

Mask Detection Using the YOLO (You Only Look Once) Method

Andi Ilham^{1*}, Mutmainnah Muchtar¹, Jayanti Yusmah Sari¹

¹ Universitas Sembilanbelas November Kolaka, Indonesia.

* Corresponding Author. E-mail: 113illang@gmail.com

Keywords

Computer vision;
CNN;
Mask detection;
YOLO;

ABSTRACT

The COVID-19 pandemic has emphasized the importance of wearing masks as a preventive measure. To facilitate mask detection and ensure compliance, computer vision techniques have been widely utilized. This research aims to develop a mask detection system using the YOLO (You Only Look Once) method. YOLO is a real-time object detection method that provides accurate and efficient results. The proposed system utilizes a pre-trained YOLO model trained on a dataset comprising images of individuals with and without masks. The YOLO model can detect and locate faces, as well as differentiate between individuals wearing masks and those who are not. The method works by dividing the image into a grid and predicting bounding boxes and class probabilities for each grid cell. This approach enables real-time mask detection with minimal computational overhead. Experimental evaluations were conducted using various relevant benchmark datasets. The evaluation results demonstrate that the mask detection system using the YOLO method achieves high detection rates and fast response times. This research is expected to contribute to the effort of monitoring mask usage to control the spread of COVID-19.

This is an open access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



PENDAHULUAN

Dalam era globalisasi yang semakin berkembang, tantangan terhadap kesehatan dan keselamatan masyarakat semakin kompleks. Salah satu tantangan utama yang dihadapi saat ini adalah pandemi penyakit COVID-19 yang disebabkan oleh virus SARS-CoV-2. Penyebaran penyakit ini dapat diminimalkan dengan berbagai upaya pencegahan, salah satunya adalah penggunaan masker sebagai perlindungan[1].

Penggunaan masker telah direkomendasikan oleh banyak otoritas kesehatan sebagai langkah efektif dalam mengurangi risiko penularan virus. Namun, masih terdapat individu yang tidak mematuhi aturan penggunaan masker. Oleh karena itu, deteksi penggunaan masker pada wajah manusia menjadi sangat penting, terutama dalam konteks pengawasan di tempat-tempat umum, transportasi, dan lingkungan kerja.

Teknologi komputer dan pengolahan citra memiliki peran yang signifikan dalam mengatasi tantangan ini. Dengan memanfaatkan algoritma deteksi wajah dan pengenalan objek, kita dapat mengembangkan sistem deteksi otomatis yang mampu mengenali apakah seseorang sedang menggunakan masker atau tidak. Salah satu pendekatan yang digunakan dalam penelitian ini adalah algoritma YOLO (*You Only Look Once*) [2], [3], yang dikenal karena kemampuannya dalam mendeteksi objek secara akurat dan cepat.

Penelitian ini bertujuan untuk mengembangkan sistem deteksi masker pada wajah manusia menggunakan algoritma YOLO. Sistem ini diharapkan dapat memberikan kontribusi dalam pengawasan dan penegakan aturan penggunaan masker demi mengurangi risiko penularan penyakit. Melalui kombinasi antara teknologi deteksi objek dan kecerdasan buatan, diharapkan hasil penelitian ini dapat menjadi solusi yang efektif dan efisien dalam menghadapi pandemi dan menjaga kesehatan masyarakat.

METODE

Penelitian ini dilakukan dengan menggunakan citra digital yang diunduh dari berbagai sumber di internet. Kegiatan pengumpulan data dilakukan dengan cara mengunduh dataset yang berupa citra individu yang mengenakan dan tidak mengenakan masker melalui internet. Untuk kebutuhan penelitian ini, citra yang digunakan berjumlah ± 4000 citra agar hasil deteksi dapat lebih akurat dengan kriteria sebagai berikut:

1. Kualitas gambar yang baik

Gambar yang digunakan dalam dataset harus memiliki kualitas yang baik dan jelas sehingga Model deteksi masker dapat bekerja dengan baik. Gambar yang buram atau terlalu kecil akan mengganggu kinerja Model.

2. Variasi pose wajah manusia

Dataset yang baik harus mencakup variasi pose wajah manusia yang berbeda-beda, seperti pose tegak lurus, miring, dan lain-lain. Hal ini akan membantu Model untuk dapat mendeteksi masker pada wajah manusia dari berbagai sudut pandang.

3. Variasi jenis masker

Dataset juga perlu mencakup variasi jenis masker yang berbeda, seperti masker bedah, masker kain, masker N95, dan lain-lain. Hal ini akan membantu Model untuk dapat mengenali dan membedakan jenis masker yang berbeda.

4. Variasi pencahayaan

Dataset perlu mencakup gambar dengan variasi pencahayaan yang berbeda-beda, seperti pencahayaan yang terang dan redup. Hal ini akan membantu Model untuk dapat bekerja dengan baik dalam kondisi pencahayaan yang berbeda.

5. Variasi latar belakang

Dataset perlu mencakup gambar dengan variasi latar belakang yang berbeda-beda, seperti latar belakang yang ramai dan latar belakang yang sepi. Hal ini akan membantu Model untuk dapat bekerja dengan baik dalam berbagai kondisi latar belakang.

6. Keanekaragaman dataset

Dataset yang baik harus memiliki keanekaragaman yang cukup sehingga dapat merepresentasikan variasi kasus yang mungkin terjadi dalam kondisi dunia nyata. Dataset yang monoton akan mengakibatkan model menjadi terbatas dalam kemampuan mendeteksi masker.

Citra Digital

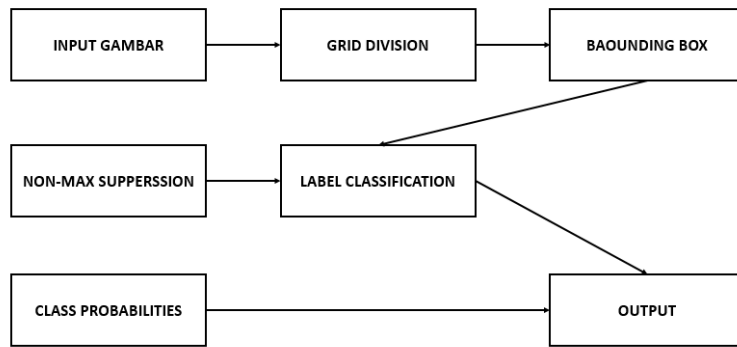
Citra digital adalah bentuk representasi dari suatu objek atau gambar yang dibuat dalam bentuk data digital, yang terdiri dari kumpulan piksel atau elemen citra [4]. Citra digital dapat dihasilkan melalui beberapa cara, seperti pemindaian (*scanning*), pengambilan gambar menggunakan kamera digital, atau pembuatan melalui perangkat lunak komputer [5].

Citra digital sering digunakan dalam berbagai bidang seperti pengolahan gambar, desain grafis, pengenalan pola, pengolahan citra medis, dan sebagainya. Dalam pengolahan gambar, citra digital dapat diproses dan ditingkatkan kualitasnya menggunakan berbagai teknik seperti filtering, enhancement, segmentasi, dan lain-lain.

Pengolahan citra digital juga dapat digunakan dalam beberapa aplikasi seperti deteksi wajah [6], pengenalan plat nomor kendaraan [7][8], deteksi objek dalam video [9], dan sebagainya. Dalam aplikasi deteksi masker dengan algoritma YOLO, citra digital digunakan sebagai input untuk melakukan deteksi objek masker pada setiap Frame video.

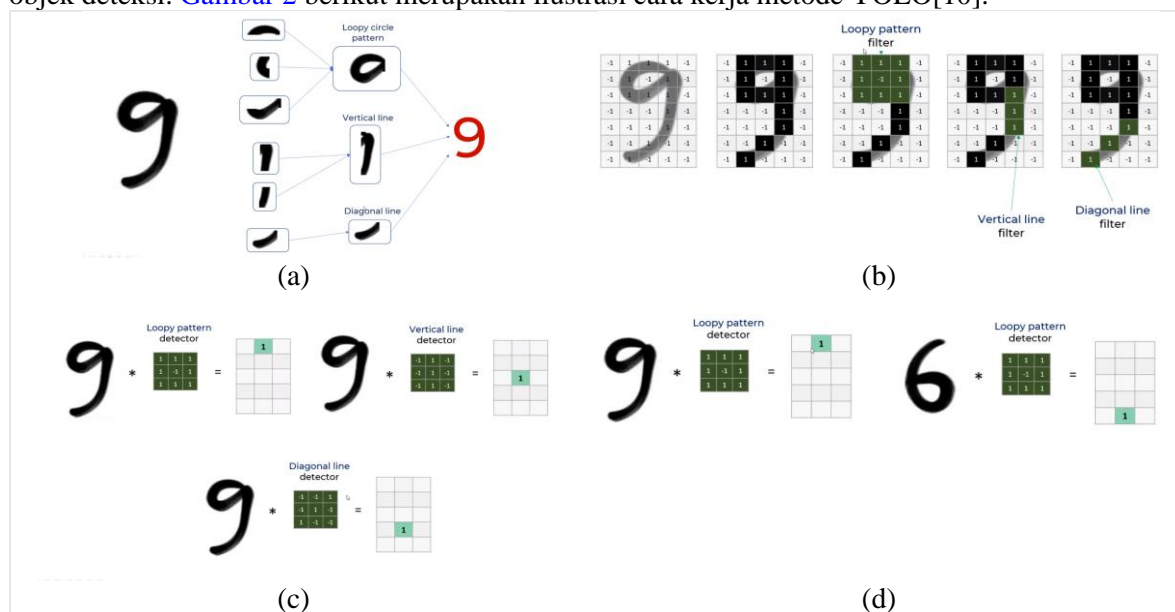
Arsitektur dan Algoritma

Arsitektur yang digunakan dalam penelitian ini melibatkan model MobileNetV2 sebagai *base* model yang telah dilatih sebelumnya. Algoritma utama yang digunakan dalam penelitian ini adalah algoritma YOLO (You Only Look Once) dalam mengenali objek dan lokasi wajah pada citra. Untuk alur deteksi masker dari penelitian ini ditunjukkan pada [Gambar 1](#) berikut:



Gambar 1. Flowchart Deteksi Masker

YOLO memiliki kemampuan untuk mengingat pattern atau pola pada dataset yang telah dilatih. Dari pattern tersebutlah yang menjadi acuan program ini dalam melakukan klasifikasi objek deteksi. Gambar 2 berikut merupakan ilustrasi cara kerja metode YOLO[10].



Gambar 2 (a)Cara Kerja Deteksi YOLO, (b) Pola Loopy, Vertical, dan Diagonal, (c) Matriks Hasil Latih, (d) Deteksi Pola

Deteksi Objek

Deteksi objek menentukan keberadaan sebuah objek, ruang lingkupnya, dan lokasi pada gambar. Objek deteksi mengidentifikasi kelas objek yang terdapat pada database yang telah di-Training. Deteksi objek diawali oleh pengenalan suatu objek [11]. Hal ini dapat diperlakukan sebagai pengenalan objek kelas dua, dimana satu kelas mewakili kelas objek dan kelas lain mewakili kelas non-objek. Deteksi objek dibagi menjadi dua, yaitu *soft detection* dan *hard detection*. *Soft detection* hanya dapat mendeteksi keberadaan suatu objek sedangkan *hard detection* mendeteksi keberadaan objek dan lokasi objek pada gambar [12].

Objek deteksi biasanya dilakukan dengan mencari setiap bagian dari gambar untuk melokalisasi bagian, yang bersifat fotometrik atau geometrinya cocok dengan objek target dalam Training basis data. Hal ini dapat dicapai dengan memindai template objek pada seluruh gambar di lokasi yang berbeda, skala, dan rotasi, dan deteksi dinyatakan jika kesamaan antara template dan gambar cukup tinggi. Kesamaan antara template dan wilayah gambar dapat diukur dengan korelasinya. Beberapa tahun terakhir telah terbukti bahwa objek deteksi berbasis gambar sensitif terhadap data *training*.

Proses dalam deteksi objek pada video sama dengan proses deteksi objek pada gambar. Video terdiri dari beberapa gambar atau *Frame*. Dalam satu video, dapat dipecah menjadi beberapa *Frame* lagi, dimana pada setiap *Frame* ini melakukan proses deteksi objek. Kemudian, *Frame* yang telah diproses disatukan kembali menjadi video utuh. Pemrosesan deteksi objek pada video dapat dilakukan dengan menggunakan *Graphical Processing Unit* (GPU).

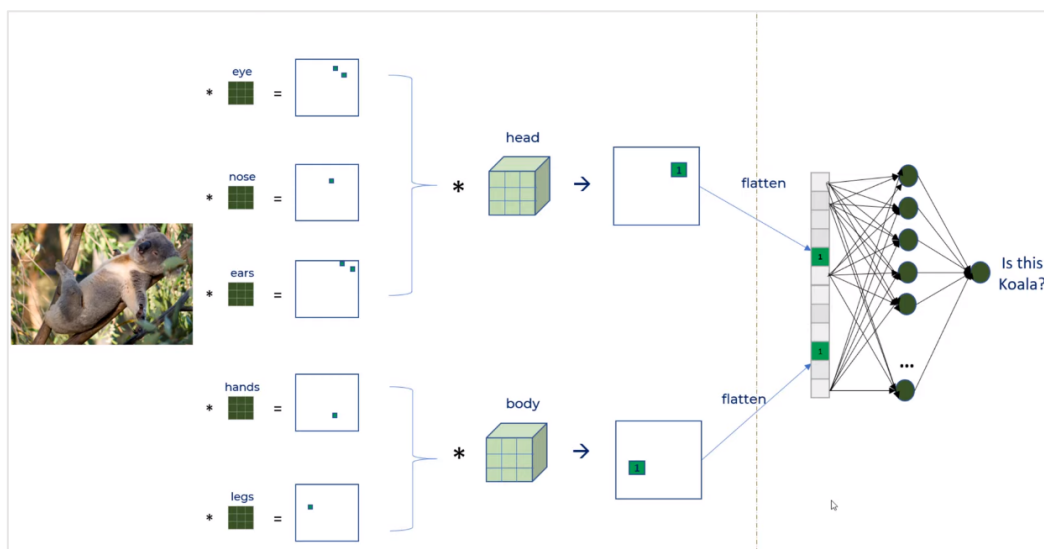
YOLO (*You Only Look Once*)

YOLO (*You Only Look Once*) adalah sebuah algoritma deteksi objek *real-time* yang dikembangkan oleh Joseph Redmon, Santosh DivVala, Ross Girshick, dan Ali Farhadi pada tahun 2016 [13]. Algoritma ini menggunakan arsitektur konvolusi jaringan syaraf (*convolutional neural network/CNN*) untuk mendeteksi objek pada gambar atau video dengan cepat dan akurat.

YOLO seperti yang ditunjukkan pada [Gambar 3](#) bekerja dengan cara membagi gambar atau *Frame* video menjadi grid kecil, dan kemudian memprediksi kotak pembatas (*Bounding Box*) dan kelas objek pada setiap *grid*. Setiap kotak pembatas yang diprediksi memiliki skor yang menunjukkan seberapa yakin YOLO bahwa objek tersebut benar-benar ada. Algoritma ini kemudian menggabungkan kotak pembatas dari semua *grid* dan menyaring kotak yang memiliki skor rendah [14].

YOLO memiliki kelebihan dalam kecepatan dan efisiensi, karena hanya melakukan prediksi objek sekali pada seluruh gambar atau video. Algoritma ini juga dapat mendeteksi objek dengan ukuran yang berbeda dalam satu gambar atau video, dan dapat memprediksi banyak objek dalam waktu yang singkat. Oleh karena itu, YOLO banyak digunakan dalam berbagai aplikasi deteksi objek *real-time* seperti deteksi wajah, deteksi kendaraan, dan deteksi masker.

YOLO hanya menggunakan lapisan konvolusi dan lapisan *pooling* [15]. Untuk lapisan konvolusi terakhir disesuaikan dengan jumlah kelas dan jumlah kotak prediksi yang diinginkan.



Gambar 3 Cara Kerja YOLO

Tensorflow [16]

Tensorflow adalah sebuah *platform open-source* untuk pengembangan dan implementasi Model machine learning, termasuk deep learning. TensorFlow dikembangkan oleh Google Brain Team dan dirilis pertama kali pada tahun 2015. *Platform* ini memungkinkan pengembang dan peneliti untuk membuat model *machine learning* dengan cepat dan mudah, serta mengimplementasikan model tersebut ke dalam aplikasi yang berbeda [17]. *TensorFlow* berbasis pada konsep tensor, yang merupakan sebuah array multi-dimensional. Dalam *TensorFlow*, data yang dimasukkan ke dalam Model diolah sebagai tensor, dan Model diimplementasikan dalam bentuk grafik aliran (*computational graph*) yang terdiri dari serangkaian operasi tensor. TensorFlow juga menyediakan API yang *user-friendly* dan mudah digunakan,

sehingga memudahkan pengguna dalam membuat, mengoptimasi, dan mengimplementasikan Model *machine learning*. Selain itu, *TensorFlow* juga dapat dijalankan pada berbagai platform seperti desktop, mobile, dan web. *TensorFlow* sangat populer di kalangan peneliti dan pengembang machine learning karena kemampuannya yang sangat fleksibel, baik untuk Model kecil maupun besar, serta performanya yang sangat baik pada tugas-tugas machine learning kompleks seperti pengenalan citra dan pemrosesan bahasa alami. *TensorFlow* juga dilengkapi dengan berbagai algoritma machine learning yang telah terbukti, serta dukungan yang kuat dari komunitas pengembang dan peneliti.

Metode Evaluasi

Metode evaluasi yang digunakan dalam penelitian ini adalah dengan menghitung akurasi, presisi, *recall*, dan *F1-score* dari hasil deteksi masker. Evaluasi dilakukan dengan membandingkan hasil deteksi dengan label yang telah diberikan [18].

Indikator Evaluasi Performa Latih Model

Untuk pengujian performa saat proses pelatihan dan evaluasi, digunakan indikator yaitu, *train_acc*, *val_acc*, *train_loss*, dan *val_loss*.

Train_acc: Mengukur akurasi (*accuracy*) model pada data pelatihan. Indikator ini menunjukkan sejauh mana model mampu mempelajari dan menggeneralisasi pola pada data pelatihan.

$$\text{train_acc} = \frac{(\text{jumlah prediksi benar pada data pelatihan})}{(\text{jumlah total data pelatihan})} \quad (1)$$

Val_acc: Mengukur akurasi model pada data evaluasi (validasi). Indikator ini memberikan perkiraan kinerja model pada data yang tidak pernah dilihat sebelumnya, dan dapat digunakan untuk mengevaluasi kemampuan model dalam menggeneralisasi pada data baru.

$$\text{val_acc} = \frac{(\text{jumlah prediksi yang benar pada data evaluasi})}{(\text{jumlah total data evaluasi})} \quad (2)$$

Train_loss: Mengukur nilai fungsi loss (kerugian) model pada data pelatihan. Indikator ini mencerminkan sejauh mana model mampu menyesuaikan bobotnya untuk mengurangi kesalahan dalam prediksi pada data pelatihan.

$$\text{train_loss} = \frac{(\text{jumlah prediksi salah pada data latihan})}{(\text{jumlah total data pelatihan})} \quad (3)$$

Val_loss: Mengukur nilai fungsi *loss* model pada data evaluasi (validasi). Indikator ini memberikan perkiraan tingkat kesalahan model pada data yang tidak pernah dilihat sebelumnya. Tujuan utamanya adalah untuk meminimalkan *val_loss* dan meningkatkan *val_acc*, yang menunjukkan model yang lebih baik dalam melakukan prediksi yang akurat dan umum pada data baru.

$$\text{val_loss} = \frac{(\text{jumlah prediksi salah pada data evaluasi})}{(\text{jumlah total data evaluasi})} \quad (4)$$

Indikator Evaluasi Performa Deteksi Model

Precision, *recall*, dan *F1-score* adalah metrik evaluasi yang umum digunakan dalam pemodelan klasifikasi. *Precision* (presisi) mengukur sejauh mana prediksi positif yang dilakukan oleh model benar. Ini menghitung persentase positif yang benar dari semua prediksi positif yang dilakukan oleh model. Rumus *precision* adalah:

$$\text{Precision} = \frac{\text{True Positives}}{(\text{True Positives} + \text{False Positives})} \quad (5)$$

Recall (recall) mengukur sejauh mana model mampu mendeteksi secara akurat semua contoh positif yang ada dalam data aktual. Ini menghitung persentase positif yang benar yang berhasil dideteksi oleh model. Rumus *recall* adalah:

$$\text{Recall} = \frac{\text{True Positives}}{(\text{True Positives} + \text{False Negatives})} \quad (6)$$

F1-score adalah ukuran gabungan dari *precision* dan *recall* yang memberikan nilai tunggal untuk mengevaluasi kinerja model secara keseluruhan. *F1-score* adalah rata-rata harmonis dari *precision* dan *recall*. *F1-score* memberikan keseimbangan antara *precision* dan *recall* dan berguna ketika ada ketidakseimbangan antara kelas positif dan negatif dalam data. Rumus *F1-score* adalah:

$$F1_Score = \frac{2 \times (\text{Precision} \times \text{Recall})}{(\text{Precision} + \text{Recall})} \quad (7)$$

Precision, *recall*, dan *F1-score* memiliki rentang nilai antara 0 dan 1, di mana nilai 1 menunjukkan kinerja yang sempurna dan nilai 0 menunjukkan kinerja yang buruk. Tujuan utama dari metrik ini adalah untuk mengukur kemampuan model dalam melakukan klasifikasi yang akurat dan meminimalkan kesalahan prediksi *false positive* dan *false negative*.

HASIL DAN PEMBAHASAN

Hasil pengujian data latih mencerminkan informasi lebih rinci tentang kinerja model dalam mengklasifikasikan data latih pada setiap kelas atau label yang ada. Dalam pengujian data latih, model menggunakan data yang sama yang digunakan untuk melatihnya. Dengan mengamati *precision*, *recall*, dan *F1-score* pada hasil pengujian data latih, kita dapat mendapatkan informasi tentang seberapa baik model mampu mengklasifikasikan data dengan benar, mendeteksi positif secara akurat, dan mencapai keseimbangan antara *precision* dan *recall*.

Hasil evaluasi pada [Gambar 4](#) merupakan klasifikasi menggunakan metrik *precision*, *recall*, dan *F1-score* menunjukkan kinerja yang sangat baik untuk kedua kelas yang diamati, yaitu "masker" dan "tanpa_masker". Untuk kelas "masker", *precision* dan *recall* mencapai 99%, menunjukkan bahwa sebagian besar prediksi positif benar dan sebagian besar sampel positif berhasil terdeteksi. Hal yang sama juga berlaku untuk kelas "tanpa_masker". *F1-score*, yang merupakan perpaduan antara *precision* dan *recall*, juga mencapai 0.99 untuk kedua kelas. Jumlah sampel yang digunakan dalam evaluasi adalah 383 untuk kelas "masker" dan 384 untuk kelas "tanpa_masker". Dengan tingkat akurasi sebesar 0.99, hasil keseluruhan dari klasifikasi ini sangat baik.

```

Epoch 17/20
95/95 [=====] - 27s 280ms/step - loss: 0.0287 - accuracy:
0.9934 - val_loss: 0.0269 - val_accuracy: 0.9935
Epoch 18/20
95/95 [=====] - 25s 266ms/step - loss: 0.0292 - accuracy:
0.9918 - val_loss: 0.0319 - val_accuracy: 0.9922
Epoch 19/20
95/95 [=====] - 25s 266ms/step - loss: 0.0290 - accuracy:
0.9901 - val_loss: 0.0243 - val_accuracy: 0.9922
Epoch 20/20
95/95 [=====] - 26s 269ms/step - loss: 0.0267 - accuracy:
0.9911 - val_loss: 0.0233 - val_accuracy: 0.9935
[INFO] evaluating network...
2023-05-24 13:12:05.587898: W tensorflow/core/framework/cpu_allocator_impl.cc:82] A
llocation of 461819904 exceeds 10% of free system memory.
24/24 [=====] - 2s 40ms/step
      precision    recall  f1-score   support
masker      0.99      0.99      0.99      383
tanpa_masker 0.99      0.99      0.99      384
accuracy                    0.99      767
macro avg      0.99      0.99      0.99      767
weighted avg   0.99      0.99      0.99      767

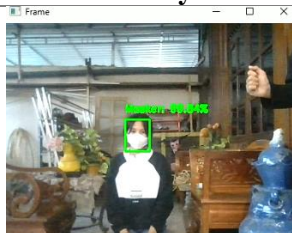
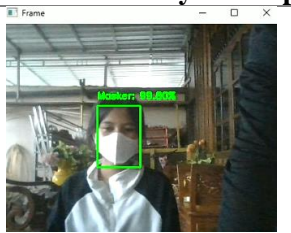
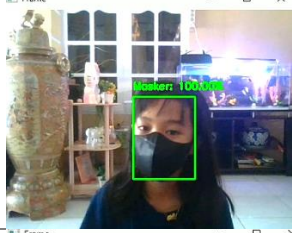

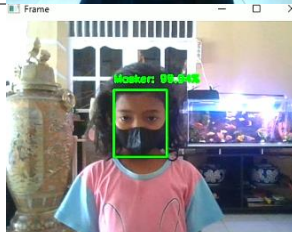
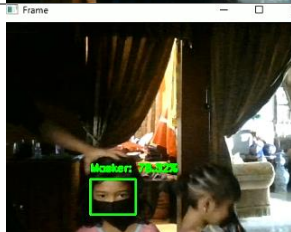
[INFO] saving mask detector model...
C:\TUGAS KULIAH\PATTERN RECOGNITION\deteksi-masker>

```

Gambar 4. Hasil Evaluasi

Untuk menguji seberapa jauh kemampuan program ini dapat melakukan deteksi, dilakukan percobaan dengan kondisi cahaya dan jarak yang berbeda-beda dan didapatkan hasil seperti pada [Tabel 1](#) berikut.



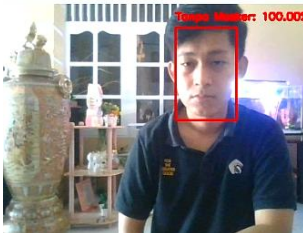

Tabel 1. Hasil Pengujian Pemakaian Masker dengan Kondisi Cahaya Berbeda

No	Kondisi Cahaya Normal (Masker)	Kondisi Cahaya Redup (Masker)
1		
2		
3		

Berdasarkan hasil pengujian pada Tabel 1, program ini dapat mendeteksi objek yang sedang mengenakan masker dengan baik pada kondisi cahaya yang normal. Namun, pada kondisi cahaya redup, akurasi deteksi program cenderung tidak konsisten dikarenakan dari beberapa percobaan, hasil yang di tampilkan berbeda-beda. Terkadang menunjukkan nilai confidence yang tinggi, terkadang rendah, bahkan sampai tidak dapat mendeteksi sama sekali. Hal ini dikarenakan cahaya berperan penting agar objek dapat terlihat lebih jelas. Sehingga program dapat membedakan saat objek sedang mengenakan masker ataupun tidak.

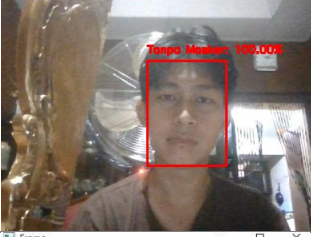
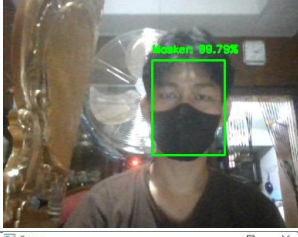
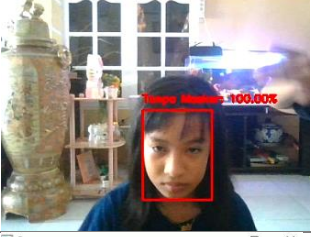


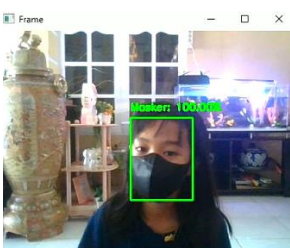
Tabel 2. Hasil Pengujian Tanpa Masker dengan Kondisi Cahaya Berbeda

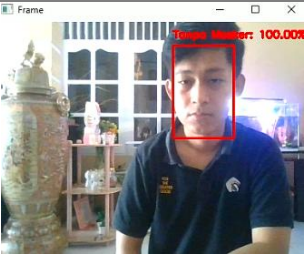
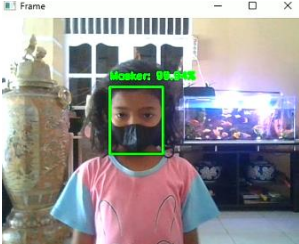
No	Kondisi Cahaya Normal (Tanpa Masker)	Kondisi Cahaya Redup (Tanpa Masker)
1		
2		

No	Kondisi Cahaya Normal (Tanpa Masker)	Kondisi Cahaya Redup (Tanpa Masker)
3		
4		

Berbeda dari pengujian sebelumnya, program dapat mendeteksi objek tanpa mengenakan masker dengan baik di kondisi cahaya normal ataupun redup. Hal ini dikarenakan jumlah dataset untuk objek tanpa masker lebih banyak. Tak hanya dari gambar saja, namun juga dari file “res10_300x300_ssd_iter_140000.caffemodel”. Angka “140000” pada file menunjukkan jumlah iterasi (Epoch) yang telah dilakukan pada model tersebut sebelumnya.

Tabel 3. Hasil Pengujian dengan Jarak 1 M

No	Jarak 1m (Tanpa Masker)	Jarak 1m (Dengan Masker)
1		
2		
3		

No	Jarak 1m (Tanpa Masker)	Jarak 1m (Dengan Masker)
4		

Pada pengujian dengan jarak 1 meter yang ditunjukkan pada Tabel 3, program dapat mendeteksi objek yang tidak mengenakan masker dan mengenakan masker dengan akurat dengan akurasi rata-rata di atas 90%.

Tabel 4. Hasil Pengujian dengan Jarak 2 M

No	Jarak 2m (Masker)	Jarak 2m (Tanpa Masker)
1		
2		
3		
4		

Pada pengujian dengan jarak 2 meter (Tabel 4), program dapat mendeteksi objek dewasa yang tidak mengenakan masker dan mengenakan masker dengan akurat. Untuk objek anak-anak, akurasi deteksi saat mengenakan masker lumayan rendah. Dan untuk objek anak balita mengenakan masker, program tidak dapat mendeteksinya. Hal tersebut dikarenakan dataset yang lebih dominan dewasa dibanding anak-anak ataupun balita. Untuk objek tidak mengenakan masker, program masih mampu mendeteksinya dengan sangat baik.

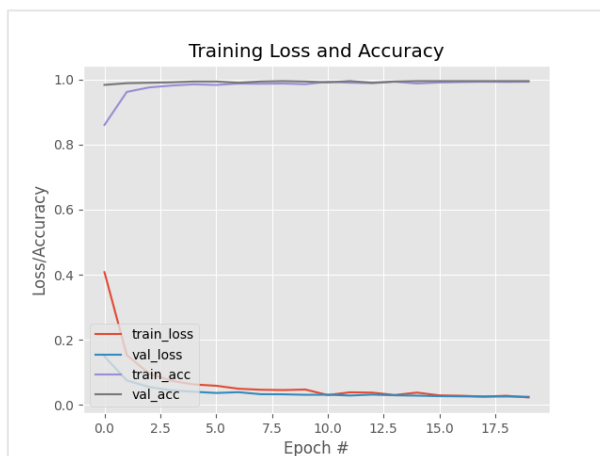
Tabel 5. Hasil Pengujian dengan Jarak 3 m

No	Jarak 3m (Masker)	Jarak 3m (Tanpa Masker)
1		
2		
3		
4		

Pada pengujian dengan jarak 3 meter lebih (Tabel 5), program ini masih mampu mendeteksi objek yang tidak mengenakan masker dengan akurat, baik itu objek dewasa ataupun anak-anak. Namun, program ini tidak dapat mendeteksi objek anak-anak yang sedang mengenakan masker pada jarak tersebut. Hal ini lagi-lagi dikarenakan class masker hanya mengandalkan dataset yang berupa gambar saja. Berbeda dengan class tanpa masker yang memiliki dataset berupa gambar dan file “res10_300x300_ssd_iter_140000.caffemodel untuk meningkatkan kemampuan deteksi objek kepala manusia.

Hasil Pengujian Akurasi Data Latih

Untuk Akurasi Latihnya dapat dilihat pada Gambar 5 dan plot grafik hasil latih datasetnya dari *epoch* 1 hingga *epoch* 20. Setiap *epoch*-nya mengalami peningkatan di akurasi dan mengalami penurunan pada *loss* yang sangat drastis. Hal ini menunjukkan bahwa model yang dilatih memiliki tingkat akurasi yang sangat tinggi. Dari plot di bawah diperoleh *Train_acc* dan *Val_acc* dengan score yang cukup baik yaitu 99% juga *Train_loss* dan *Val_loss* dengan *score* hampir mendekati 0%.



Gambar 5. Hasil Pengujian (*Training Loss dan Accuracy*)

Gambar 5 merupakan hasil training berdasarkan dataset berupa gambar yang diambil pada kondisi cahaya yang normal dengan jarak yang tidak terlalu jauh dari kamera dengan jumlah objek dewasa lebih banyak dibandingkan objek anak-anak.

SIMPULAN

Dari hasil penelitian yang sudah dilakukan, dapat ditarik kesimpulan sebagai berikut:

1. Hasil dari pendeteksian masker menggunakan algoritma YOLO dapat dinilai bekerja dengan baik. Hasil dari deteksi menggunakan masukan video real time terkadang menghasilkan nilai confidence yang berbeda-beda pada setiap Frame-nya, hal ini disebabkan karena objek yang terkadang berpindah posisi.
2. Hasil Training data pada kedua kelas menunjukkan tingkat Val_Acc jauh lebih besar dibandingkan jumlah Val_Loss hal ini menunjukkan sistem telah dapat mendeteksi objek dengan baik dengan tingkat akurasi mendekati 100% berdasarkan pada gambar 4.10 dengan kondisi cahaya yang baik dan jarak tidak terlalu jauh dari kamera.
3. Jika deteksi dilakukan pada jarak yang cukup jauh dengan kondisi cahaya redup dan objeknya berupa anak-anak mengenakan masker, maka deteksi program ini tidak dapat berjalan dengan baik dikarenakan dataset lebih dominan pada objek dewasa dibandingkan anak kecil dan juga untuk class masker memiliki satu jenis dataset dibandingkan class tanpa masker yang didukung dengan dua jenis dataset.
4. Kualitas video dapat mempengaruhi pendeteksian karena semakin tinggi kualitas video maka hasil klasifikasi dan Bounding Box semakin tinggi pula. Maka dari itu, semakin tinggi spesifikasi kamera yang digunakan, maka semakin baik pula akurasi deteksi yang dapat ditampilkan.

DAFTAR PUSTAKA

- [1] Y. Theopilus, T. Yogasara, C. Theresia, and J. R. Octavia, "Analisis risiko produk alat pelindung diri (apd) pencegah penularan covid-19 untuk pekerja informal di indonesia," *J. Rekayasa Sist. Ind.*, vol. 9, no. 2, pp. 115–134, 2020.
- [2] M. J. Shafiee, B. Chywl, F. Li, and A. Wong, "Fast YOLO: A fast you only look once system for real-time embedded object detection in video," *arXiv Prepr. arXiv1709.05943*, 2017.
- [3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.
- [4] S. Sumijan and P. A. W. Purnama, "Teori dan Aplikasi Pengolahan Citra Digital Penerapan

- dalam Bidang Citra Medis.” PENERBIT INSAN CENDEKIA MANDIRI, 2021.
- [5] F. Marpaung, F. Aulia, and R. C. Nabila, “COMPUTER VISION DAN PENGOLAHAN CITRA DIGITAL.” PUSTAKA AKSARA, 2022.
 - [6] F. Fandiansyah, J. Y. Sari, and I. P. Ningrum, “Pengenalan Wajah Menggunakan Metode Linear Discriminant Analysis dan k Nearest Neighbor,” *Ultimatics*, vol. 9, no. 1, pp. 1–9, Jun. 2017.
 - [7] A. Setiyorini, I. P. N. Purnama, J. Y. Sari, M. Muchtar, and E. Ngii, “Vehicle number plate identification using template matching algorithm for automatic parking system,” in *ACM International Conference Proceeding Series*, 2019. doi: 10.1145/3330482.3330483.
 - [8] Y. P. Pasrun, M. Muchtar, and A. N. Basyarah, “Indonesian License Plate Detection Using Morphological Operation,” in *IOP Conference Series: Materials Science and Engineering*, 2020, vol. 797, no. 1, p. 12037.
 - [9] I. Lugianti, J. Y. Sari, and I. P. Ningrum, “Deteksi Kecepatan Kendaraan Bergerak Berbasis Video Menggunakan Metode Frame Difference,” in *Seminar Nasional*, 2012, vol. 1, pp. 324–332.
 - [10] D. N. Alfarizi, R. A. Pangestu, D. Aditya, M. A. Setiawan, and P. Rosyani, “Penggunaan Metode YOLO Pada Deteksi Objek: Sebuah Tinjauan Literatur Sistematis,” *AI dan SPK J. Artif. Intell. dan Sist. Penunjang Keputusan*, vol. 1, no. 1, pp. 54–63, 2023.
 - [11] A. N. Sugandi and B. Hartono, “Implementasi Pengolahan Citra pada Quadcopter untuk Deteksi Manusia Menggunakan Algoritma YOLO,” in *Prosiding Industrial Research Workshop and National Seminar*, 2022, vol. 13, no. 01, pp. 183–188.
 - [12] S. S. Sindarto, D. E. Ratnawati, and I. Arwani, “Klasifikasi Citra Sistem Isyarat Bahasa Indonesia (SIBI) dengan Metode Convolutional Neural Network pada Perangkat Lunak berbasis Android,” *J. Pengemb. Teknol. Inf. Dan Ilmu Komput.*, vol. 6, no. 5, pp. 2129–2138, 2022.
 - [13] N. E. Budiayanta, M. Mulyadi, and H. Tanudjaja, “Sistem Deteksi Kemurnian Beras berbasis Computer Vision dengan Pendekatan Algoritma YOLO,” *J. Inform. J. Pengemb. IT*, vol. 6, no. 1, pp. 51–55, 2021.
 - [14] C. Geraldly and C. Lubis, “Pendeteksian dan Pengenalan Jenis Mobil Menggunakan Algoritma You Only Look Once dan Convolutional Neural Network,” *J. Ilmu Komput. dan Sist. Inf.*, vol. 8, no. 2, pp. 197–199, 2020.
 - [15] P. Y. Putra, A. S. Arifianto, Z. E. Fitri, and T. D. Puspitasari, “Deteksi Kendaraan Truk pada Video Menggunakan Metode Tiny-YOLO v4,” *J. Inform. Polinema*, vol. 9, no. 2, pp. 215–222, 2023.
 - [16] E. Bisong, *Building machine learning and deep learning models on Google cloud platform*. Springer, 2019.
 - [17] F. A. Nugraha, N. H. Harani, and R. Habibi, *Analisis Sentimen Terhadap Pembatasan Sosial Menggunakan Deep Learning*. Kreatif, 2020.
 - [18] S. W. Dari and J. Triloka, “Kajian Algoritme Mask Region-Based Convolutional Neural Network (Mask R-CNN) dan You Look Only Once (YOLO) Untuk Deteksi Penyakit Kulit Akibat Infeksi Jamur,” in *Prosiding Seminar Nasional Darmajaya*, 2022, vol. 1, pp. 132–138.