

# Monitoring Pemakaian Masker Berbasis Video Menggunakan Metode *Faster R-CNN*

Muhammad Syahrul Romadhon dan Budi Setiyono

Departemen Matematika, Institut Teknologi Sepuluh Nopember (ITS)

*e-mail:* budi@matematika.its.ac.id.

**Abstrak**—*Coronavirus Disease 2019 (COVID-19)* menjadi masalah kesehatan utama yang menyebabkan penyakit pernapasan akut pada manusia. Tidak adanya kekebalan tubuh terhadap COVID-19 meningkatkan kerentanan terpapar virus, serta belum adanya *approved drug* mengakibatkan upaya dalam mengendalikan penularan wabah COVID-19 sangat bergantung pada intervensi *non-pharmaceutical* seperti upaya pencegahan secara perorangan, sebagai contoh penerapan protokol kesehatan. Pada 2020 lalu, WHO mengumumkan anjuran mengenai penerapan protokol kesehatan, salah satunya penggunaan masker guna mencegah penularan wabah COVID-19. Namun anjuran tersebut akan percuma, bila tidak ada kesadaran tiap individu masyarakat untuk memakai masker sesuai dengan prosedur yang ada. Dari permasalahan tersebut, dibutuhkan *monitoring* untuk menjamin pemakaian masker diterapkan sesuai dengan prosedur yang ada. *Monitoring* dapat dilakukan secara manual, tetapi membutuhkan biaya yang mahal dan sumber daya yang tidak sedikit. Oleh karena itu, dalam penelitian ini dirancang sistem *monitoring* otomatis dengan melakukan deteksi terhadap pengguna masker. Dalam penelitian ini, terdapat tiga jenis kelas (label) penggunaan masker, yaitu Pemakaian Masker Benar (*Mask*), Pemakaian Masker Salah (*ImproperlyMask*), dan Tidak Memakai Masker (*NoMask*). Proses pendeteksian penggunaan masker dalam penelitian ini dilakukan pada tahap *testing*, dengan langkah-langkah yaitu *input video*, *akuisisi video*, *pendefinisian ROI*, *deteksi objek dengan menggunakan metode Faster R-CNN*. Berdasarkan uji coba yang telah dilakukan pada penelitian ini, didapatkan nilai rata-rata presisi sebesar 98.21%, nilai rata-rata *recall* sebesar 97.09%, dan nilai rata-rata akurasi sebesar 95.36%.

**Kata Kunci**—Pemakaian Masker; *Faster R-CNN*; Deteksi Objek; *Monitoring*.

## I. PENDAHULUAN

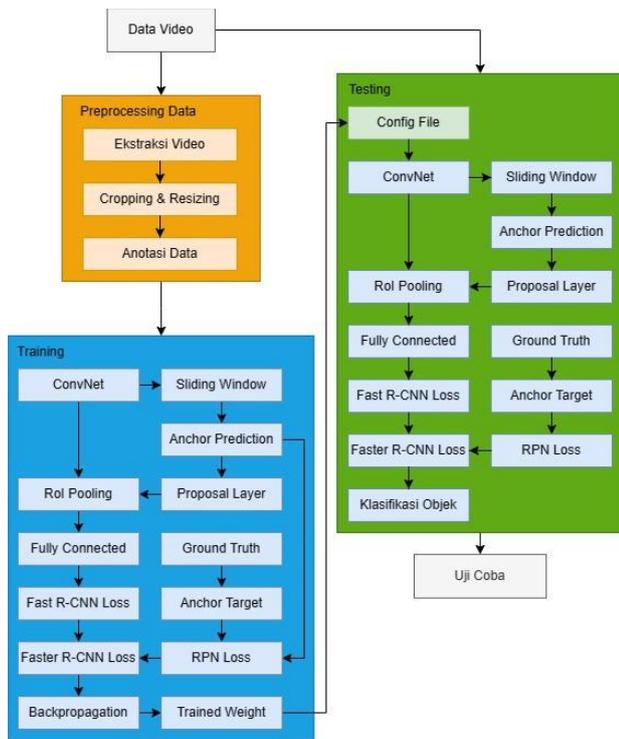
**C**ORONA Viruses Disease 2019 (COVID-19) menjadi masalah kesehatan utama yang menyebabkan penyakit pernapasan akut pada manusia. Virus ini ditemukan pada Desember 2019 di Wuhan, Provinsi Hubei. Berdasarkan Worldometer, pada tanggal 18 Januari 2022 dilaporkan bahwa pandemi COVID-19 telah menginfeksi 222 negara dengan banyaknya 56,862,824 kasus terkonfirmasi positif dan 5,563,834 kasus kematian [1]. Tidak adanya kekebalan tubuh terhadap COVID-19 meningkatkan kerentanan terpapar virus, serta belum adanya *approved drug* mengakibatkan upaya untuk mengendalikan penularan COVID-19 sangat bergantung pada intervensi *non-pharmaceutical* seperti tindakan pencegahan secara perorangan, sebagai contoh penerapan protokol kesehatan[2].

Di era pandemi, menerapkan protokol kesehatan merupakan hal yang wajib, khususnya dalam hal ini

penggunaan masker. Pada 5 Juni 2020, WHO mengumumkan anjuran mengenai penggunaan masker guna mencegah penularan wabah COVID-19. Namun anjuran tersebut akan percuma, bila tidak ada kesadaran tiap individu masyarakat untuk memakai masker sesuai dengan prosedur yang ada, khususnya yang menjadi sorotan dalam penelitian ini penggunaan masker tanpa menutupi bagian hidung dan mulut sepenuhnya, seperti memakai masker di bawah dagu ataupun di bawah hidung (*half-masking*) [3]. Menurut studi dari beberapa ahli kesehatan baru-baru ini, memakai masker di bawah dagu ataupun di bawah hidung (*half-masking*) bisa memberikan kesempatan virus *Corona* masuk ke dalam tubuh. Dokter bagian gawat darurat di sebuah rumah sakit, Fred Davis, juga menegaskan pentingnya memakai masker untuk menutupi hidung dan mulut. Menurutnya, hal ini bisa mengurangi risiko penyebaran maupun tertular virus COVID-19.

Untuk menangani hal tersebut, pemerintah perlu melakukan *monitoring* terhadap perilaku pemakaian masker masyarakat, dalam penelitian ini khususnya pejalan kaki. Upaya *monitoring* dapat dilakukan secara manual tetapi memerlukan biaya yang mahal dan sumber daya yang tidak sedikit. Oleh karena itu diperlukan suatu sistem yang dapat melakukan *monitoring* pemakaian masker secara otomatis. *Monitoring* dapat dilakukan apabila lokasi setiap objek yang diinginkan telah diketahui dan kriteria tiap objek dapat diketahui, seperti Pemakaian Benar, Pemakaian Salah, dan Tidak Memakai. Sehingga *monitoring* juga merupakan permasalahan pada deteksi objek dan klasifikasi objek berbasis data video. Oleh karena itu, diperlukan model yang sesuai untuk permasalahan *monitoring* berbasis data video tersebut.

Dalam beberapa tahun terakhir, *Deep Learning* telah menyita banyak perhatian, baik dalam penelitian ilmiah maupun penerapan praktis. *Deep learning* sendiri merupakan metode yang memiliki kemampuan dalam memodelkan data kompleks seperti citra dan suara, sehingga tepat apabila diterapkan untuk pengolahan data video. Metode *Convolutional Neural Network (CNN)* merupakan salah satu metode *deep learning* yang memiliki kemampuan untuk melakukan proses pembelajaran secara mandiri, khususnya dalam deteksi objek, ekstraksi objek dan klasifikasi pada input berupa citra. Seiring berjalannya waktu, metode CNN telah disempurnakan oleh para peneliti untuk memperoleh waktu komputasi yang semakin baik. Salah satunya adalah metode *Faster R-CNN*. *Faster R-CNN* merupakan sistem deteksi objek yang tersusun dari dua modul, yaitu RPN untuk menghasilkan region proposal dan *Fast R-CNN* untuk deteksi objek. *Fast R-CNN* dapat mendeteksi objek hingga mencapai tingkat hampir *real-time* dengan menggunakan jaringan *deep*



Gambar 1. Metode Tugas Akhir yang diusulkan untuk monitoring pemakaian masker menggunakan Faster R-CNN.

learning, ketika mengabaikan waktu dalam pencarian *region proposal* [4]. Padahal pada *Fast R-CNN* diperlukan input gambar yang memiliki *region proposal*, dalam *Fast R-CNN* terdapat *Selective Search*. *Selective Search* merupakan salah satu metode yang paling populer dalam menghasilkan *region proposal*. Tetapi *Selective Search* masih terbilang lambat, yaitu sekitar 2 detik per gambar. Sehingga diperkenalkan *Region Proposal Network* (RPN) yang berbagi konvolusi dengan jaringan deteksi objek yang canggih [5]. Dengan berbagi konvolusi, RPN mampu menghasilkan *region proposal* sekitar 10 ms per gambar. RPN yang ditambah dengan beberapa layer konvolusional, secara bersamaan dapat menghasilkan batas region dan skor objektivitas. Dengan demikian, RPN dapat ditraining secara khusus untuk menghasilkan region proposal. Untuk menyatukan *Fast R-CNN* dengan RPN, diperlukan skema pelatihan yang bergantian antara *fine-tune* untuk region proposal dan *fine-tune* untuk deteksi objek. Hasil ini menunjukkan bahwa metode *Faster R-CNN* memiliki waktu komputasi yang rendah dan juga merupakan cara yang efektif untuk meningkatkan akurasi dalam melakukan pendeteksian objek [4].

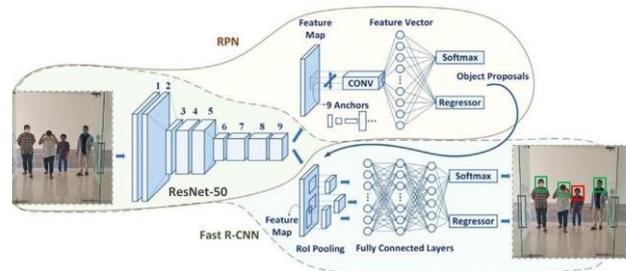
Berdasarkan kelebihan *Faster R-CNN* tersebut, dalam Studi penelitian ini akan dilakukan monitoring pemakaian masker, dengan kategori sebagai berikut *Mask*, *NoMask*, dan *ImproperlyMask* dengan menggunakan metode *Faster R-CNN* pada data video *pedestrian* (pejalan kaki). Dan kemudian, penyesuaian arsitektur dan pemilihan parameter pada metode *Faster R-CNN* akan dilakukan yang bertujuan untuk mendapatkan tingkat akurasi sebaik mungkin.

## II. PENELITIAN TERDAHULU

Yadav dan Binay dengan penelitian "*Comparative Study of Object Detection Algorithms*" [6], bertujuan menemukan kombinasi terbaik dari kecepatan dan akurasi sambil



Gambar 2. Tahap *Preprocessing*.



Gambar 3. Arsitektur *Faster R-CNN*

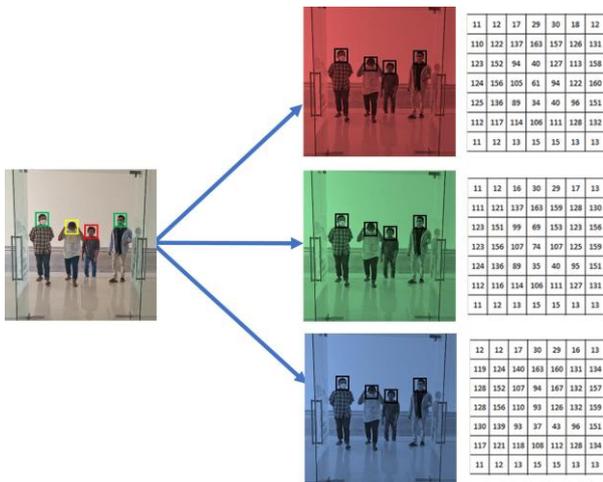
membandingkan berbagai algoritma deteksi objek yang menggunakan jaringan saraf konvolusional untuk melakukan deteksi objek.

Xiao dkk dengan penelitian "*Object Detection Based on Faster R-CNN Algorithm with Skip Pooling and Fusion of Contextual Information*" [7], mengulas tentang algoritma baru berdasarkan kerangka kerja *Faster R-CNN* yang dikombinasikan dengan *Faster R-CNN* dengan *skip pooling* dan *fusion*. Algoritma ini dapat meningkatkan kinerja pendeteksian dalam kondisi khusus berdasarkan *Faster R-CNN*.

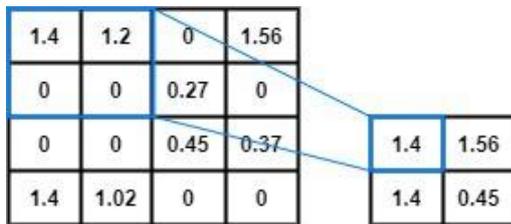
Halawa dkk dengan penelitian "*Face Recognition Using Faster R-CNN with Inception-V2 Architecture for CCTV Camera*" [8], dalam penelitian ini dilakukan pengenalan wajah yang digunakan untuk pengenalan pemilik kendaraan di tempat parkir yang telah dipasang CCTV. Metode *Faster R-CNN* digunakan untuk deteksi wajah dan juga untuk pengenalan wajah. Dalam penelitian ini juga dilakukan pengoptimalan *learning rate* dan *epoch* terbaik yang digunakan dalam model *Faster R-CNN* untuk meningkatkan akurasi dalam pengenalan wajah menggunakan CCTV.

Qin dan Li dengan penelitian "*Identifying Facemask-Wearing Condition Using Image Super-Resolution with Classification Network to Prevent COVID-19*" [9], dalam penelitian ini dilakukan pengembangan suatu metode identifikasi kondisi pemakaian masker wajah dengan menggabungkan jaringan super-resolusi dan klasifikasi (SRCNet), yang terdapat empat langkah utama di dalamnya: *Image Preprocessing*, deteksi wajah dan *cropping*, *image super-resolution*, dan identifikasi kondisi pemakaian masker wajah.

Lalu, Singh dkk dengan penelitian "*Face Mask Detection using YOLOv3 and Faster R-CNN Models: COVID-19 Environment*" [10]. Pada penelitian tersebut, Singh, dkk melatih dua model deteksi objek, yaitu YOLOv3 dan *Faster R-CNN*. *Dataset* yang digunakan terdiri dari gambar wajah



Gambar 4. Input Citra RGB.



Gambar 5. Proses Max Pooling.

Tabel 1. Tabel Data Testing

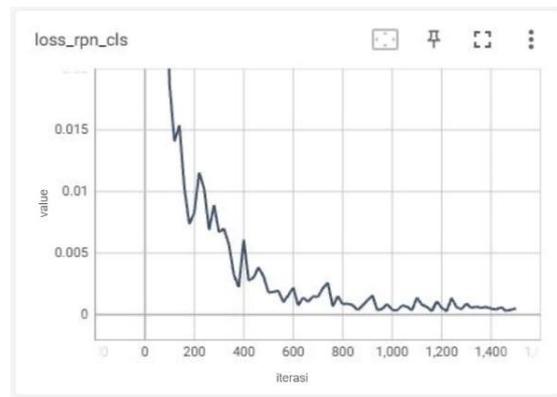
No.	Nama	Durasi	Jumlah Partisipan	Lokasi
1	t-01.mp4	0:01:08	10 orang	Gedung FSAD ITS
2	t-02.mp4	0:00:45	8 orang	T. 101B
3	t-03.mp4	0:00:23	5 orang	T. 101B

orang dengan dua kategori yaitu dengan dan tanpa masker. Uji coba ekstensif dilakukan, dan hasil yang didapat *Faster R-CNN* memiliki presisi yang lebih baik dibandingkan *YOLOv3*. Tetapi dari segi *frame rate*, *YOLOv3* jauh lebih unggul jika dibandingkan dengan *Faster R-CNN*.

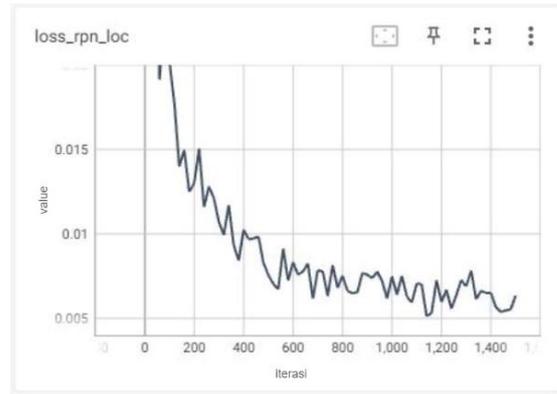
### III. METODE PENELITIAN

Penelitian ini bertujuan untuk mengetahui kinerja *Faster R-CNN* untuk mengklasifikasikan jenis pemakaian masker pejalan kaki di Institut Teknologi Sepuluh Nopember. Metode yang diusulkan pada Tugas Akhir ini memiliki empat tahapan utama yaitu tahap pengumpulan data, tahap *preprocessing data*, tahap implementasi metode *Faster R-CNN*, selanjutnya tahap uji coba dan evaluasi.

Tahap pengumpulan data bertujuan untuk mengetahui proses pengambilan data beserta isi *dataset* yang digunakan, dalam penelitian ini *dataset* yang dikumpulkan berupa data video. Tahap *preprocessing data* dibagi menjadi tiga proses, yaitu proses ekstraksi video, *cropping and resizing*, serta anotasi data. Anotasi data adalah tahap pemberian koordinat kotak pembatas atau *bounding box* serta label objek pada citra hasil ekstraksi video. Pembuatan data anotasi ini akan dipergunakan untuk melakukan proses *training*. Tahap implementasi *Faster R-CNN* bertujuan untuk mengklasifikasikan jenis pemakaian masker para pejalan kaki berdasarkan data video yang sudah dikumpulkan. Pada tahap



(a) Loss RPN cls



(b) Loss RPN loc

Gambar 6. Grafik Loss RPN

uji coba penelitian ini memiliki alur diantaranya yaitu *input* video, akuisisi video, pendefinisian ROI, dan proses deteksi menggunakan metode *Faster R-CNN*. Diagram alir Metode Penelitian dari langkah-langkah yang dikerjakan pada penelitian ini disajikan dalam Gambar 1.

#### A. Data Input

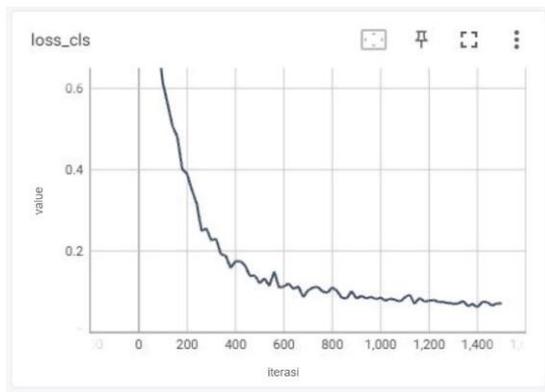
Data *input* sistem berupa video *offline* pejalan kaki yang diambil secara langsung dengan kamera seluler di beberapa lokasi. Data video digital digunakan sebagai *input* proses perancangan dan pengujian program. Pada data video akan dilakukan proses ekstraksi sehingga menghasilkan *frame-frame* citra. Kumpulan *frame* citra ini dijadikan data citra. Data citra yang berasal dari ekstraksi video akan dijadikan *input* dalam melakukan pelatihan program.

Data *input* sistem ini berupa video rekaman simulasi pejalan kaki yang diambil dengan menggunakan kamera seluler pada waktu siang hari dengan pencahayaan yang cukup. Pererekaman video yang dilakukan dengan menggunakan kamera seluler ini memiliki spesifikasi file berupa:

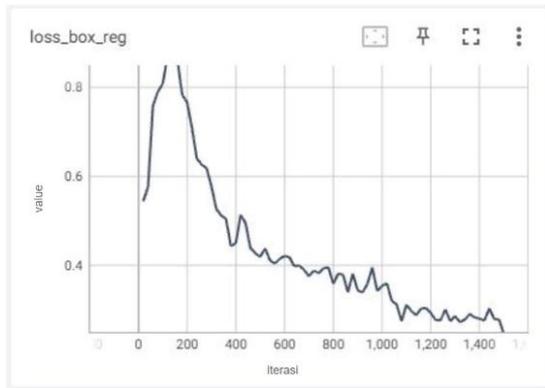
- 1) Format ekstensi rekaman berupa “.mp4”.
- 2) Rekaman video memiliki kecepatan 25 fps.

#### B. Data Preprocessing

Tahap awal sebelum membangun sistem *monitoring* adalah dengan mempersiapkan data berupa data video untuk proses pengujian program dan data citra untuk proses *training*. Data citra diperoleh dari tahap pemrosesan data video yang telah dilakukan yaitu berupa proses ekstraksi video, *cropping and resizing*, serta dilanjutkan dengan anotasi data citra. Tahap pemrosesan data dilakukan sebelum proses *training* program



(a) *Loss cls*



(b) *Loss box reg*

Gambar 7. Grafik Loss

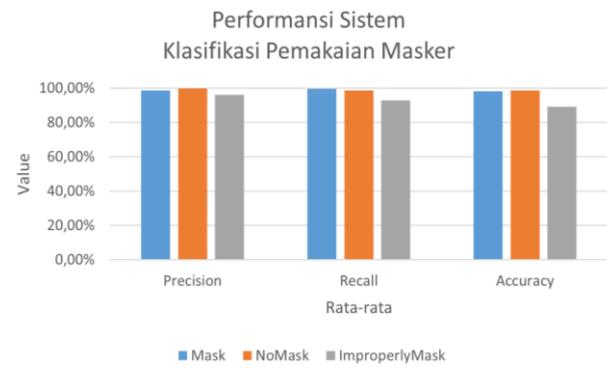
Tabel 4.2. Tabel Rata-rata Performansi Sistem

Jenis Klasifikasi	Precision	Recall	Accuracy
Mask	98.57%	99.54%	98.12%
NoMask	99.93%	98.78%	98.71%
ImproperlyMask	96.13%	92.96%	89.26%

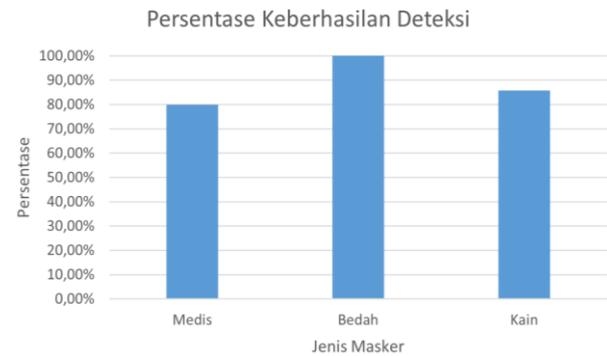
agar data *input* dapat digunakan secara optimal di tahap-tahap selanjutnya. Gambar 2 menunjukkan beberapa tahapan *preprocessing* yang dilakukan dalam penelitian ini.

C. Implementasi Metode Faster R-CNN

Desain arsitektur *Faster R-CNN* yang digunakan dalam penelitian ini ditunjukkan pada Gambar 3. Dalam desain arsitektur *Faster R-CNN* terdapat tahapan-tahapan utama diantaranya tahap ekstraksi fitur, *sliding window*, dan *RoI pooling layer*. Tahap ekstraksi fitur dilakukan dengan menggunakan metode ResNet-50 yang terdiri atas 50-layer dengan beberapa proses didalamnya. Saat proses ekstraksi fitur menggunakan ResNet-50 beberapa bagian didalamnya adalah *convolutional layer*, *max pooling layer*, ReLU, dan *batch normalization*. Dalam ResNet-50 terdapat *bottleneck block* yang berfungsi mengurangi terjadinya *overfitting* sehingga dapat diperoleh hasil yang maksimal. *Overfitting* terjadi saat model menghafal semua data *training* tanpa menganalisis *trend* dari data *training*. Tahap ekstraksi fitur menghasilkan *feature map* yang akan dijadikan *input* pada RPN. Tahap *sliding window* pada RPN digunakan untuk meningkatkan waktu komputasi metode *Faster R-CNN*. Seluruh *feature map* dilakukan pergeseran *sliding window* dengan *filter* 3 × 3 dan *stride* 2. Tiap lokasi *sliding window* dapat memprediksi *region proposal* pada masing-masing lokasi. Hasil dari tahap RPN berupa *feature vector* yang



Gambar 8. Grafik Performansi Sistem.



Gambar 9. Persentase Keberhasilan

disimpan sebagai *proposal layer*. Sedangkan pada tahap *Fast R-CNN* memiliki *input* berupa *feature map* dan *proposal layer* yang diperoleh dari tahap sebelumnya. Pada *Fast R-CNN* diterapkan *RoI pooling layer* yang berfungsi untuk mengubah ukuran *input* menjadi ukuran yang sama, yaitu diubah menjadi ukuran 14 × 14 × 1024. Setelah *input* memiliki ukuran yang sama, kemudian dilakukan operasi konvolusi pada blok konvolusi *conv5*. Operasi konvolusi ini berulang sebanyak tiga kali dan dilakukan proses *average pooling*. Output akhir proses ini memiliki ukuran 1 × 1 × 2048, kemudian dilanjutkan proses pada *fully connected layer* sehingga menghasilkan *output* berupa data klasifikasi pemakaian masker dan *box regresinya*.

1) Convolutional Layer

Proses konvolusi yang terjadi pada blok *conv1* memiliki *filter* atau *weight* yang berasal dari model *imagenet pre-trained* yang telah dilatih pada suatu *dataset*. Gambar 4 menunjukkan ilustrasi matriks *input* dan *filter* dalam blok *conv1* dengan *input* berupa citra berukuran 224 × 224 × 3, jumlah *filter* sebanyak 64 dengan ukuran 7 × 7 dan *stride* 2. Seluruh matriks pada tiap *channel* warna dengan matriks *filter* akan diterapkan operasi konvolusi. Dalam hal ini, matriks *filter* berisi nilai bobot *training* fitur yang telah diinisialisasi dari nilai *imagenet pre-trained*. Secara keseluruhan, apabila pada proses konvolusi input *convolutional layer* berupa gambar dengan ukuran  $W_1 \times H_1 \times D_1$  dengan  $W_1 = H_1$ , maka *output* dari layer tersebut adalah sebuah *image* baru dengan ukuran  $W_2 \times H_2 \times D_2$  dengan  $W_2 = H_2$ , di mana:

$$W_2 = \frac{W_1 - F + 2P}{S} + 1 = H_2 \tag{1}$$

$$P = \frac{F-1}{2} \tag{2}$$

$$D_2 = K \tag{3}$$

di mana,

$F$  = ukuran spasial *filter* (lebar/tinggi)

$K$  = banyaknya *filter* yang dipakai

$P$  = jumlah penambahan nol (*padding*)

$S$  = besar pergeseran *filter* dalam konvolusi (*stride*)

## 2) Rectified Linear Unit (ReLU)

Pada fungsi aktivasi ReLU semua nilai negatif akan dipetakan menjadi nol dengan fungsi  $f(x) = \max(0, x)$ .

$$f(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases} \quad (4)$$

## 3) Pooling Layer:

Matriks yang dihasilkan dari proses aktivasi dengan fungsi aktivasi ReLU akan dimasukkan ke dalam *pooling layer*. Tujuan dari *pooling layer* adalah untuk mengurangi ukuran piksel dari *feature map*, tahap ini biasa disebut dengan *down sampling*. Hal ini dapat mempercepat proses komputasi, karena selain piksel yang diproses sedikit parameter yang harus *update* juga lebih sedikit.

Metode *pooling* yang digunakan dalam *layer* ini adalah *max pooling* dengan pembagian *grid*  $2 \times 2$ , menggunakan *stride* sebesar 2, tanpa adanya *zero padding*. Gambar 5 menunjukkan ilustrasi proses dari *Max Pooling*.

## 4) RPN Loss

Nilai RPN *loss* terdiri dari nilai skor objektivitas dan *regressor box*. Dengan menggunakan rumus *loss function*, maka dapat dilakukan perbandingan antara hasil prediksi dengan data sesungguhnya. Selisih antara *anchor prediction* yang diperoleh dari proses RPN dengan *anchor target* dari *ground-truth*, merupakan nilai *error* yang disebut sebagai nilai RPN *loss*. Formula *loss function* adalah sebagai berikut.

$$L(\{p_i\}, \{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*) \quad (5)$$

di mana,

$$L_{cls}(p_i, p_i^*) = (-p_i^* \ln p_i - (1 - p_i^*) \ln(1 - p_i)) \quad (6)$$

$$L_{reg}(t_i, t_i^*) = R(t_i - t_i^*) \quad (7)$$

$$t_x = \frac{(x - x_a)}{w_a} \quad (8)$$

$$t_y = \frac{(y - y_a)}{h_a} \quad (9)$$

$$t_w = \log\left(\frac{w}{w_a}\right) \quad (10)$$

$$t_h = \log\left(\frac{h}{h_a}\right) \quad (11)$$

$$t_x^* = \frac{(x^* - x_a)}{w_a} \quad (12)$$

$$t_y^* = \frac{(y^* - y_a)}{h_a} \quad (13)$$

$$t_w^* = \log\left(\frac{w^*}{w_a}\right) \quad (14)$$

$$t_h^* = \log\left(\frac{h^*}{h_a}\right) \quad (15)$$

$$t_i = t_x + t_y + t_w + t_h \quad (16)$$

$$t_i^* = t_x^* + t_y^* + t_w^* + t_h^* \quad (17)$$

## 5) Fast R-CNN Loss

Selisih antara *anchor* prediksi dari proses *Fast R-CNN* dengan *anchor target* dari *ground-truth* merupakan nilai *error* yang disebut sebagai *Fast R-CNN loss*. Sama halnya seperti RPN *loss*, nilai dari *Fast R-CNN loss* terdiri dari dua nilai yaitu nilai *regressor box* dan skor objektivitas. Sedangkan untuk perhitungan *Fast R-CNN loss* menggunakan *loss function* yang sama seperti pada RPN *loss*.

## 6) Fully-Connected Layer

Dalam *fully-connected*, neuron-neuron aktivasi pada lapisan sebelumnya dihubungkan dengan neuron di lapisan selanjutnya, artinya semua neuron yang ada secara keseluruhan saling terhubung satu sama lain. *Fully-connected layer* dalam sistem ini terdiri dari satu *hidden layer*. Dalam *hidden layer* terdapat proses perkalian fitur-fitur hasil pembelajaran sebelumnya dengan bobot jaringan dan dijumlahkan dengan bias. Bobot jaringan  $\omega_{i,j}$ , memiliki ukuran  $2048 \times 3$  karena jumlah *input* pada *layer* ini sebanyak 2048 dan untuk diklasifikasikan ke dalam 3 jenis kelas data. Nilai bobot dan bias pada *training* awal diinisialisasi dari model *imagenet pre-trained* yang nantinya akan diperbarui hingga diperoleh model yang optimal.

## 7) Softmax Layer

Nilai output dari *fully-connected* diaktivasi dengan menggunakan fungsi aktivasi *softmax*. Dengan nilai maksimum yang didapat dari nilai *softmax* ini akan menjadi dasar hasil akhir pendefinisian jenis pemakaian masker.

Nilai *loss function* digunakan untuk menyatakan seberapa baik model dari jaringan yang dilatih. Jaringan dapat mengevaluasi kinerja yang dilakukan, apabila nilai *error* yang dihasilkan tinggi maka jaringan akan melakukan pembelajaran kembali hingga memperoleh nilai *error* yang rendah. Nilai *loss function* dihitung dengan menggunakan *cross entropy*

$$L_{CE}(s, t) = - \sum_i t_i \log(s_i) \quad (18)$$

di mana,

$t_i$  = label kepercayaan (*truth label*)

$s_i$  = nilai probabilitas *softmax* untuk  $i^{th}$  class

$i$  = iterasi class

Dari perhitungan *cross entropy* nantinya akan dilakukan proses *backpropagation* pada nilai *Faster R-CNN loss* untuk memperbaiki parameter-parameter yang digunakan dalam sistem ini.

## IV. HASIL DAN PEMBAHASAN

Pada bagian ini dijelaskan mengenai hasil uji coba dan pembahasan dari program yang telah dibuat. Pengujian program dilakukan dengan menggunakan *input* video rekaman pejalan kaki di berbagai tempat pengambilan video.

### A. Data Uji Coba

Data video yang digunakan untuk uji coba berekstensi “.mp4”, yang memiliki resolusi  $1920 \times 1080$  dengan kecepatan 30 fps (*frame per second*). Uji coba pada penelitian ini dilakukan pada data *testing* berupa data video yang dibedakan menjadi tiga macam jenis data. Pada data video “t-01.mp4” jumlah partisipan sebanyak 10 orang, dalam video tersebut untuk setiap orangnya berjalan secara bergiliran, dan juga bersamaan. Pada video “t-02.mp4” jumlah partisipan

sebanyak 8 orang, dalam video tersebut pejalan kaki berjalan secara acak (tak beraturan). Pada video "t-03.mp4" jumlah partisipan sebanyak 5 orang, di dalam video tersebut terdapat 2-3 orang berjalan secara bersamaan.

Data video diambil di dua lokasi yang sesuai dengan Tabel 1. Dari data video yang dipilih terdapat dua lokasi yang berbeda dengan kondisi jenis masker dan jumlah partisipan yang berbeda. Hal tersebut bertujuan untuk melihat pengaruh jenis masker yang digunakan dan jumlah partisipan yang berbeda terhadap performansi dari sistem. Selain itu, untuk mengetahui performansi dari sistem apabila terdapat perubahan jumlah partisipan pada suatu data video (partisipan masuk dan/atau keluar *frame*).

### B. Hasil Tahap Training

Pada tahap *training* diperoleh *loss value* dari model RPN berupa nilai *loss RPN cls* dan *loss RPN loc*. Grafik *Loss RPN* dapat dilihat pada Gambar 6. Dari grafik tersebut terlihat bahwa *loss value* RPN untuk klasifikasi menurun sangat signifikan pada 400 iterasi pertama, kemudian perlahan *loss value* mulai stabil dengan mendekati nilai 0 pada iterasi-iterasi berikutnya. Begitu juga dengan *loss value* RPN regresi untuk 600 iterasi pertama, terlihat bahwa *loss value* menurun sangat signifikan, kemudian pada iterasi berikutnya *loss value* perlahan mulai stabil dengan mendekati nilai 0. Artinya, dapat dikatakan bahwa *loss value* dari objektivitas RPN dan koordinat kotak pembatas RPN perlahan akan menurun sesuai dengan banyaknya iterasi yang digunakan.

Proses pembelajaran serupa pada model *classification* memiliki *loss value* berupa *loss cls* dan *loss reg* yang ditunjukkan oleh grafik Gambar 7. Dari grafik tersebut terlihat bahwa *loss* untuk klasifikasi atau *loss cls* menurun sangat signifikan pada 400 iterasi pertama, kemudian perlahan *loss value* mulai stabil dengan mendekati nilai 0 pada iterasi-iterasi berikutnya. Begitu juga dengan *loss* untuk regresi atau *loss reg*. Pada beberapa tahap awal iterasi grafik memperlihatkan bahwa model memerlukan penyesuaian saat *training*. Sehingga pada beberapa tahap awal iterasi terlihat bahwa *loss reg* mengalami kenaikan, setelah itu pada iterasi 200 – 400 *loss value* regresi menurun sangat signifikan, kemudian pada iterasi berikutnya *loss value* perlahan mulai stabil dengan mendekati nilai 0. Artinya, dapat dikatakan bahwa *loss value* dari objektivitas dan koordinat kotak pembatas perlahan akan menurun sesuai dengan banyaknya iterasi yang digunakan.

### C. Pembahasan Uji Coba Tahap Testing

Pada pembahasan uji coba, performansi sistem dihitung menggunakan presisi, *recall*, dan akurasi sebagai berikut.

$$Precision = \frac{TP}{TP+FP} \times 100\% \quad (19)$$

$$Recall = \frac{TP}{TP+FN} \times 100\% \quad (20)$$

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \quad (21)$$

Presisi atau *precision* digunakan untuk mengetahui tingkat ketepatan antara anotasi yang dilakukan dengan hasil yang diberikan oleh sistem. *Recall* digunakan untuk mengetahui tingkat keberhasilan sistem dalam menemukan kembali suatu informasi. Sedangkan akurasi digunakan untuk mengetahui tingkat kedekatan antara hasil prediksi sistem dengan nilai

aktual atau nilai sebenarnya. Rata-rata performansi di dapatkan sebagai berikut.

Grafik dari rata-rata performansi sistem ditunjukkan pada gambar di bawah ini, di mana dalam grafik Gambar 8 di bawah ini terdapat nilai *Precision*, *Recall*, dan *Accuracy*.

Dalam penelitian ini pemakaian masker dikatakan berhasil dideteksi ketika nilai dari Terdeteksi Benar (TB) lebih besar dibandingkan jumlah dari Terdeteksi Salah (TS) dan Tidak Terdeteksi (TT). Dari seluruh data uji coba akan dihitung berapa kali pemakaian masker terlihat, kemudian di lakukan analisa apakah jenis pemakaian masker yang terlihat tersebut tergolong berhasil dideteksi atau tidak. Berikut grafik dari Hasil Uji Coba Keakuratan Jenis Masker.

Dari grafik Gambar 9 tersebut, disimpulkan bahwa persentase keberhasilan deteksi jenis pemakaian masker berdasarkan jenis masker yang dilakukan sistem terlihat lebih baik pada video dengan jenis masker bedah (*surgical mask*) daripada dua jenis masker lainnya. Dengan perhitungan nilai persentase keberhasilan deteksi pada jenis masker bedah adalah 100%. Hal ini dapat disebabkan oleh beberapa faktor diantaranya karena distribusi masker bedah pada data *train* besar atau juga karena bentuk dari masker bedah yang mudah untuk dideteksi.

## V. KESIMPULAN DAN SARAN

### A. Kesimpulan

- 1) Telah berhasil dirancang program yang mampu mendeteksi jenis pemakaian masker pejalan kaki melalui video dengan langkah-langkah yaitu *input* video, akuisisi video, pendefinisian ROI, deteksi objek menggunakan metode *Faster R-CNN*.
- 2) Berdasarkan uji coba menunjukkan sistem dapat melakukan pendeteksian jenis pemakaian masker dengan nilai rata-rata presisi sebesar 98.21%, nilai rata-rata *recall* sebesar 97.09%, nilai rata-rata akurasi sebesar 95.36% serta rata-rata keberhasilan deteksi sebesar 84.61%.
- 3) Uji Coba Berdasarkan data video didapatkan hasil terbaik pada data video t-03 dengan rata-rata *precision* sebesar 99.86%, *recall* sebesar 99.78%, dan *accuracy* 99.64%.
- 4) Saat dilakukan uji coba perhitungan persentase keberhasilan deteksi jenis klasifikasi pemakaian masker berdasarkan jenis masker, hasil terbaik didapat pada jenis masker bedah (*surgical mask*) dengan persentase sebesar 100%.

### B. Saran

- 1) Untuk penelitian selanjutnya dapat mengembangkan sistem notifikasi suara berdasarkan pemakaian masker salah dan tidak memakai masker diberbagai kondisi, agar nantinya dapat menghimbau secara otomatis pejalan kaki yang tidak mengenakan masker sesuai dengan prosedur yang ada.
- 2) Perlu dikembangkan suatu sistem untuk dapat memproses video yang terhubung secara langsung dengan kamera CCTV untuk deteksi dan klasifikasi pemakaian masker.
- 3) Untuk penelitian selanjutnya yang berkaitan dengan *deep learning*, disarankan agar memperbanyak data *training* dalam berbagai kondisi, dan menambah jumlah partisipan.

- 4) Disarankan menggunakan perangkat keras yang mumpuni, seperti CPU, RAM, dan GPU untuk menjalankan sistem berbasis *deep learning*, agar dapat meningkatkan performansi dan kecepatan komputasi sistem yang dibuat.

#### DAFTAR PUSTAKA

- [1] "COVID Live - Coronavirus Statistics," *Worldometer*. <https://www.worldometers.info/coronavirus/> (accessed Aug. 15, 2022).
- [2] Satuan Tugas Penanganan COVID-19, *Pengendalian COVID-19 Dengan 3M, 3T, Vaksinasi, Disiplin, Kompak, dan Konsisten*. Jakarta, Indonesia: Satuan Tugas Penanganan COVID-19, 2021.
- [3] World Health Organization, "Anjuran Mengenai Penggunaan Masker dalam Konteks COVID-19." 2020.
- [4] S. Ren, K. He, R. G. Jian, 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, 2017, doi: 10.2307/j.ctt1d98bxx.10.
- [5] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition BT - Computer Vision – ECCV 2014," in *Computer Vision – ECCV 2014. ECCV 2014. Lecture Notes in Computer Science*, 2014, pp. 346–361. doi: 10.1007/978-3-319-10578-9\_23.
- [6] N. Yadav and U. Binay, "Comparative study of object detection algorithms," *International Research Journal of Engineering and Technology (IRJET)*, vol. 4, no. 11, pp. 586–591, 2017.
- [7] Y. Xiao, X. Wang, P. Zhang, F. Meng, and F. Shao, "Object detection based on faster R-CNN algorithm with skip pooling and fusion of contextual information," *Sensors* 2020, vol. 20, no. 19, p. 5490, 2020, doi: 10.3390/S20195490.
- [8] L. J. Halawa, A. Wibowo, and F. Ernawan, "Face recognition using faster R-CNN with Inception-V2 architecture for CCTV camera," in *ICICOS 2019 - 3rd International Conference on Informatics and Computational Sciences: Accelerating Informatics and Computational Research for Smarter Society in The Era of Industry 4.0, Proceedings*, 2019. doi: 10.1109/ICICOS48119.2019.8982383.
- [9] B. Qin and D. Li, "Identifying facemask-wearing condition using image super-resolution with classification network to prevent COVID-19," *Sensors*, vol. 20, no. 18, p. 5236, 2020, doi: 10.3390/S20185236.
- [10] S. Singh, U. Ahuja, M. Kumar, K. Kumar, and M. Sachdeva, "Face mask detection using YOLOv3 and faster R-CNN models: COVID-19 environment," *Multimed Tools Appl*, vol. 80, no. 13, pp. 19753–19768, 2021, doi: 10.1007/S11042-021-10711-8/FIGURES/8.