Deteksi Penggunaan Masker Pada Citra Menggunakan YOLOv5 Dengan CNN

1st Maulana Nur Hidayah
Fakultas Informatika
Universitas Telkom
Bandung, Indonesia
maulananur@students.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* — Pandemi COVID-19 bermula di negara China lebih tepatnya kota Wuhan pada tahun 2019. Pemerintah Indonesia menerapkan kebijakan demi memutus penularan virus ini, di antaranya penggunaan masker di ruang publik. Selama ini penggunaan masker diperiksa secara manual oleh petugas. Cara ini memiliki banyak keterbatasan salah satunya sulit untuk dilakukan pada berbagai waktu dan tempat, sehingga perlu dibuat sistem pendeteksi masker berbasis visi komputer dengan tujuan untuk menutupi kekurangan pendeteksian masker tradisional. Penelitian ini mengusulkan pembuatan sistem deteksi masker menggunakan algoritma *You Only Look Once* versi 5 (YOLOv5) sebagai metode pendeteksi wajah dan *Convolutional Neural Network (CNN)* sebagai metode klasifikasi penggunaan masker. Hasil dari klasifikasi penggunaan masker pada skenario terbaik mendapatkan nilai *f1-score* 98%, pada data *testing* mendapatkan akurasi 97,88%.**

***Kata kunci*— deteksi masker, CNN, citra digital, YOLOv5**

I. PENDAHULUAN

A. Latar Belakang

COVID-19 ialah sebuah pandemi yang diakibatkan dari virus severe acute respiratory syndrome coronavirus 2, pandemi ini bermula di negara China lebih tepatnya kota Wuhan pada tahun 2019. WHO menghimbau pemerintah di setiap negara agar melakukan tindakan untuk mengurangi penyebaran virus ini. Himbauan yang dilakukan di antaranya menerapkan protokol kesehatan dengan selalu menjaga jarak, menjaga kebersihan, dan menggunakan masker ketika beraktivitas. Berdasarkan Keputusan Menteri Kesehatan (Kemenkes) Republik Indonesia nomor HK 01.07/MENKES/328/2020 masyarakat bisa beraktivitas di luar rumah jika melaksanakan protokol kesehatan dengan disiplin, di antaranya dengan menggunakan masker pada saat beraktivitas diruang publik[1]. Penggunaan masker bisa untuk menunda dan menurunkan penyebaran pandemi, tetapi bergantung juga pada media penularan virus, efisiensi masker (kualitas masker), dan cakupan penggunaan masker di masyarakat[2]. Berdasarkan hasil observasi lapangan masih terdapat masyarakat yang tidak patuh dalam menggunakan masker saat beraktivitas. Ketidakpatuhan ini merupakan salah satu hal yang mengkhawatirkan dan tidak boleh untuk dibiarkan.

Berbagai cara dilakukan untuk menegakkan protokol kesehatan di antaranya adalah pemeriksaan masker oleh petugas yang berwenang di berbagai tempat umum, namun cara seperti ini masih manual atau masih menggunakan manusia. Hal ini memiliki celah karena pada dasarnya setiap manusia berbeda dan manusia berbeda pula dengan mesin. Dalam rangka mengurangi celah pelanggaran protokol kesehatan yang ada, maka dibuatkanlah sebuah sistem yang dapat mendeteksi penggunaan masker pada manusia. Pembuatan sistem dilakukan dengan menggunakan cabang ilmu pengetahuan Computer Vision khususnya pada bidang Object detection.

Tersedia beberapa metode untuk mengklasifikasikan dan mengidentifikasi objek pada sebuah citra, CNN salah satunya. CNN banyak diimplementasikan karena akurasinya yang relatif tinggi dan memberikan hasil pengenalan citra yang baik jika dibandingkan dengan pengklasifikasi ORB dan SVM[3]. CNN memiliki kekurangan yaitu tidak dapat mengklasifikasi objek lebih dari satu dalam sebuah citra[4]. YOLOv5 merupakan metode yang baik dan cepat untuk mendeteksi objek[5]. Algoritma YOLOv5 dan CNN digunakan karena algoritma CNN tidak bisa mendeteksi objek lebih dari satu pada sebuah citra, sementara YOLOv5 bisa memperbaiki kekurangan CNN karena YOLOv5 mampu mendeteksi lebih dari satu objek pada sebuah citra. Pembuatan sistem ini bertujuan untuk mengimplementasikan metode CNN dengan dibantu metode YOLOv5 untuk memprediksi apakah wajah pada citra menggunakan masker atau tidak. YOLOv5 digunakan untuk mendeteksi keberadaan objek wajah pada sebuah citra, kemudian objek wajah diklasifikasikan oleh CNN apakah menggunakan masker atau tidak.

B. Topik dan Batasannya

Topik pada penelitian ini adalah membangun sebuah sistem yang mendeteksi penggunaan masker pada sebuah citra. Pertama, dilakukan pendeteksian objek wajah pada suatu citra yang dilakukan menggunakan algoritma YOLOv5 setelah itu algoritma CNN mengklasifikasikan penggunaan

masker pada objek wajah tersebut, penelitian ini juga melakukan analisis terhadap performansi model CNN yang di buat. Batasan pada pengerjaan tugas ini adalah penelitian berfokus pada beberapa parameter yang mempengaruhi performansi CNN dalam melakukan klasifikasi, sementara YOLOv5 menggunakan parameter *default* dari YOLOv5. Kondisi pencahayaan pada citra yang dideteksi tidak boleh redup karena bisa mempengaruhi performa dari model.

C. Tujuan

Melakukan studi metode YOLOv5 untuk mendeteksi penggunaan masker

1. Mendeteksi penggunaan masker pada citra menggunakan YOLOv5 dan CNN
2. Bagaimana tingkat akurasi model CNN dalam mendeteksi penggunaan masker

D. Organisasi Tulisan

Pendahuluan yang berisi uraian masalah yang menjadi topik bahasan. Studi Terkait menjelaskan teori dan *review* jurnal atau makalah yang berkaitan dengan topik yang diangkat. Sistem yang dibangun merupakan penjelasan mengenai sistem yang dibangun. Evaluasi merupakan bagian untuk menampilkan hasil dari *sistem*. Kesimpulan berisi kesimpulan serta saran pada penelitian selanjutnya.

II. KAJIAN TEORI

A. Studi Terkait

1. Studi literatur

Penelitian tentang deteksi masker pada wajah manusia mulai ramai pada saat terjadinya pandemi COVID-19 pada tahun 2019.

A. G. Sandesara, D. Joshi, dan S. Joshi (2020) membangun model pendeteksian masker menggunakan model CNN. Model dilatih menggunakan bantuan Adam optimizer dan binary cross entropy sebagai loss function. Penelitian ini mendapatkan akurasi training 97,5%, pada data validation akurasi sebesar 95%[6].

U. Upadhyay, B. Rudra, dan dkk (2022) mengusulkan penggunaan deep learning untuk mendeteksi penggunaan masker. Dataset terdiri dari dua kategori menggunakan masker dan tidak menggunakan masker. Metode yang diusulkan adalah Convolution Neural Network (CNN). Penelitian ini menghasilkan akurasi 99,1%[7].

O. Sevli dan N. Kemaloğlu (2020) melakukan penelitian mengenali bahasa isyarat Turki. Penelitian berfokus untuk mengetahui performa penggunaan optimizer pada CNN. Hasil dari penelitian tersebut dapat disimpulkan penggunaan Adam optimizer menghasilkan performa yang cukup baik di mana akurasi training 98,42% dan akurasi Testing 98,55% dibandingkan dengan penggunaan SGD, RMSprop, dan Adamax[8].

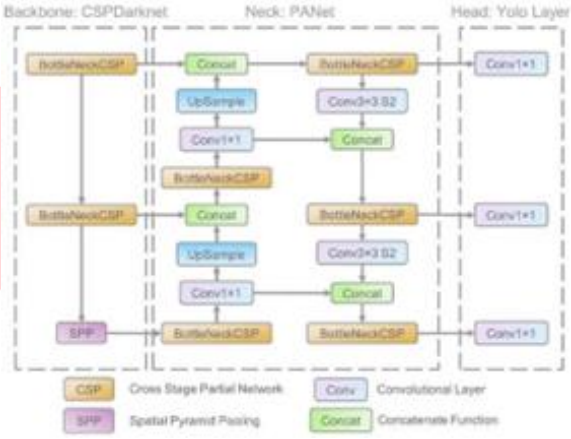
Raj S, Gupta Y, Malhotra R (2022) melakukan penelitian mengenali nomor plat kendaraan menggunakan metode YOLOv5 dan CNN. Penelitian ini dilakukan menggunakan YOLOv5 sebagai pendeteksi pada plat mobil, sementara CNN digunakan untuk pendeteksian nomor plat kendaraan[9].

Yang G, Feng W, dan dkk (2020) melakukan penelitian mengenali penggunaan masker menggunakan algoritma YOLOv5. YOLOv5 digunakan karena memiliki performa yang baik dalam mendeteksi objek. Sistem yang dibangun memiliki akurasi testing 97,9%[10].

Polsinelli M, Cinque L, dan Placidi G. (2020) membangun model pendeteksian penyakit COVID- 19 pada hasil CT scans menggunakan algoritma CNN. Arsitektur CNN yang digunakan menggunakan arsitektur SqueezeNet. Model dilateh menggunakan 20 epochs. Penelitian ini menghasilkan akurasi testing sebesar 85,03% dan F1-Score 86,20%[11].

2. YOLOv5

Berikut adalah penggambaran arsitektur YOLOv5 pada gambar 2.1[12].

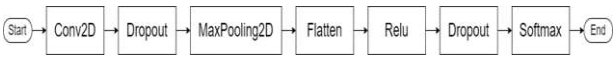


GAMBAR 2.1
Visualisasi arsitektur YOLOv5

YOLOv5 adalah algoritma pengenalan objek generasi kelima yang dirilis pada April 2020. Secara garis besar arsitektur YOLOv5 ini hanya memiliki sedikit perbedaan jika dibandingkan dengan generasiYOLO sebelumnya. YOLOv5 hanya dapat dijalankan menggunakan bahasa pemrograman Python. Penggunaan bahasa python memiliki kelebihan mudah dalam proses instalasi dan integrasi pada perangkat IoT. Arsitektur YOLOv5 mencakup pembaruan terkini yang mirip dengan arsitektur YOLOv4, jadi secara teori seharusnya tidak ada banyak perbedaan. Pengembang YOLOv5 tidak menerbitkan dokumen terperinci, pengembang hanya menerbitkan repositori di Github dan memperbaruinya di repositori tersebut. Menganalisis struktur kode yang ada pada *file* .yaml, model YOLOv5 memiliki *backbone Focus structure* dan CSP *network*, *neck* menggunakan SPP dan PANet, dan *head* menggunakan *GIoU-loss*. Glenn Jocher mengintegrasikan proses pemilihan *anchor box* ke dalam YOLOv5[13]. Oleh karena itu, model otomatis mempelajari *anchor box* terbaik dalam kumpulan *dataset* dan menggunakannya selama proses pelatihan. YOLOv5 digunakan karena memiliki akurasi dan kecepatan yang lebih baik jika dibandingkan dengan YOLOv3 dan MobileNet-SSDV[14]. Peran YOLOv5 pada penelitian ini adalah untuk mendeteksi objek wajah pada citra. Model yang akandigunakan adalah model YOLOv5s.

3. CNN

Arsitektur algoritma CNN divisualisasikan pada gambar 2.2.



GAMBAR 2.2
Visualisasi arsitektur *sequential* CNN

Algoritma *convolutional neural network* (CNN) menjadi algoritma yang sedang populerbelakangan ini. CNN adalah tipe *neural network* didedikasikan dalam pemrosesan data yang memiliki *topology* model *grid*. Citra merupakan contoh dari data tersebut. Citra merupakan model data yang bisa disebut sebagai *grid* piksel 2 dimensi. CNN menggunakan operasi perhitungan yang disebut konvolusi. Konvolusi merupakan jenis operasi linier[16]. Penelitian ini menggunakan model *deep learning* CNN. Metode *deep learning* dipilih karena memiliki kinerja yang lebih baik daripada menggunakan model *shallow learning* pada *dataset* yang cukup besar[17]. Penelitian ini menggunakan arsitektur CNN *sequential*. Peran CNN pada penelitian ini untuk klasifikasi objek wajah pada citra.

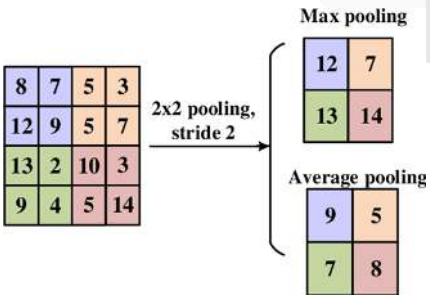
4. *Convolutional layer* + ReLU

Seperti namanya, *convolutional layer* memainkan peran penting dalam CNN. Data yang dimasukkan pada lapisan konvolusional akan dilakukan proses konvolusi. Lapisan konvolusional memiliki beberapa proses, yaitu *input* data, filter, dan *feature map* atau *activation map*. *Input* citra memiliki panjang, tinggi, dan tebal. Lapisan mengubah tiap filter menuju semua bagian data *input* dan membuat *feature map*. Setiap filter melewati operasi pergeseran dan operasi perhitungan antara *input data* dan nilai dari filter. Hasil dari proses konvolusi adalah target piksel dengan *output array* [18].

Setelah proses *convolution* selesai dilakukan proses *activation function* menggunakan ReLU. ReLU (*Rectified Linier Unit*) digunakan untuk meningkatkan non-linearitas pada citra dengan mengganti nilai negatif menjadi 0 dan nilai positif kembali ke nilai asal.

5. Pooling Layer

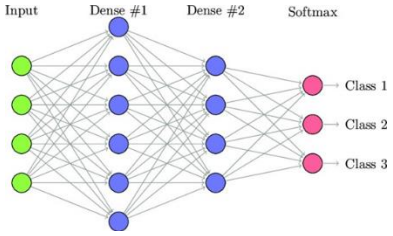
Lapisan ini digunakan untuk mengurangi kompleksitas komputasi, lapisan *pooling* dirancang untuk mengecilkan dimensi *feature maps*. *Pooling layer* ini berisi filter dengan besaran *filter* dan *stride*, yang akan bergerak di seluruh *activation map*. *Pooling* yang umum dipakai untuk CNN adalah *average pooling* dan *max pooling*. A*verange pooling* menggunakan besaran rata - rata suatu citra, sementara *max pooling* menggunakan besaran terbesar suatu citra. Pada penelitian ini menggunakan *max pooling* digunakan sebagai *pooling layer*. *Pooling layer* digambarkan pada gambar 2.4[19].



GAMBAR 2.3
Gambaran *Pooling Layer* pada CNN

6. *Fully-Conected Layer*

Lapisan *fully-Connected* berfungsi untuk menghubungkan node input dengan node *softmax* sebagai *activation function*. Gambar 2.5 merupakan gambaran dari *fully connected layer*[20].



GAMBAR 2.4
Gambaran fully connected layer pada CNN

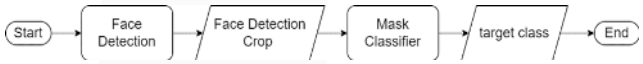
Feature map yang dibuat oleh lapisan konvolusi dan lapisan *pooling* masih berupa *array* multidimensi, sehingga *activation map* harus dilakukan *reshape* menjadi vektor untuk digunakan sebagai *input* lapisan *fully-connected layer*. Lapisan ini melakukan klasifikasi menggunakan fungsi aktivasi *softmax*. Hasil akhir dari *softmax* adalah nilai probabilitas dengan skala nol sampai satu.

III. METODE

A. Sistem yang Dibangun

1. Desain Sistem

Sistem yang dibangun mempunyai 2 tahapan. Tahapan yang pertama adalah pendeteksian wajah menggunakan YOLOv5, tahapan yang kedua adalah klasifikasi penggunaan masker menggunakan CNN. Berikut adalah pemodelan sistem deteksi pada gambar 3.1.



GAMBAR 1.1
Pemodelan sistem deteksi masker

2. Dataset

Dataset memiliki jumlah 7,553 buah citra, *dataset* ini digunakan untuk klasifikasi penggunaan masker pada citra wajah menggunakan CNN. Dataset berisi kumpulan penggunaan masker dan tidak menggunakan masker. Setiap data memiliki satu objek penggunaan masker. Dataset dibagi menjadi 2 kelas "without_mask" dan "with_mask". *Dataset* kemudian dibagi menjadi 80% data *train set*, 10% data *validation set*, dan 10% data *testing set*. Dataset bersumber dari situs kaggle[21]. Gambar 3.2 merupakan sampel dari *dataset*.



(a) (b)

GAMBAR 3.2
kelas "without_mask"; (b) kelas "with_mask"

3. Algoritma *YOLOv5* dan CNN

Algoritma YOLOv5 dan CNN digunakan secara berurutan. Gambar 3.2 merupakan alur dari pemrosesan dari algoritma YOLOv5 dan CNN.



GAMBAR 3.3
Alur pemrosesan YOLOv5 dan CNN

Algoritma YOLOv5 di *train* menggunakan *dataset* YOLOv5 menggunakan parameter *default* yang sudah disediakan dengan *input size* 416, *batch size* 8 dan menggunakan nilai *epoch* 50 . Hasil dari proses *train* kemudian menghasilkan *file* bernama "last.pt" yang merupakan hasil dari *weights* dari *epoch training* terakhir. Kemudian program dijalankan menggunakan nilai dari *weight*. AlgoritmaYOLOv5 menerima *input* citra kemudian dari citra tersebut dilakukan pendeteksian keberadaan wajah, setelah keberadaan wajah terdeteksi, maka citra wajah menghasilkan *bounding box*, hasil dari *bounding box* tersebut di potong oleh algoritma, hasil di pemotongan wajah tersebut menjadi *output* dari YOLOv5. Hasil dari YOLOv5 kemudian diteruskan kepada algoritma CNN untuk dilakukan klasifikasi citra. Algoritma melakukan proses *data preprocessing* dengan mengubah ukuran citra pada *dataset* menjadi 244x244 pixel, *dataset* kemudian diberi label *'with_mask'* untuk *dataset* menggunakan masker dan *'without_mask'* untuk tidak menggunakan masker. Pembuatan tiap lapisan konvolusi menggunakan ReLU sebagai *activation function* dan *max p*ooling sebagai *pooling layer*, *flatten* dan *dense* jugadigunakan untuk menurunkan dimensi, *dropout* digunakan untuk mencegah *overfitting*, *optimizer* yang digunakan adalah Adam, *Loss* yang digunakan adalah *binary cross entrophy*, beberapa parameter yang diuji seperti jumlah layer, *batch size*, dan *learning rate* dibahas pada bab 4. Kemudian CNN melakukan proses *training* menggunakan *dataset* dengan perbandingan 80% data *train set*, 10% data *testing set*, dan 10% *validation set*. Setelah proses *training* selesai citra wajah dari algoritma YOLOv5 di klasifikasi oleh CNN.

4. Evaluasi Kinerja Sistem CNN

Evaluasi kinerja sistem difokuskan pada bagian CNN untuk mengukur seberapa baik kinerja sistem. Sistem evaluasi kinerja CNN dihitung menggunakan nilai *confusion matrix* dari proses *testing*. Tabel 3.1 menunjukkan *confusion matrix*.

TABEL 3.1
Tabel *confusion matrix*

		Aktual	
		Positif	Negatif
Prediksi	Positif	*True Positive*	*False Positive*
	Negatif	*False Negative*	*True Negative*

Keterangan dari tabel 3.1:
True Positive (TP) : Di mana nilai aktualnya benar dan model memprediksi nilai benar.
True Negative (TN) : Di mana nilai aktualnya salah dan model memprediksi nilai salah.
False Positive (FP) : Di mana nilai aktualnya benar dan model memprediksi nilai salah.
False Negative (FN) : Di mana nilai aktualnya salah dan model memprediksi nilai benar. Terdapat 4 indikator untuk mengukur kinerja sistem, di antaranya akurasi, presisi, *recall*, dan *f-1 score.*
Akurasi : Akurasi dihitung untuk menghitung rasio prediksi yang benar (positif dan negatif).Nilai akurasi didapatkan dengan menggunakan perhitungan rumus 3.1.

$$\text{akurasi} = \frac{TP + TN}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

RUMUS 3.1
Akurasi

Presisi : Presisi adalah jumlah estimasi data yang benar dibandingkan dengan total hasil prediksi positif. Nilai presisi dapat dihitung dengan menggunakan rumus 3.2.

$$\text{presisi} = \frac{TP}{\text{TP} + \text{FP}}$$

RUMUS 3.2
Presisi

Recall : *Recall* merupakan perbandingan prediksi positif yang benar terhadap semua data positif yang benar. Nilai *recall* bisa dihitung dengan menggunakan rumus 3.3.

$$recall = \frac{TP}{\text{TP} + \text{FN}}$$

RUMUS 3.3
R*ecall*

F1-score : *f1-score* adalah perhitungan penilaian yang menggabungkan *recall* dan presisi. *f1- score* dapat dihitung menggunakan rumus 3.4.

$$\text{f1} - score = 2 * \frac{(recall * \text{presisi})}{(\text{recall} + \text{presisi})}$$

RUMUS 3.4
f1- score

IV. HASIL DAN PEMBAHASAN

A. Evaluasi

1. Skenario Pengujian Sistem

a. Pengaruh Layer Konvolusi

Pada skenario satu, dilakukan pengujian terhadap jumlah layer konvolusi pada Convolutional Neural Network. Pengujian pertama ini menggunakan tujuh jenis lapisan layer, yaitu: satu layer, tiga layer, lima layer dan tujuh layer. Penggunaan layer ditujukan untuk proses ekstraksi ciri pada citra. Menggunakan berbagai jenis layer dapat memengaruhi akurasi dan loss dari suatu model.

b. Pengaruh Batch Size

Pada skenario dua, dilakukan pengujian terhadap pemilihan nilai batch size. Pada pengujian ini digunakan tiga ukuran batch sebagai parameter, yaitu 8, 16, 32, dan 64.

c. Pengaruh Learning Rate

Pada skenario tiga, dilakukan pengujian terhadap pemilihan nilai learning rate. Pada pengujian ini digunakan tiga ukuran learning rate sebagai parameter, yaitu 0,01, 0,001, 0,0001, 0,00001.

2. Hasil Pengujian Skenario

Sistem menggunakan *baseline* model yang diperlihatkan pada tabel 4.1.

TABEL 4.1
Baseline parameter model

Conv2D	*Batch Size*	*Learning Rate*
1	32	0,001

Tabel 4.1 menunjukkan ukuran parameter *baseline* model yang digunakan dalam pengujian model.

a. Skenario Satu: Jumlah Layer Konvolusi

Pada skenario satu, dilakukan pengujian terhadap jumlah layer konvolusi pada *Convolutional Neural Network*. Pengujian pertama ini menggunakan empat jenis lapisan *layer*, yaitu: satu *layer*, tiga *layer*, lima *layer,* dan tujuh *layer*. Penggunaan layer dimaksudkan untuk proses ekstraksi fitur pada gambar. Tabel 4.2 menunjukkan hasil untuk skenario satu.

TABEL 4.2
Eksperimen pada penambahan lapisan Conv2D

Conv2D	*Computing Time (s)*	*F-1 score*	*Training Accuracy*	*Testing Accuracy*
1	459	91%	93,38%	91,52%
3	545	96%	97,48%	95,89%
5	461	96%	98,80%	96,42%
7	**482**	**97%**	**97,35%**	**97,21%**

Penggunaan berbagai jenis layer dapat mempengaruhi tingkat *accuracy* dari suatu model. Berdasarkan tabel 4.2 didapatkan hasil layer Conv2D terbaik yaitu menggunakan 7 layer dengan akurasi *testing* 97,35%. Hasil terbaik digunakan untuk percobaan selanjutnya.

b. Skenario Dua: Jumlah *Batch Size.*

Pada skenario dua, dilakukan pengujian terhadap pemilihan nilai batch size. Pada pengujian ini digunakan empat ukuran batch sebagai parameter, yaitu 8, 16, 32, dan 64. Tabel 4.3 menunjukkan hasil untuk skenario dua.

TABEL 4.3
Eksperimen pada *batch size*

Batch Size	*Computing Time (s)*	*F-1 score*	*Training Accuracy*	*Testing Accuracy*
8	577	96%	96,69%	96,42%
16	515	96%	97,75%	96,15%
32	482	97%	97,35%	97,21%
64	**462**	**98%**	**97,88%**	**97,88%**

Penggunaan berbagai ukuran *batch size* dapat mempengaruhi tingkat *accuracy* dari suatu model. Berdasarkan tabel 4.3 didapatkan hasil penggunaan *batch size* terbaik yaitu menggunakan 64 *batch* dengan akurasi *testing* 97,88%. Hasil terbaik digunakan untuk percobaan selanjutnya.

c. Skenario Tiga: Pengaruh *Learning Rate*

Pada skenario tiga, dilakukan pengujian terhadap pemilihan nilai *learning rate*. Pemilihan *learning rate* harus disesuaikan dengan nilai *epoch* yang digunakan, jika pemilihan nilai terlalu kecil dapat mengakibatkan model mengalami *stuck* sementara pemilihan terlalu besar dapat mengakibatkan model memiliki *weights* yang tidak optimal atau proses *training* yang tidak stabil. Pada pengujian ini digunakan empat ukuran *learning rate* sebagai parameter, yaitu 0,01, 0,001, 0,0001, 0,00001. Tabel 4.4 menunjukkan hasil untuk skenario tiga.

TABEL 4.4
Eksperimen pada *learning rate*

Learning Rate	*Computing Time (s)*	*F-1 score*	*Training Accuracy*	*Testing Accuracy*
0,01	619	68%	47,35%	51,92%
0,001	**462**	**98%**	**97,88%**	**97,88%**
0,0001	647	96%	96,82%	96,42%
0,00001	610	90%	89,81%	90,33%

Penggunaan berbagai ukuran *learning rate* dapat mempengaruhi akurasi model. Berdasarkan tabel 4.4 didapatkan hasil penggunaan *learning rate* terbaik yaitu menggunakan *learning rate* 0,001 dengan akurasi *testing* 97,88%.

3. Analisis Hasil Pengujian

Pengujian sistem dilakukan dengan menggunakan *epoch* 20 dan menguji 15 citra yang didapatkan dari hasil pendeteksian YOLOv5 berupa citra wajah yang sudah dipotong. Pengujian awal model dilakukan dengan menggunakan parameter berikut Conv2D 1, *batch size* 16, *Learning Rate* 0.001. Hasil dari skenario satu dapat disimpulkan dengan menggunakan jumlah tujuh layer terbukti memiliki akurasi terbaik dengan nilai *f-1 score 97%,* akurasi *testing* 97,35%, hasil akurasi terbaik digunakan untuk eksperimen selanjutnya. Penggunaan tujuh layer dapat

menghasilkan nilai yang baik karena menambahkan lebih banyak *layer* mengekstrak lebih banyak fitur sehingga bisa menambah akurasi sistem. Pengujian kedua dilakukan dengan memodifikasi nilai *batch size*, hasil dari skenario dua dapat disimpulkan dengan menggunakan *batch size* 64 terbukti terdapat peningkatan nilai akurasi menjadi *f-1 score* 98%, akurasi *testing* 97,88%. Penggunaan *batch size* terbukti dapat meningkatkan akurasi, hal ini dapat dijelaskan bahwa semakin besar *batch size* yang digunakan mengakibatkan semakin besar waktu dan memori perangkat yang digunakan untuk proses *training* sehingga meningkatkan akurasi model.

Pengujian ketiga dilakukan untuk mengetahui efek perubahan parameter *learning rate* terhadap performa model. Hasil dari skenario tiga dapat disimpulkan jika digunakan nilai *learning rate* 0,001 mendapatkan nilai terbaik dengan nilai *f-1 score* 98%, akurasi *testing* 97,88%. Penggunaan *learning rate* yang optimal terbukti dapat meningkatkan akurasi, hal ini terjadi karena pemilihan *learning rate* harus sesuai dengan jumlah *epoch* yang digunakan. Pada bagian kolom *Computing Time* pada tabel 4.2, 4.3, dan 4.4 ditarik kesimpulan bahwa tidak selamanya akurasi yang terbaik membutuhkan waktu komputasi besar.

Analisis terhadap klasifikasi yang tidak tepat dilakukan untuk mengetahui kondisi bagaimana sistem tidak tepat dalam mengklasifikasikan penggunaan masker. Gambar 4.1 merupakan contoh pengklasifikasian tidak tepat oleh sistem.



GAMBAR 4.1
Contoh pengklasifikasian tidak tepat

Analisis terhadap klasifikasi yang tidak tepat dilakukan untuk mengetahui kondisi bagaimana sistem salah dalam mengklasifikasikan penggunaan masker. Gambar 4.1 merupakan contoh pengklasifikasian tidak tepat oleh sistem. Pengklasifikasian tidak tepat ini dikarenakan kurangnya *training set* yang memiliki citra wajah menghadap ke samping sehingga sistem tidak optimal dalam mengklasifikasikan penggunaan masker jika kondisi wajah menghadap ke samping.

V. KESIMPULAN

A. Kesimpulan

Pada penelitian ini dibangun sistem deteksi masker pada citra menggunakan YOLOv5 dan CNN. Berdasarkan hasil uji pada model CNN menggunakan parameter lapisan layer konvolusi Conv2D 7, batch size 64, dan learning rate 0,001, menghasilkan akurasi testing 97,88% dan f1-score 98%. Dapat diambil kesimpulan bahwa ukuran parameter dapat berpengaruh terhadap model dalam melakukan pendeteksian masker. Angka hasil akurasi dalam penelitian yang dilakukan belum tentu berlaku untuk dataset yang lain. Karena parameter yang digunakan dapat menghasilkan output yang berbeda pada dataset yang lain.

Saran untuk penelitian selanjutnya adalah mencari pengaruh parameter – parameter lain untuk mendapatkan akurasi yang lebih baik seperti parameter optimizer yang digunakan. Penambahan variasi dataset perlu dilakukan pada penelitian selanjutnya supaya sistem dapat mengenali berbagai kondisi citra wajah. Penelitian ini juga bisa digunakan sebagai bahan perbandingan penelitian untuk mendapatkan hasil evaluasi yang lebih optimal.

REFERENSI

[1] "KEPUTUSAN MENTERI KESEHATAN REPUBLIK INDONESIA," May 20, 2020. http://hukor.kemkes.go.id/uploads/produk_hukum/KMK_No HK_01_07-MENKES-328-2020_ttg_Panduan_Pencegahan_Pengendalian_COVID-19_di_Perkantoran_dan_Industri.pdf (accessed Jan. 26, 2023).

[2] S. E. Eikenberry et al., "To mask or not to mask: Modeling the potential for face mask use by the general public to curtail the COVID-19 pandemic," Infect Dis Model, vol. 5, pp. 293–308, 2020.

[3] S. S. Yadav and S. M. Jadhav, "Deep convolutional neural network based medical image classification for disease diagnosis," J Big Data, vol. 6, no. 1, pp. 1–18, 2019.

[4] P. Sharma, "Image Classification vs Object Detection vs Image Segmentation," Medium (20 August)[Online], 2019.

[5] G. Yang et al., "Face mask recognition system with YOLOV5 based on image recognition," in 2020 IEEE 6th International Conference on Computer and Communications (ICCC), 2020, pp. 1398–1404.

[6] A. G. Sandesara, D. D. Joshi, and S. D. Joshi, "Facial mask detection using stacked CNN model," Int. J. Sci. Res. Comput. Sci. Eng. Inform. Technol, 2020.

[7] U. Upadhyay, B. Rudra, U. Upadhyay, U. Upadhyay, and B. Rudra, "Face Mask Detection Using Convolutional Neural Network (CNN)," EasyChair, 2021.

[8] O. Sevli and N. Kemaloğlu, "Turkish sign language digits classification with CNN using different optimizers," International Advanced Researches and Engineering Journal, vol. 4, no. 3, pp. 200–207, 2020.

[9] S. Raj, Y. Gupta, and R. Malhotra, "License plate recognition system using yolov5 and cnn," in 2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS), 2022, vol. 1, pp. 372–377.

[10] G. Yang et al., "Face mask recognition system with YOLOV5 based on image recognition," in 2020 IEEE 6th International Conference on Computer and Communications (ICCC), 2020, pp. 1398–1404.

[11] M. Polsinelli, L. Cinque, and G. Placidi, "A light CNN for detecting COVID-19 from CT scans of the chest," Pattern Recognit Lett, vol. 140, pp. 95–100, 2020.

[12] R. Xu, H. Lin, K. Lu, L. Cao, and Y. Liu, "A forest fire detection system based on ensemble learning," Forests, vol. 12, no. 2, p. 217, 2021.

[13] D. Thuan, "Evolution of Yolo algorithm and Yolov5: The State-of-the-Art object detention algorithm," 2021.
[14] R. Iyer, P. S. Ringe, and K. P. Bhensdadiya, "Comparison of YOLOv3, YOLOv5s and MobileNet-SSD V2 for real-time mask detection," Artic. Int. J. Res. Eng. Technol, vol. 8, pp. 1156–1160, 2021.
[15] C. G. Pachón, D. M. Ballesteros, and D. Renza, "Fake banknote recognition using deep learning," Applied Sciences, vol. 11, no. 3, p. 1281, 2021.
[16] J. Wu, "Introduction to convolutional neural networks," National Key Lab for Novel Software Technology. Nanjing University. China, vol. 5, no. 23, p. 495, 2017.
[17] E. R. DeLancey, J. F. Simms, M. Mahdianpari, B. Brisco, C. Mahoney, and J. Kariyeva, "Comparing deep learning and shallow learning for large-scale wetland classification in Alberta, Canada," Remote Sens (Basel), vol. 12, no. 1, p. 2, 2019.
[18] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," arXiv preprint arXiv:1511.08458, 2015.
[19] H. Yingge, I. Ali, and K.-Y. Lee, "Deep neural networks on chip-a survey," in 2020 IEEE International Conference on Big Data and Smart Computing (BigComp), 2020, pp. 589–592.
[20] C. Pelletier, G. I. Webb, and F. Petitjean, "Temporal convolutional neural network for the classification of satellite image time series," Remote Sens (Basel), vol. 11, no. 5, p. 523, 2019.
[21] O. Gurav, "Face Mask Detection Dataset," kaggle.com, 2020. https://www.kaggle.com/datasets/omkargurav/face-mask-dataset (accessed Feb. 11, 2023).