Pendeteksi Masker pada Gambar Menggunakan Model Deep Learning Yolo-v2 dengan ResNet-50

1st Muhammad Rizki Atria Salim
Fakultas Informatika
Universitas Telkom
Bandung, Indonesia
rizkiatriasalim@student.telkomuniversity.ac.id

2nd Febryanti Sthevanie
Fakultas Informatika
Universitas Telkom
Bandung, Indonesia
sthevanie@telkomuniversity.ac.id

3rd Kurniawan Nur Ramadhani
Fakultas Informatika
Universitas Telkom
Bandung, Indonesia
kurniawannr@telkomuniversity.ac.id

***Abstrak*— Sistem deteksi masker merupakan suatu upaya untuk mencegah penyebaran COVID-19. Pada penelitian ini sistem deteksi masker dikembangkan menggunakan model *deep learning* Yolo-v2 dengan bantuan ResNet-50. ResNet-50 digunakan sebagai *backbone layer* pengganti Yolo-v2, sedangkan Yolo-v2 menjadi komponen utama pendeteksi *face mask*. Penelitian ini menggunakan *Face Mask Dataset* dan *Medical Mask Dataset* berupa citra gambar yang diambil dari kaggle. Pengujian parameter konfigurasi saat *training model* dilakukan dengan harapan dapat meningkatkan akurasi dan kinerja dari sistem deteksi masker. Sistem deteksi masker menggunakan metode ini mendapatkan hasil F1-*Score* sebesar 84%.**

***Kata Kunci* — deteksi masker, ResNet-50, YOLO-v2, COVID-19**

I. PENDAHULUAN

A. Latar Belakang

Pandemi COVID-19 masih menjadi masalah dunia pada saat ini. Menurut World Health Organization(WHO), penyebab COVID-19 menular antar manusia umumnya dikarenakan kontak dengan orang melalui mulut dan hidung. Lalu penularan virus melalui pasien COVID-19 yang berinteraksi melalui droplet seperti air liur, ingus dan air tetesan batuk yang menempel terhadap benda atau permukaan di sekitar, secara langsung atau tidak langsung apabila tersentuh oleh orang sehat dan menyentuh bagian mulut, hidung ataupun mata dapat menyebabkan penularan virus corona [1].

Untuk menghindari kontak dengan droplet, hal yang dapat dilakukan adalah social distancing dengan orang banyak, sering mencuci tangan, dan menutup mulut saat bersin atau batuk. Sering kali menjaga jarak fisik tidak memungkinkan pada suatu kondisi, pemakaian masker adalah langkah penting untuk mencegah penyebaran COVID-19. Penggunaan masker merupakan tindakan pencegahan terhadap penyebaran penyakit COVID-19 dan virus pernapasan lain. Tindakan ini ditujukan untuk perlindungan diri ketika melakukan kontak dengan orang terinfeksi atau untuk mencegah penularan dari individu terinfeksi [2]. Untuk berjuang melawan pandemi COVID-19, Saat ini pemerintah masih membutuhkan pengawasan terhadap masyarakat pada tempat-tempat umum yang sering menjadi tempat keramaian, seperti stasiun, rumah sakit, sekolah, pasar dan tempat lainnya. Dalam menerapkan aturan penggunaan masker di tempat umum, penggunaan model artificial intelligence menjadi salah satu jawaban.

Sebuah sistem deteksi objek yang dapat mendeteksi dan mencari masker medis pada gambar menjadi fokus utama penelitian ini. Semua metode deteksi objek diklasifikasikan menjadi sistem berbasis machine learning dan deep learning. Deteksi objek menggunakan machine learning, sebagai contohnya Scale-invariant Feature Transform (SIFT) [3] dan Histogram of Oriented Gradients (HOG) [4]. Deteksi objek menggunakan deep learning, sebagai contohnya Region Convolutional Neural Network (R-CNN) [5], Fast R-CNN [6], Faster R-CNN [7]. Selanjutnya deteksi objek secara realtime, sebagai contohnya, You Only Look Once v1 (YOLO v1) [8], Yolo- v2 [9], Yolo-v3 [10], dan Single Shot Multi-Box Detector (SSD) [11]. Dari semua model pendeteksi objek, YOLO memiliki kinerja yang lebih cepat, efektif dan performansi yang baik. Penelitian terkait penggunaan Yolo-v2 sebagai deteksi masker model deep learning dengan bantuan ResNet-50 sebagai feature extractor [12]. Penelitian terkait berhasil membangun model dengan average precision sebesar 81%. Model dalam penelitian tersebut dapat ditingkatkan dengan mengubah parameter konfigurasi saat training model, contohnya: nilai batch, subdivision dan learning rate. Dataset pada penelitian ini menggunakan penggabungan Medical Mask Dataset dan Face Mask Dataset. Penelitian ini mencakup pengubahan layer Yolo-v2 dengan menggunakan layer ResNet-50. Lalu model Yolo-v2 dibandingkan dengan Yolo-v2 backbone ResNet-50. Semua hal tersebut dilakukan dengan harapan dapat meningkatkan akurasi dan kinerja dari sistem deteksi masker medis.

B. Topik dan Batasannya

Topik pada penelitian ini adalah membangun sistem pendeteksi masker dengan metode Yolo-v2 sebagai pendeteksi masker dan ResNet-50 sebagai *backbone layer*. Penelitian ini hanya menggunakan label yang terdiri dari tiga kelas, yaitu *with mask*, *without mask*, *incorrectly using*

mask. Training model dilakukan pada *repository darknet* AlexeyB. Penelitian difokuskan pada model Yolo-v2 dengan *backbone* ResNet-50.

C. Tujuan

Tujuan dari penelitian ini, yaitu untuk membangun, mengimplementasi, dan mengukur performansi dari sistem deteksi masker medis menggunakan metode Yolo-v2 dengan bantuan ResNet-50 sebagai *backbone layer*.

D. Organisasi Tulisan

Pendahuluan menjelaskan terkait latar belakang, topik dan tujuan penelitian. Studi terkait yang menjelaskan dengan penilitian terkait yang pernah dilakukan sebelumnya. Sistem yang dibangun merupakan penjelasan rancangan sistem secara detail. Evaluasi yang merupakan hasil dari rancangan sistem yang telah dibangun. Kesimpulan yang berisi kesimpulan serta saran sebagai acuan penelitian berikutnya.

II. KAJIAN TEORI

A. Studi Terkait

1. Penelitian terkait

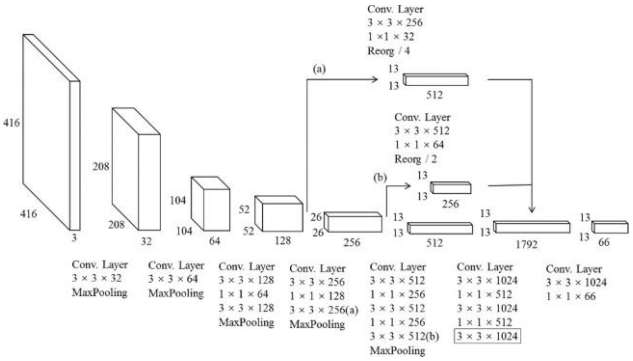
Mandal, dkk [13] menggunakan Simulated Masked Face Recognition Datasets (SMFRD) dan Real-world Masked Face Verification (RMFD) di mana total 5.000 mask face dan 90.000 unmasked faces. Melakukan penelitian menggunakan arsitektur ResNet-50. Peformance metrics yang didapatkan, pada unmasked data, hasil latihan mendapatkan precision bernilai 0,8933; recall bernilai 0,8970 dan F1-Score bernilai 0,897. Sedangkan Masked data model mendapatkan precision value bernilai 0,4613; lalu recall bernilai 0,4719 dan F1-Score bernilai 0,4473.

Vickya, dkk [14] Penelitian ini membandingkan beberapa model deep learning dalam fungsi mendeteksi masker wajah. Model yang dievaluasi beserta hasil akurasi yang diperoleh penelitian ini sebagai berikut: ResNet- 50 dengan akurasi 0,844; VGG11 dengan akurasi 0,843; InceptionV3 dengan akurasi 0,873; EfficientNetB4 dengan akurasi 0,957; Yolo-v2 sebesar 0,787; Yolo-v3 sebesar 0,863 dan Yolo-v4 sebesar 0,934. Dari semua hasil tersebut ResNet-50 merupakan arsitektur tercepat saat dilakukan training.

Loey, dkk [12] menggunakan arsitektur Yolo-v2 dan transfer learning ResNet-50 . Optimizer adam dan SGDM juga digunakan untuk menaikan proses deteksi objek. Proses pembuatan model dilakukan menggunakan Tensorflow dan MATLAB. Hasil tertinggi yang diperoleh dengan metode tersebut adalah 81% Average Precision.

2. Yolo-v2

You Only Look Once (YOLO) adalah *detector* yang dapat belajar untuk melakukan deteksi pada banyak objek secara bersamaan [8]. Algoritma ini dapat mendeteksi, mengenali dan mengklasifikasi banyak objek pada gambar dalam waktu yang *real-time*. Yolo-v2 memungkinkan prediksi beberapa *bounding box*. Selain itu, Yolo-v2 meningkatkan *average precision* dengan *batch normalization*. Arsitektur YOLO dapat divisualisasikan pada Gambar 1.



GAMBAR 1
Arsitektur YOLO

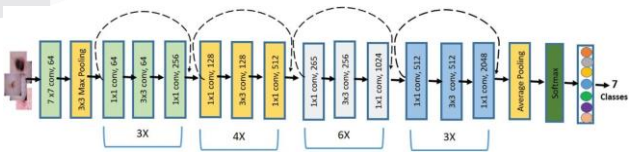
Arsitektur Yolo-v2 memiliki total 23 *layer*. Layer tesebut dibagi menjadi tiga yaitu 1 *input convolutional layer*, 20 *convolutional layer*, 5 *maxpool layer* dan 2 *fully connected layer*. Hasil prediksi pada gambar dengan model Yolo-v2 adalah *bounding box*, *confident score* dan *label class*. Sistem Yolo-v2 dalam pendeteksiannya digunakan sebagai *classifier* dan *localizer*. Yolo menetapkan sebuah citra gambar sebagai sebuah pendeteksian ketika terdapat daerah pada citra dengan nilai IoU *score* tertinggi. IoU pada deteksi objek adalah mengevaluasi tingkat jarak ataupun *overlap* antara *ground truth* dan *prediction bounding box*. Berikut adalah perhitungan IoU yang dijelaskan pada Rumus 1.

$$\text{IoU} = \frac{Area\ of\ Overlap}{Area\ of\ Union} \quad (1)$$

IoU ditentukan menggunakan IoU *Threshold*. *Threshold* merupakan suatu batasan nilai untuk mengukur tingkat kedekatan *ground truth bounding box* dan *prediction bounding box*. Untuk IoU *Threshold* = α, *True Positive (TP)* adalah deteksi yang IoU $\geq a$ *dan False Positive adalah* deteksi yang IoU $< a$. *False* Negatif adalah kebenaran dasar yang terlewatkan di mana IoU $< \alpha$.

3. Arsitektur Residual Network-50 (ResNet-50)

ResNet memiliki banyak varian yang sama dengan jumlah layer berbeda. ResNet-50 adalah salah satu varian ResNet yang dapat bekerja pada 50 lapisan *neural network*. Pada visi komputer, arsitektur ini dapat bekerja sebagai *image classification*, *object localization*, *object detection*. Berikut adalah penggambaran arsitektur ResNet pada Gambar 2.



GAMBAR 2
Arsitektur ResNet-50

Residual block terdiri dari dua atau tiga *sequential convolutional layers* yang dalam satu *residual block* memiliki *size* 1x1, 3x3 dan 1x1 dan *shortcut layer* atau *skip connection* yang menghubungkan input dari lapisan pertama dan outputnya yang melewati beberapa lapisan model [15]. *Residual block* pertama diulang hingga tiga kali dan secara berurutan *filters* adalah 64, 64 dan 256. *Residual block*

kedua diulang hingga empat kali dengan *filters* 128, 128 dan 512. *Residual block* ketiga diulang hingga enam kali dengan 256, 256 dan 1024. Residual block kedua diulang hingga empat kali dengan filters 128, 128 dan 512. Residual block keempat diulang hingga tiga kali dengan filters 512, 512 dan 2048.

4. Object Recognition

Object Recognition adalah kumpulan tugas dari visi komputer dalam citra digital. Tugas dari visi komputer ini dapat dibedakan menjadi tiga hal, yaitu Image classification, object localization, dan object detection. Image classification adalah algoritma yang menghasilkan daftar kategori objek yang ada di dalam gambar. Object Localization adalah algoritma yang menemukan keberadaan objek dalam sebuah gambar dan menunjukan posisinya dalam bentuk bounding box. Deteksi objek adalah algoritma yang menunjukkan posisi objek dengan bounding box dan tipe class dari sebuah objek dalam gambar. Jadi yang dilakukan oleh object recognition adalah mengembalikan beberapa objek yang terdeteksi beserta informasi terkait kelas objek, tingkat keyakinan, dan koordinat setiap objek.

5. Bounding Box

Bounding box pada sistem deteksi objek bertujuan untuk mengidentifikasi posisi dan jenis beberapa objek yang ada di dalam citra gambar. Pada implementasinya, dapat dilihat pada Gambar 3 *bounding box* berupa kotak- kotak pembatas yang ada di sekitar masker dan berbentuk persegi. Fungsinya adalah mementukan posisi, kelas, dan tingkat keyakinan (besar kemungkinan objek berada di lokasi).

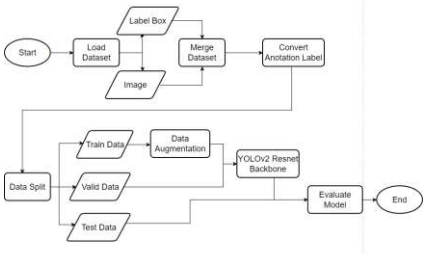


GAMBAR 3
Bounding Box

III. METODE

A. Sistem yang Dibangun

Sistem yang dirancang adalah sistem deteksi masker medis melalui gambar dengan model *deep learning* Yolo-v2 dengan *backbone* ResNet-50. Berikut adalah Gambar 4 mengenai desain arsitektur sistem deteksi masker dijelaskan pada subbab - subbab berikutnya.



GAMBAR 4
Pemodelan sistem deteksi masker

1. Dataset

Penelitian ini menggunakan dua dataset yaitu Medical Masks Dataset (MMD) yang tersedia pada situs kaggle berisi 670 gambar dan box anotasi dan Face Mask Dataset (FMD) yang tersedia pada situs kaggle yang berisi 853 gambar dan box anotasi. Anotasi tersebut dibedakan menjadi tiga kelas, yaitu with_mask, without_mask dan incorrect using mask. Sampel Face Mask Dataset dan Medical Mask Dataset diperlihatkan pada Gambar 5 dan Gambar 6



GAMBAR 5
Sampel *face mask dataset*



GAMBAR 6
Sampel *medical mask dataset*

2. Load Dataset dan Merge Dataset

Proses ini mencakup pemuatan dua dataset dari website kaggle ke google colabolatory. Selanjutnya dilakukan penggabungan kedua dataset. Box anotasi berupa file txt dan gambar juga disatukan menjadi satu kelas dataset.

3. Preprocessing

Pada repository YOLO darknet, file anotasi saat melakukan pelatihan model diharuskan menggunakan anotasi yolo yaitu txt file. Proses ini meliputi pengubahan format anotasi dari xml ke anotasi YOLO yaitu txt file.

4. Splitting Dataset

Splitting dataset dilakukan sehingga dataset menjadi tiga bagian yaitu data train, validation dan test. Hasil setelah melakukan data augmentasi dan splitting data adalah citra gambar dengan total 2.132 data train, 304 pada data validation dan 153 data test.

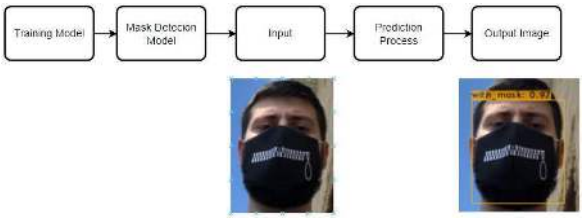
5. Data Augmentation

Selanjutnya dilakukan data augmentation untuk meningkatkan keragaman dataset untuk training detector. Data augmentation dilakukan dengan membalik gambar secara horizontal lalu mengubah brightness dan contrast pada citra secara random. Perubahan juga dilakukan dengan menambahkan blur pada citra dan mengubah secara random nilai hue, value dan saturation. Tujuan data augmentation tersebut adalah memperbanyak dataset masker dan menambah variasi gambar. Pada pemodelan augmentation dijalankan mengggunakan library Ablumentation sebagai augmentator gambar dan anotasi. Jumlah citra gambar setelah augmentasi data adalah 2.589 gambar

6. Pemodelan Yolo-v2 dengan backbone ResNet-50

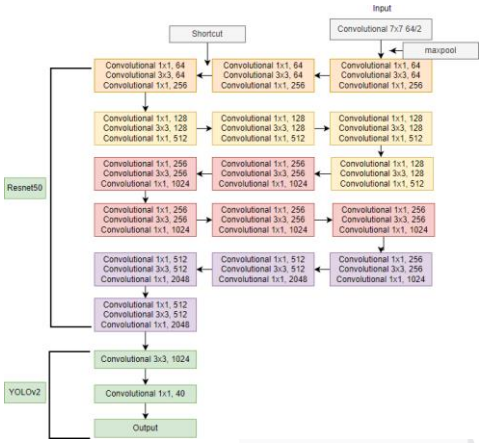
Pada tahap ini, pemodelan dilakukan dengan implementasi model Yolo-v2 dengan backbone ResNet-50. ResNet-50 digunakan sebagai backbone layer. Sedangkan Yolo-v2 digunakan sebagai pendeteksi masker pada gambar. Dalam pelatihan modelnya , terdapat 2132 data train dan

304 data validation untuk melatih model. Model yang dilatih menghasilkan sistem deteksi masker yang dapat memprediksi tiga kelas yaitu with_mask, without_mask dan incorrect using mask disertai bounding box dan confident score. Berikut adalah flowchart mengenai alur deteksi model pada Gambar 7.



GAMBAR 7
Proses Deteksi Masker

Sistem Yolo-v2 secara default memiliki 23 layer. Dibagi menjadi tiga yaitu 1 *input layer*, 21 *convolutional layer* dan 2 *fully connected layer*. Pengubahan model dilakukan pada penggantian 1 *input layer* dan 21 *convolutional layer* YOLO ke ResNet-50 dengan jumlah lima puluh *layer convolutional* beserta *shortcut layer* setelah 3 *convolutinal layer*, lalu dilanjutkan *fully connected layer* YOLO. Berikut ilustrasi dari pembuatan *model training* yang diperlihatkan pada Gambar 8.



GAMBAR 8
Yolo-v2 dengan *backbone* ResNet-50

7. Evaluasi Model

Pada tahap ini dilakukan evaluasi model dengan menggunakan *evaluation matrix* seperti: *average precision*, *precision*, *recall* dan *F1-Score* pada *training model* dan *testing model*. Evaluasi model dilakukan dengan membandingkan beberapa parameter konfigurasi saat *training model* di antaranya dengan mengubah nilai *learning rate* dan *batch.* Model Yolo-v2 *orginal* dengan Yolo-v2 dengan *backbone* ResNet-50 juga dibandingkan menggunakan *evaluation matrix* pada *data valdation* dan *data test*. Perhitungan kalkulasi tersebut diperlihatkan pada Rumus 2 terkait *precision*, Rumus 3 terkait *recall* dan Rumus 4 terkait F1-*Score*.

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

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

$$F1 - Score = 2 \cdot \frac{Precision * Recall}{Precision + Recall} \quad (4)$$

TP (*True Positif*) artinya data aktual benar diprediksi benar.
TN (*True Negatif*) artinya data aktual salah diprediksi salah.
FN (*False Positif*) artinya data aktual benar diprediksi salah.
FP (*False Positif*) artinya data aktual salah diprediksi benar.

IV. HASIL DAN PEMBAHASAN

a. Evaluasi

Pada penelitian ini, pengujian awal dilakukan dengan menggunakan model Yolo-v2 dengan backbone ResNet-50. Model ini dilatih menggunakan dataset Face Mask Detection dan Medical Mask Detection yang telah di augmentasi. Total dataset berisi sebanyak 2.589 citra gambar yang di bagi menjadi tiga bagian yaitu: training sebanyak 2.132 citra, validation sebanyak 304 citra dan testing sebanyak 153 citra. Di dalam dataset juga terdapat label yang terdiri dari tiga kelas yaitu: with_mask sebanyak 10.809 wajah, without_mask sebanyak 2.369 wajah dan incorrectly using mask sebanyak 411 wajah. Pembagian tersebut menandakan terdapatnya imbalance class pada dataset dikarenakan label wihout mask dan incorrectly using mask jauh lebih sedikit dibandingkan with mask. Berikut adalah jumlah label di setiap file anotasi yang ada di dalam dataset diperlihatkan pada Tabel 1.

TABEL 1
Label pada setiap *Class Dataset*

Class	*with_mask*	*without_mask*	*incorrectlyusing mask*
Testing	540 Wajah	100 Wajah	19 Wajah
Training	8.974 Wajah	1.984 Wajah	328 Wajah
Validation	1.295 Wajah	285 Wajah	64 Wajah

Pengujian dilakukan pada model Yolo-v2 dengan *backbone* ResNet-50 dengan menggunakan banyak skenario. Skenario ini dibentuk untuk mengevaluasi kualitas dan performansi model. Skenario ini dilakukan dengan mengubah parameter konfigurasi pada *training model*. Berikut adalah parameter konfigurasi yang diuji, diperlihatkan pada Tabel 2.

TABEL 2

Default

Batch Size	*Subdivision*	*Learning Rate*	IoUT
64	32	0,01	0,5

Configuration

1. Pengujian Skenario Pertama

Pada skenario pertama ini, pengujian Yolo-v2 dengan backbone ResNet-50 dilakukan dengan parameter konfigurasi secara default. Di mana jumlah *batch iteration* sebanyak 2000, *batch size* sebesar 64, *subdivision* sebesar 32, *learning rate* bernilai 0,01 dan *size image* 416 x 416 *pixel*. Selanjutnya performansi model diuji pada dua *dataset* yaitu *training* dan *validation*. Berikut adalah hasil pengujian dan prediksi model Yolo-v2 dengan *backbone* ResNet-50 dengan penggunaan IoU threshold sebesar 0,5 diperlihatkan pada Tabel 3.

TABEL 3
Default Model Yolo-v2 dengan *backbone* ResNet-50

Training Performance				*Validation Performance*		
Trained Images	*Precision*	*Recall*	F1-*Score*	*Precision*	*Recall*	F1-*Score*
128.000	0,75	0,71	0,73	0,75	0,74	0,74

Pada Tabel 3 terdapat hasil *training* dan *validation performance* dari model yang telah dilatih berupa nilai *precision*, *recall* dan F1-*Score*. Pelatihan model dilakukan hingga citra gambar yang dilatih mencapai 128.000 gambar. Pada *training performance*, model mendapatkan hasil F1-*Score* sebesar 0,73. Pada *validation performance*, model mendapatkan hasil F1-*Score* sebesar 0.74.

2. Pengujian Kedua

Pada skenario kedua ini, dilakukan *pengujian value learning rate* pada training model Yolo-v2 dengan *backbone* ResNet-50. Pengujian dilakukan dengan menggunakan nilai *batch size* sebesar 64 dan *subdivision* sebesar 32. Hasil pengujian dapat dilihat pada Tabel 4.

TABEL 4
Pengujian *learning rate* terhadap Yolo-v2 dengan *backbone* ResNet-50

Learning Rate	*Training Performance*				*Validation Performance*		
	Trained Images	*Precision*	*Recall*	F1-*Score*	*Precision*	*Recall*	F1-*Score*
0,001	140.000	0,74	0,69	0,72	0,75	0,74	0,74
0,003	128.000	**0,84**	0,69	**0,76**	**0,84**	0,74	**0,78**
0,005	128.000	0,73	**0,75**	0,74	0,72	**0,79**	0,75
0,007	128.000	0,81	0,70	0,75	0,81	0,73	0,77
0,01	128.000	0,75	0,71	0,73	0,75	0,74	0,74
0,03	128.000	0,78	0,71	0,75	0,75	0,73	0, 74

Pada Tabel 4 hasil pengujian dari perubahan *learning rate* pada saat *training model*. Pada *training* dan *validation performance*, *precision* dan F1-*Score* tertinggi berada di *learning rate* sebesar 0,003. Sedangkan *recall* tertinggi terdapat saat model menggunakan *learning rate* sebesar 0,005.

3. Pengujian Skenario Ketiga

Pada skenario ketiga, dilakukan penggantian nilai batch size dan subdivison saat training model. Pengujian dilakukan dengan menggunakan nilai batch size antara 64, 96, 128 dan 256. Sedangkan subdivision di setiap batch size adalah 32. Selain itu, model dilatih dengan menggunakan learning rate sebesar 0,003. Berikut hasil pengujian model nilai batch dan subdivison pada model Yolo-v2 dengan backbone ResNet-50 Tabel 5.

TABEL 5
Pengujian *Batch* dan *Subdivision* terhadap Yolo-v2 dengan *backbone* ResNet-50

Batch	*Training Performance*				*Validation Performance*		
	Iteration	*Precision*	*Recall*	F1-*Score*	*Precision*	*Recall*	F1-*Score*
64	2.000	0,84	0,69	0,76	0,84	0,74	0,78
96	2.000	0,77	0,76	0,77	0,77	0,79	0,78
128	2.000	**0,84**	0,76	0,80	0,84	0,80	0,82
256	2.000	0,82	**0,80**	**0,81**	**0,85**	**0,84**	**0,85**

Pada hasil pengujian *batch* pada Tabel 5. Pada *training* dan *validation performance* F1-*Score* tertinggi didapatkan saat menggunakan *batch size* 256 sebesar 0,81 dan 0,85.

4. Pengujian Skenario Keempat

Pada skenario keempat ini, dilakukan pengujian training model Yolo-v2 dengan backbone ResNet-50 menggunakan parameter konfigurasi nilai batch adalah 256, subdivision bernilai 32, learning rate sebesar 0,003 dan IoUT sebesar 0,5. Model tersebut dibandingkan dengan Yolo-v2 Original dari sisi validation dan testing dataset. Berikut adalah perbandingan model Yolo-v2 dengan backbone ResNet-50 dan Yolo-v2 Original yang diperlihatkan pada Tabel 6.

TABEL 6
Perbandingan *model* Yolo-v2 ResNet-50 *backbone* dan Yolo-v2 *Original*

Model	IoUT	*Iteration*	*Validation Performance*			*Testing Performance*		
			Precision	*Recall*	F1-*Score*	*Precision*	*Recall*	F1-*Score*
Yolo-v2 dengan *backbone* ResNet-50	0,5	2.000	**0,85**	0,84	**0,85**	**0,86**	0,82	**0,84**
Yolo-v2	0,5	2.000	0,80	**0,87**	0,83	0,77	0,82	0,80

Pada hasil perbandingan model Yolo-v2 *Original* dengan Yolo-v2 dengan *backbone* ResNet-50 pada Tabel 6. Nilai F1-*Score* tertinggi pada *validation performance* dan *testing performance* diraih oleh model Yolo-v2 dengan *backbone* ResNet-50 sebesar 0,85 dan 0,84.

5. Analisis Pengujian

a. Default Model Yolo-v2 dengan backbone ResNet-50

Berdasarkan hasil pengujian pada Tabel 2, YOLO-V2 dengan backbone ResNet-50 yang menggunakan default parameter configuration hanya F1-Score tertinggi sebesar 0,73 pada training performance dan 0,74 pada validation performance. Banyak parameter configuration yang dapat diubah untuk menaikan performansi model Yolo-v2 dengan backbone ResNet-50, contohnya mengubah nilai learning rate dan batch size saat traning model.

b. Pengaruh Perubahan Learning Rate untuk Training Model

Dari hasil pengujian pada Tabel 4, dilihat dari F1-Score pada training performance, nilai F1-Score cenderung menaik dari learning rate 0,001 hingga 0,007 dan turun jika learning rate lebih tinggi. Sedangkan dari validation performance hasil F1-Score tertinggi terdapat saat training model dengan parameter learning rate sebesar 0,003.

Dari hasil tersebut, maka Yolo-v2 dengan backbone ResNet-50 yang dilatih menggunakan learning rate dari 0,003 hingga 0,007 lebih optimal dibandingkan dengan learning rate pada konfigurasi bawaan yang bernilai 0,01. Pada pengujian skenario selanjutnya, model yang diuji menggunakan learning rate 0,003 dikarenakan mempunyai hasil F1-Score tertinggi di validation performance.

c. Pengaruh Nilai Batch dan Subdivision untuk Training Model

Dari hasil pengujian pada Tabel 5, perubahan nilai batch dan subdivision dapat meningkatkan performansi model. Hasil F1-Score tertinggi pada training dan validation peformance didapatkan saat model dilatih dengan jumlah batch size sebesar 256 dan subdivision sebesar 32. Semakin besar nilai batch maka semakin tinggi performansi model. Memperbesar nilai batch artinya memperbanyak citra gambar yang dilatih dalam satu batch iteration. Dengan nilai batch sebanyak 256 maka jumlah gambar yang dilatih pada 2.000 iterasi adalah 512.000 gambar, sedangkan pada batch sebanyak 68 jumlah gambar yang dilatih hanya mencapai 128.000 image saja. Lalu subdivision dengan nilai 32 membagi batch menjadi 32 bagian (minibatch), artinya dalam satu iterasi terdapat 128/32 = 4 gambar yang diproses di GPU dan diulang hingga 32 kali. Pada pengujian skenario selanjutnya, model yang diuji menggunakan batch size sebesar 256 dan subdivision sebesar 32 dikarenakan hasil F1-Score tertinggi di training dan validation performance.

d. Perbandingan Yolo-v2 dengan backbone ResNet dan Yolo-v2

Berdasarkan hasil pengujian skenario satu hingga empat, Yolo-v2 dengan backbone ResNet-50 dengan parameter yang optimal dibandingkan dengan Yolo-v2 original. Dari hasil pada Tabel 6, nilai model Yolo-v2 dengan backbone ResNet-50 memiliki nilai F1-Score yang sedikit unggul dibandingkan Yolo-v2 original pada testing performance.

V. KESIMPULAN

A. Kesimpulan

Berdasarkan hasil pengujian pada penelitian ini, Model Yolo-v2 dengan *backbone* Resnet dapat digunakan sebagai alat pendeteksi masker pada citra gambar dengan memanfaatkan *dataset Face Mask Dataset* dan *Medical Mask Dataset* sebagai *data training*, *validation* dan *testing*. Selain itu, pengubahan *learning rate* menjadi 0,003, *batch* menjadi 256 dan subdivison menjadi 32 saat melakukan *training* memiliki dampak yang cukup signifikan dalam peningkatan performansi model. Peningkatan performansi model tersebut dapat melampaui nilai pada model Yolo-v2 *Original* pada *testing performance*. Penelitian ini dapat memberikan skenario penggantian arsitektur Yolo-v2 dengan *backbone* ResNet-50. Penelitian ini menghasilkan nilai F1-*Score* tertinggi sebesar 84%. Penelitian ini juga bisa digunakan sebagai referensi dan pembanding penelitian untuk mendapatkan hasil evaluasi lebih lanjut.

REFERENSI

[1] "QA how is COVID-19 transmitted." https://www.who.int/indonesia/news/novel-coronavirus/qa/qa- how-is-covid-19-transmitted (accessed Jan. 31, 2023).

[2] World Health Organization, "Mask use in the context of COVID-19: interim guidance, 1 December 2020," World Health Organization, WHO/2019-nCoV/IPC_Masks/2020.5, 2020. Accessed: Jan. 31, 2023. [Online]. Available: https://apps.who.int/iris/handle/10665/337199

[3] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, Sep. 1999, vol. 2, pp. 1150–1157 vol.2. doi: 10.1109/ICCV.1999.790410.

[4] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004, doi: 10.1023/B:VISI.0000029664.99615.94.

[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587. Accessed: Jan. 31, 2023. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2014/html/Girshick_Rich_Feature_Hierarchies_2014_CVPR_paper
.html

[6] R. Girshick, "Fast R-CNN," presented at the Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1440–1448. Accessed: Jan. 31, 2023. [Online]. Available: https://openaccess.thecvf.com/content_iccv_2015/html/Girshick_Fast_R-CNN_ICCV_2015_paper.html

[7] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection

with Region Proposal Networks," in *Advances in Neural Information Processing Systems*, 2015, vol. 28. Accessed: Jan. 31,2023. [Online]. Available: https://proceedings.neurips.cc/paper/2015/hash/14bfa6bb14875e45bba028a21ed38046-Abstract.html

[8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 779–788. Accessed: Jan. 31, 2023. [Online]. Available: https://www.cv-foundation.org/openaccess/content_cvpr_2016/html/Redmon_You_Only_Look_CVPR_2016_paper.html

[9] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 7263–7271. Accessed: Jan. 31, 2023. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2017/html/Redmon_YOLO9000_Better_Faster_CVPR_2017_paper.html

[10] Y. Lee, C. Lee, H.-J. Lee, and J.-S. Kim, "Fast Detection of Objects Using a YOLOv3 Network for a Vending Machine," in *2019 IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, Mar. 2019, pp. 132–136. doi: 10.1109/AICAS.2019.8771517.

[11] W. Liu *et al.*, "SSD: Single Shot MultiBox Detector," in *Computer Vision – ECCV 2016*, Cham, 2016, pp. 21–37. doi: 10.1007/978-3-319-46448-0_2.

[12] M. Loey, G. Manogaran, M. H. N. Taha, and N. E. M. Khalifa, "Fighting against COVID-19: A novel deep learning model based on YOLO-v2 with ResNet-50 for medical face mask detection," *Sustainable Cities and Society*, vol. 65, p. 102600, Feb. 2021, doi: 10.1016/j.scs.2020.102600.

[13] B. Mandal, A. Okeukwu, and Y. Theis, "Masked Face Recognition using ResNet-50." arXiv, Apr. 18, 2021. doi: 10.48550/arXiv.2104.08997. G. Kaur *et al.*, "Face mask recognition system using CNN model," *Neuroscience Informatics*, vol. 2, no. 3, p. 100035, Sep. 2022, doi: 10.1016/j.neuri.2021.100035.