I0508 13:34:00.792168 12384 learning.py:507] global step 1: loss = 1.6358 (22.214 sec/step)  
INFO:tensorflow:global step 2: loss = 2.1431 (1.328 sec/step)  
I0508 13:34:02.354300 12384 learning.py:507] global step 2: loss = 2.1431 (1.328 sec/step)  
INFO:tensorflow:global step 3: loss = 1.9846 (1.312 sec/step)  
I0508 13:34:03.666492 12384 learning.py:507] global step 3: loss = 1.9846 (1.312 sec/step)  
INFO:tensorflow:global step 4: loss = 2.1641 (1.328 sec/step)  
I0508 13:34:04.994307 12384 learning.py:507] global step 4: loss = 2.1641 (1.328 sec/step)  
INFO:tensorflow:global step 5: loss = 1.3257 (1.312 sec/step)  
I0508 13:34:06.306499 12384 learning.py:507] global step 5: loss = 1.3257 (1.312 sec/step)  
INFO:tensorflow:global step 6: loss = 1.2244 (1.328 sec/step)  
I0508 13:34:07.634314 12384 learning.py:507] global step 6: loss = 1.2244 (1.328 sec/step)  
INFO:tensorflow:global step 7: loss = 2.3721 (1.297 sec/step)  
I0508 13:34:08.930885 12384 learning.py:507] global step 7: loss = 2.3721 (1.297 sec/step)  
INFO:tensorflow:global step 8: loss = 2.7407 (1.312 sec/step)  
I0508 13:34:10.243075 12384 learning.py:507] global step 8: loss = 2.7407 (1.312 sec/step)  
INFO:tensorflow:global step 9: loss = 2.0269 (1.328 sec/step)  
I0508 13:34:11.570889 12384 learning.py:507] global step 9: loss = 2.0269 (1.328 sec/step)  
INFO:tensorflow:global step 10: loss = 2.7903 (1.328 sec/step)  
I0508 13:34:12.893783 12384 learning.py:507] global step 10: loss = 2.7903 (1.328 sec/step)  
INFO:tensorflow:global step 11: loss = 1.7945 (1.328 sec/step)  
I0508 13:34:14.226519 12384 learning.py:507] global step 11: loss = 1.7945 (1.328 sec/step)
```

```

Anaconda Prompt [Anaconda3]: python train.py --logstddir --train_dir=training/ --pipeline_config_path=training/faster_rcnn_inception_v2_pets.co...
INFO:tensorflow:global step 12751: loss = 0.0537 (1.328 sec/step)
I0401 00:00:17.574921 7640 learning.py:507] global step 12751: loss = 0.0537 (1.328 sec/step)
INFO:tensorflow:global step 12752: loss = 0.2046 (1.312 sec/step)
I0401 00:00:18.902732 7640 learning.py:507] global step 12752: loss = 0.2046 (1.312 sec/step)
INFO:tensorflow:global step 12753: loss = 0.0188 (1.296 sec/step)
I0401 00:00:20.213856 7640 learning.py:507] global step 12753: loss = 0.0188 (1.296 sec/step)
INFO:tensorflow:global step 12754: loss = 0.0871 (1.328 sec/step)
I0401 00:00:21.541669 7640 learning.py:507] global step 12754: loss = 0.0871 (1.328 sec/step)
INFO:tensorflow:global step 12755: loss = 0.0649 (1.312 sec/step)
I0401 00:00:22.869488 7640 learning.py:507] global step 12755: loss = 0.0649 (1.312 sec/step)
INFO:tensorflow:global step 12756: loss = 0.1495 (1.312 sec/step)
I0401 00:00:24.181672 7640 learning.py:507] global step 12756: loss = 0.1495 (1.312 sec/step)
INFO:tensorflow:global step 12757: loss = 0.0539 (1.312 sec/step)
I0401 00:00:25.493863 7640 learning.py:507] global step 12757: loss = 0.0539 (1.312 sec/step)
INFO:tensorflow:global step 12758: loss = 0.2258 (1.328 sec/step)
I0401 00:00:26.821675 7640 learning.py:507] global step 12758: loss = 0.2258 (1.328 sec/step)
INFO:tensorflow:global step 12759: loss = 0.0762 (1.328 sec/step)
I0401 00:00:28.149488 7640 learning.py:507] global step 12759: loss = 0.0762 (1.328 sec/step)
INFO:tensorflow:global step 12760: loss = 0.0396 (1.328 sec/step)
I0401 00:00:29.477302 7640 learning.py:507] global step 12760: loss = 0.0396 (1.328 sec/step)
INFO:tensorflow:global step 12761: loss = 0.2628 (1.312 sec/step)
I0401 00:00:30.805113 7640 learning.py:507] global step 12761: loss = 0.2628 (1.312 sec/step)
INFO:tensorflow:global step 12762: loss = 0.0764 (1.297 sec/step)
I0401 00:00:32.131682 7640 learning.py:507] global step 12762: loss = 0.0764 (1.297 sec/step)
INFO:tensorflow:global step 12763: loss = 0.0659 (1.328 sec/step)
I0401 00:00:33.429495 7640 learning.py:507] global step 12763: loss = 0.0659 (1.328 sec/step)
INFO:tensorflow:global step 12764: loss = 0.0473 (1.297 sec/step)
I0401 00:00:34.726866 7640 learning.py:507] global step 12764: loss = 0.0473 (1.297 sec/step)
INFO:tensorflow:global step 12765: loss = 0.2485 (1.312 sec/step)
I0401 00:00:36.024237 7640 learning.py:507] global step 12765: loss = 0.2485 (1.312 sec/step)

```

Gambar 4. Pelatihan dan loss dari metode Faster R-CNN

Dari proses tersebut dapat digambarkan pada grafik berikut:



Gambar 5. Grafik loss value dari proses training.

3.2 Pengujian deteksi

Pengujian dilakukan untuk melihat tingkat akurasi dari metode dalam melakukan deteksi terhadap objek sampah. Digunakan 100 gambar untuk uji coba dengan hasil 74 yang berhasil dideteksi dan benar dalam klasifikasi. Sedangkan sisanya gagal dalam melakukan deteksi maupun salah dalam melakukan klasifikasi. Dari hasil tersebut dapat diambil nilai persamaan sebesar $\frac{74}{100} \times 100$ sehingga dapat diambil kesimpulan nilai akurasi sebesar 74%.



Gambar 6. Berhasil melakukan deteksi dan klasifikasi



Gambar 7. Gagal dalam melakukan deteksi dan klasifikasi

Selain uji coba menggunakan gambar, uji coba juga dilakukan menggunakan video. Pengujian terhadap video dilakukan untuk mengetahui seberapa cepat dan memungkingkannya untuk digunakan pada kehidupan sehari-hari. Video berupa video real-time sehingga video yang digunakan untuk dilakukan proses deteksi benar-benar dalam waktu yang bersamaan dengan proses deteksi. Uji coba dilakukan menggunakan video dengan 6 perbedaan resolusi, hal ini mengacu pada perbedaan frame per seconds (fps) yang dimiliki dari masing-masing resolusi sehingga dapat membuat perbedaan dari setiap proses deteksi. Adapun hasil uji coba ini berupa nilai fps sebesar 0.96 pada resolusi 1280x720 dan nilai terbesar sebesar 1.35 fps pada resolusi 176x144. Keseluruhan uji coba dapat dilihat pada tabel 1.

Tabel 1. Kecepatan deteksi

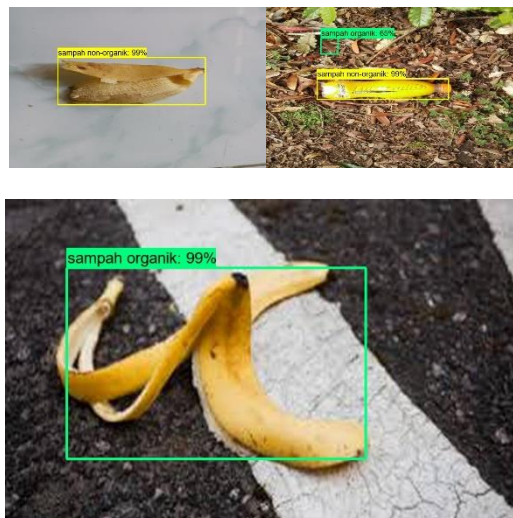
| No. | Resolusi | Rate |
|-----|----------|------|
|-----|----------|------|

| | | |
|---|----------|----------|
| 1 | 1280x720 | 0.96 fps |
| 2 | 720x480 | 1.08 fps |
| 3 | 640x480 | 1.14 fps |
| 4 | 480x320 | 1.24 fps |
| 5 | 320x240 | 1.24 fps |
| 6 | 176x144 | 1.35 fps |

Dari tabel 1 dapat diambil kesimpulan bahwa Faster R-CNN masih membutuhkan device dengan level yang lebih tinggi apabila dibandingkan dengan penelitian yang dilakukan author dari Faster R-CNN.

3.3 Analisis Model

Dari hasil uji coba didapat nilai akurasi sebesar 74%. 26% merupakan nilai dari keseluruhan uji coba yang gagal dalam melakukan deteksi terhadap objek sampah. Kegagalan tersebut dapat terjadi karena hasil feature map dari CNN yang dapat serupa dengan objek yang berbeda. Hal ini juga menunjukkan bahwa CNN tidak sesuai untuk melakukan klasifikasi terhadap objek yang memiliki bentuk tidak tetap dan pasti seperti objek sampah.



Gambar 8. Objek sampah yang salah dan benar diklasifikasi

4. KESIMPULAN

Pada bagian ini akan dijelaskan kesimpulan dari penelitian deteksi sampah menggunakan metode Faster R-CNN.

1. Tingkat akurasi dari metode Faster R-CNN dalam mendeteksi objek sampah sebesar 74%.

2. Digunakan video real-time dengan 6 resolusi berbeda dan didapatkan hasil deteksi pada video dengan kecepatan tertinggi sebesar 1.35 pada video dengan resolusi 176x144.
3. Walaupun dianggap sebagai metode yang memiliki kecepatan dalam pemrosesan citra, faktanya Faster R-CNN masih membutuhkan hardware dengan spesifikasi tinggi untuk diimplementasikan pada kehidupan sehari-hari.

5. DAFTAR PUSTAKA

- [1] M. Z. Elamin *et al.*, "Analysis of waste management in the Village of Disanah, District of Sreseh Sampang, Madura," *J. Kesehat. Lingkung.*, vol. 10, no. 4, pp. 368–375, 2018.
- [2] R. P. N. Budiarti, J. Maulana, and S. Sukaridhoto, "Aplikasi DIY Smart Trash Berbasis IoT Open Platform," *Appl. Technol. Comput. Sci. J.*, vol. 1, no. 2, pp. 93–104, 2018.
- [3] P. Purwaningrum, "Upaya mengurangi timbulan sampah plastik di lingkungan," *Indones. J. Urban Environ. Technol.*, vol. 8, no. 2, pp. 141–147, 2016.
- [4] R. Girshick, "Fast R-CNN. Proceedings of the IEEE International Conference on Computer Vision (ICCV)," 2015.
- [5] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2016.
- [6] I. Salimi, B. S. B. Dewantara, and I. K. Wibowo, "Visual-based trash detection and classification system for smart trash bin robot," in *2018 International Electronics Symposium on Knowledge Creation and Intelligent Computing (IES-KCIC)*, 2018, pp. 378–383.
- [7] H. Jiang and E. Learned-Miller, "Face detection with the faster R-CNN," in *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, 2017, pp. 650–657.
- [8] N. K. Manaswi, *Deep Learning with Applications Using Python*. 2018.
- [9] D. Alamsyah and M. Fachrurrozi, "Faster R-CNN with inception v2 for fingertip detection in homogenous background image," in *Journal of Physics: Conference Series*, 2019, vol. 1196, no. 1, p. 12017.
- [10] K. R. Anjani, "Pembuatan Video Klip Band Marsmellow dengan Teknik Penggabungan antara Stop Motion dan Live Shot Berjudul 'Jangan Bilang I Love You,'" STIKOM Surabaya, 2011.