

SISTEM WARNING BERBASIS DETEKSI MASKER PADA WAJAH SECARA REALTIME MENGGUNAKAN METODE HAAR-CASCADE

Andi Patombongi^{*1}, Muhammad Ardi Putra², Dea Angelia Kamil³, Farida Asriani⁴, Imelda Dua Raja⁵

¹Sistem Komputer, STMIK Catur Sakti Kendari

^{1,2,3,4,5}Doktor Ilmu Komputer, Universitas Gadjah Mada

e-mail: andipatombongi1084@mail.ugm.ac.id

Pada paper ini penulis mengembangkan sebuah sistem berbasis computer vision yang mampu mendeteksi wajah dengan dan tanpa masker. Tidak hanya itu, penggunaan masker yang tidak sempurna juga dapat dideteksi oleh sistem. Model yang diterapkan pada sistem ini adalah Adaboost yang bekerja menggunakan Haar-like features. Hasilnya, sistem mampu bekerja dengan akurasi 90% terhadap data uji dengan kecepatan pemrosesan sebesar 7,9 FPS.

Kata Kunci: Object detection, Haar cascade, OpenCV, Deteksi Masker, Sistem Warning.

I. PENDAHULUAN

Coronavirus Disease 2019 (COVID-19) merupakan penyakit menular yang disebabkan oleh jenis coronavirus yang baru ditemukan pada akhir 2019. perkembangan dan penularan virus ini sangat cepat dan mewabah di banyak negara di dunia, dan pada 2020 WHO menetapkan sebagai pandemi global. COVID-19 dapat menyebar terutama dari orang ke orang melalui percikan-percikan dari hidung atau mulut yang keluar saat orang yang terinfeksi COVID-19 batuk, bersin atau berbicara. Berbagai upaya pencegahan penularan covid 19 telah dilakukan diantaranya adalah kewajiban penggunaan masker dan penerapan protocol Kesehatan yang telah ditentukan. Praktek penerapan protocol Kesehatan guna pencegahan penyebaran covid 19 memerlukan pengawasan yang ketat. Diarea-area publik pengawasan penerapan protokol Kesehatan telah dilakukan dengan mewajibkan pengunjung untuk mencuci tangan sebelum masuk, pengecekan suhu tubuh dan penggunaan masker. Namun demikian pengecekan penggunaan masker masih dilakukan secara manual. Oleh karenanya dipandang perlunya dikembangkan sistem pengecekan penggunaan masker secara otomatis dan real time guna menekan human error dalam pengawasan.

Deteksi masker wajah adalah masalah penting dalam keamanan dan pencegahan Covid-19. Beberapa algoritma telah diterapkan diantaranya deteksi penggunaan masker dengan you only look one versi 3 (YOLOv3)[1], YOLO v3 Tiny sangat baik untuk deteksi objek dan memiliki

keunggulan dalam hal presisi dan ringan pada beberapa aspek,. Untuk endapatkan hasil yang maksimal, diperlukan data yang sangat banyak sebelum proses pelatihan sehingga akurasi bagus. Namun, kedalaman lapisan konvolusional dikurangi sehingga YOLOv3-tiny tidak dapat melakukan multi-skala. Proses pelatihan memakan waktu lama, sehingga membutuhkan computer dengan spesifikasi yang tinggi.

Arsitektur yolov4 telah diterapkan untuk deteksi penggunaan masker secara real time. Setelah dilatih sebanyak 4000 epoch, algoritma yolov4 memberikan hasil yang baik mean average precission (mAP) sebesar 98 [2]. Deteksi masker secara real time dengan menggunakan algoritma yolov5 telah dikembangkan. dengan akurasi mencapai 95,9% dan mAP mencapai 84,8% [3].

Deteksi msker dengan metode convolutional neural networks (CNN) [4][5]. Dan juga deteksi penggunaan masker secara real time dengan CNN juga te;ah dikembangkan [6][7]. CNN memiliki keunggulan dalam hal akurasi, dan loss nya bisa mendekati 0, tetapi dalam trainingnya membutuhkan data yang sangat banyak sehingga memerlukan waktu yang lama. Penggunaan arsitektur CNN dan MobileNetV2 sebagai pengklasifikasi, bobotnya ringan, menggunakan lebih sedikit parameter dan dapat digunakan pada sistem tertanam (raspberry pi, Onion Omega2) untuk melakukan deteksi masker secara real time [7].

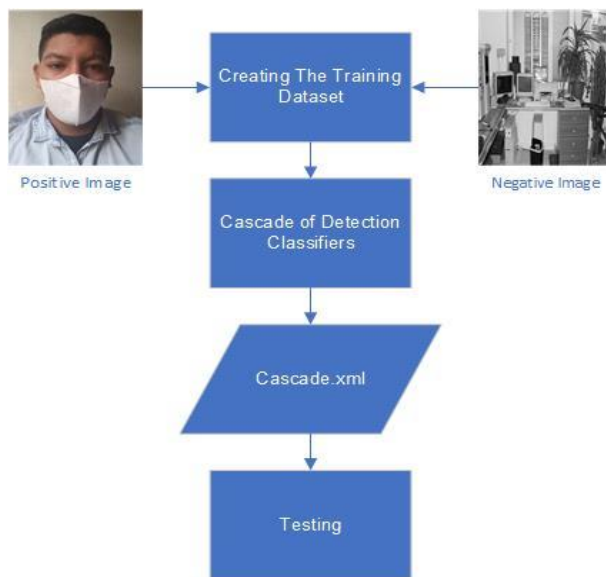
Jansi Rani Sella Veluswami pada 2021 menerapkan model deteksi masker dengan Teknik deep learning. model dilatih dengan dataset lebih dari 11.000 gambar wajah baik menggunakan atau tanpa masker. Model SSDNET diterapkan untuk deteksi wajah, yang keluarannya diteruskan ke CNN Ringan yang dibuat khusus untuk deteksi masker dengan akurasi 96% [8].

CNN dengan arsitektur MobileNetV2, dan library OpenCV, tensorflow, keras, dan Pytorch diterapkan untuk mendeteksi apakah orang memakai masker wajah atau tidak [9]. (Nagrath et al., 2021) Single Shot Multibox Detector sebagai pendeteksi wajah dan MobileNetV2 (SSDMNV2) telah digunakan untuk melakukan deteksi masker wajah secara real-time. Penelitian ini menghasilkan akurasi 92.64% [10]. Selanjutnya juga telah dikembangkan arsitektur InceptionV3 dan transfer

learning untuk mendeteksi masker wajah pada Simulated Masked Face Dataset (SFMD). Akurasi yang dihasilkan pada penelitian ini adalah 100%. Namun penelitian ini diterapkan pada dataset simulasi wajah dengan menambahkan masker secara tidak nyata [11].

II. METODE PENELITIAN

Tahapan awal yang dilakukan pada penelitian sistem deteksi masker pada wajah dengan metode Haar Cascade adalah melakukan studi literatur. Studi literatur dilakukan untuk menganalisa dan mencari sistem yang tepat untuk diaplikasikan. Tahapan-tahapan deteksi masker pada wajah dapat dilihat pada Gambar 1..



Gambar 1. Flowchart metode Haar Cascade Classifier.

Bagian pertama dalam pelatihan diperlukan adalah mengumpulkan gambar, kemudian membuat sampel berdasarkan gambar tersebut untuk memulai proses pelatihan. Selanjutnya penggunaan OpenCV yang akan membaca gambar dan file fitur. Jadi pada titik ini, ada array Numpy di titik data utama. Dengan File XML tersebut dapat dimuat dengan metode cascadeClassifier untuk mengekstraksi fitur wajah.

Algoritma Haar digunakan untuk melakukan pendeteksian wajah. Deteksi masker pada wajah menggunakan Bahasa pemrograman Python dengan Pseudocode sebagai berikut:

Pseudocode Mask Detection

```

1  Inisialisasi cascade classifier untuk wajah, hidung,
   mulut, dan masker.
2  while camera is on:
3      konversi ke grayscale
4      blurring menggunakan kernel Gaussian
5      lakukan deteksi wajah
6      lakukan deteksi masker
7      if ada wajah terdeteksi:
8          for each wajah yang terdeteksi:
9              lakukan deteksi hidung dan mulut
10             if hidung OR mulut terdeteksi:

```

```

11         print "Hidung terlihat, tidak
            menggunakan masker"
12     if ada masker terdeteksi:
13         lakukan deteksi hidung
14         if ada hidung terdeteksi:
15             print "Hidung terlihat, tidak
                menggunakan masker"
16     else:
17         print "Menggunakan masker"

```

Metode haar cascade digunakan untuk mendeteksi masker pada wajah sehingga menghasilkan 2 kondisi Menggunakan masker atau tidak menggunakan masker. Jika ada wajah terdeteksi lakukan deteksi hidung dan mulut, jika hidung atau mulut yang terdeteksi maka muncul pesan hidung terlihat, tidak menggunakan masker. Jika kondisi ada masker terdeteksi lakukan deteksi hidung, jika hidung terdeteksi maka muncul pesan Hidung terlihat, tidak menggunakan masker. Selain itu muncul pesan Menggunakan masker.

A. Proses Membuat Cascade Classifier

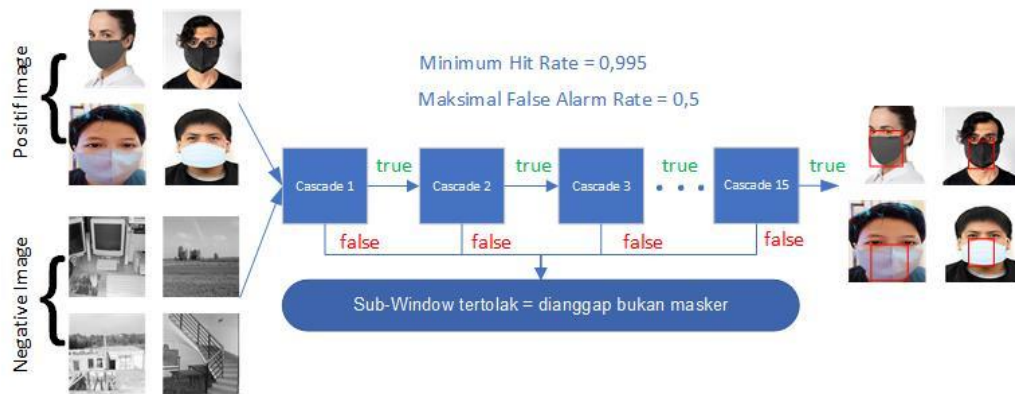
Algoritma pembelajaran AdaBoost digunakan sebagai tahap awal untuk deteksi object menggunakan data citra. Haar-like feature mempunyai sifat learner dan classifier yang lemah. Jika ingin mendapatkan hasil yang lebih akurat maka harus dilakukan proses haar-like feature secara massal, semakin banyak proses haar-like feature yang dilakukan maka akan semakin akurat hasil yang dicapai. Oleh karena itu diadopsi metode cascade classifier untuk pemrosesan fitur Haar. Untuk alur kerja yang lebih lengkap akan dijelaskan pada bagian Training.

Pada tahapan kerja Cascade Classifier, dengan menggunakan teknik sliding window, tiap window atau sub citra dilakukan ekstraksi fitur menggunakan Haar-like feature. Proses ini dilakukan berulang pada tiap stage atau setiap cascade. Semakin cascade yang digunakan maka semakin banyak fitur pada tiap cascadenya. Proses filter dilakukan hingga jumlah sub window yang lolos klasifikasi pun akan berkurang hingga mendekati citra yang ada pada sampel.

B. Training

Gambar 2 Merupakan proses training yang dilakukan untuk mendapatkan clasifier yang dapat mendeteksi masker pada sebuah citra, secara detail masing masing proses adalah sebagai berikut:

Membuat dataset pelatihan: Proses dimulai dengan mengumpulkan data citra positif dan negatif. Dataset citra positif terdiri dari gambar yang berisi objek wajah yang menggunakan masker. Gambar negatif adalah gambar yang tidak mengandung objek wajah, misalnya. gambar latar belakang, gambar non-wajah, dan lain lain sebelum dilakukan proses training dataset positif terlebih dahulu dilakukan pelabelan untuk menentukan dimana posisi masker pada masing masing citra positif. Setelah melakukan pelabelan, maka dibuat sebuah vektor yang akan menggabung semua citra pada dataset positif yang telah dilakukan pelabelan sebelumnya.



Gambar 2. Tahapan training menggunakan Haar Cascade Classifier.

Tabel 1. Parameter Training

No	Haar Cascade Parameter	
	Parameter	Nilai
1	Jumlah sampel positif yang digunakan dalam pelatihan untuk setiap tahapan classifier.	400
2	Jumlah sampel negatif yang digunakan dalam pelatihan untuk setiap tahapan classifier.	1000
3	Jumlah tahapan kaskade yang akan dilatih.	15
4	Ukuran buffer untuk nilai fitur yang dihitung sebelumnya (dalam Mb)	2048
5	Lebar basis sub window (dalam piksel).	24
6	Tinggi basis sub window (dalam piksel).	24
7	Hit rate minimal yang diinginkan untuk setiap tahap classifier	0.995
8	Maksimal false alarm rate	0.5

Cascade of Detection Classifiers: merupakan proses training, proses training dilakukan dengan beberapa parameter yang dapat dilihat pada Tabel 1. Basic classifier disatukan untuk membentuk stage classifier yang akan dikelompokkan. Serangkain classifier tersebut diterapkan ke setiap sub window citra. Hasil positif dari classifier pertama akan menstimulus evaluasi pada classifier kedua, dan begitu seterusnya. Hasil negatif pada stage manapun akan dilakukan penolakan. Stage dalam cascade dilatih dengan menggunakan algoritma Ada Boost classifier dan nilai tresholdnya divariasikan untuk meminimalisir false negatives. Dengan demikian semakin banyak jumlah stage maka akan menghasilkan akurasi yang lebih tinggi namun akan memakan waktu training yang lebih lama.

Setelah proses training maka model akan menghasilkan classifier pada masing masing stage, dan pada akhirnya semua classifier tersebut digabungkan dan membentuk file cascade.xml, yang akan digunakan untuk melakukan pengujian untuk mendeteksi masker pada citra.

III. HASIL DAN PEMBAHASAN

Pengujian dilakukan dengan melakukan prediksi pada 20 citra wajah bermasker dan tidak bermasker. Hasilnya, didapati bahwa 18 dari 20 citra terprediksi secara tepat. Ini menunjukkan bahwa sistem memiliki akurasi terhadap data uji sebesar 90%, dapat dilihat pada Gambar 3.

Meskipun demikian, terkadang sistem masih menghasilkan lokalisasi yang kurang akurat seperti terlihat pada Gambar 4.



Gambar 3. Contoh hasil prediksi benar (kiri: kelas menggunakan masker, kanan: kelas tanpa masker).



Gambar 4. Contoh hasil prediksi benar namun lokalisasi kurang tepat.

Tabel 2. Efek Perubahan Scalefactor Terhadap Akurasi

scaleFactor	minNeighbors	Akurasi
1,1	3	90%
1,2	3	85%
1,3	3	65%
1,4	3	65%
1,5	3	55%

Di sini penulis menggunakan parameter scaleFactor untuk face dan mask cascade sebesar 1,1 serta parameter minNeighbors sebesar 3. Kedua nilai ini didapat secara eksperimental sebagaimana sehingga mampu menghasilkan akurasi paling baik. Dengan konfigurasi seperti ini, sistem mampu bekerja dengan kecepatan pemrosesan rata-rata sebesar 5,82 FPS (Frames per Second). Tabel 2 menunjukkan bahwa bertambahnya nilai scaleFactor membuat kecepatan pemrosesan meningkat. Hal ini terjadi karena lompatan skala ukuran filter yang digunakan semakin besar yang mengakibatkan semakin

sedikitnya jumlah filter. Di sisi lain, perubahan nilai minNeighbors bisa dikatakan tidak berpengaruh pada kecepatan pemrosesan. Terbukti pada Tabel 3 bahwa meningkatnya jumlah minNeighbors (ambang batas penentuan hasil deteksi) tidak mengakibatkan perubahan FPS yang signifikan.

Tabel 3. Eksperimen terhadap variabel scaleFactor

scaleFactor	minNeighbors	FPS
1,2	3	7,9
1,3	3	9,6
1,4	3	11,86
1,5	3	12,25

Tabel 4. Eksperimen terhadap variabel minNeighbors

scaleFactor	minNeighbors	FPS
1,3	4	10,45
1,3	5	10,02
1,3	6	9,49
1,3	7	9,99

Terdapat beberapa skenario simulasi untuk mengetahui apakah sistem yang dikembangkan layak untuk diterapkan pada situasi real-world. Beberapa skenario tersebut meliputi penggunaan kacamata, topi, warna masker berbeda, serta jarak dari kamera.

Pada eksperimen pertama tidak ada aksesoris wajah apapun yang digunakan. Hasilnya, sistem dapat mendeteksi wajah dengan mudah. Hal ini terjadi karena selain menggunakan Haar cascade untuk objek masker, sistem juga menggunakan Haar cascade untuk mendeteksi wajah dan hidung. Model Haar cascade wajah itu sendiri diambil dari repositori GitHub resmi milik OpenCV sehingga wajar saja jika hasilnya sangat baik. Gambar 5 menunjukkan screenshot dari eksperimen pertama.



Gambar 5. Hasil eksperimen pertama.

Berikutnya, pada eksperimen kedua dilakukan simulasi dengan menggunakan kacamata yang terlihat pada Gambar 6. Hasilnya wajah tetap dapat terdeteksi oleh sistem dengan baik. Ini menunjukkan bahwa fitur Haar dari wajah tanpa dan dengan kacamata memiliki tingkat kemiripan yang cukup tinggi. Sayangnya, sistem mulai tidak dapat bekerja ketika topi digunakan yang terlihat pada Gambar 7. Hal ini mungkin terjadi karena model mengenali wajah salah satunya berdasarkan keberadaan rambut. Ketika topi digunakan, fitur-fitur yang ditangkap

dari warna rambut tadi tidak tampak, sehingga wajah tidak terdeteksi. Begitu pula dengan hidung. Pada kasus ini hidung juga tidak terdeteksi karena sepertinya bayang-bayang topi membuat hidung terlihat lebih gelap dari biasanya.



Gambar 6. Hasil eksperimen kedua.

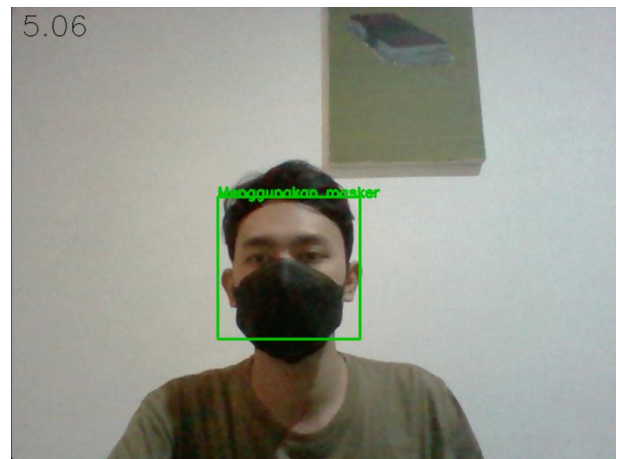


Gambar 7. Hasil eksperimen ketiga. Wajah tidak terdeteksi ketika topi digunakan.

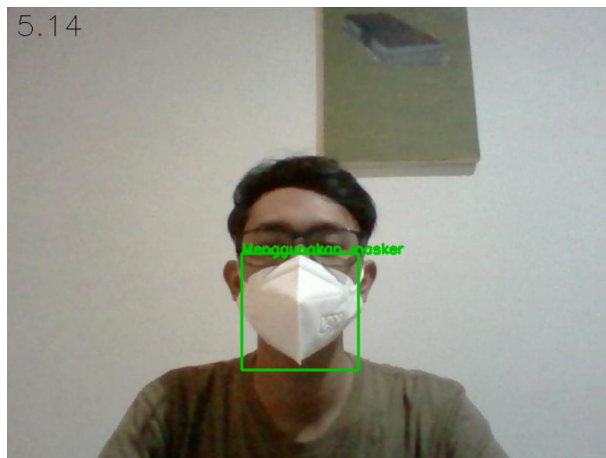
Selanjutnya, hasil eksperimen keempat, kelima, dan keenam yang terlihat pada Gambar 8, Gambar 9 dan Gambar 10 menunjukkan bahwa sistem mampu mendeteksi masker pada wajah baik pada saat orang tidak menggunakan kacamata, memakai kacamata, dan memakai topi secara berturut-turut. Meskipun hasil eksperimen sebelumnya menunjukkan bahwa adanya topi mengakibatkan model tidak mampu mendeteksi wajah, namun pada kenyataannya masker justru dapat terdeteksi. Ini terjadi karena pada dasarnya sistem melakukan deteksi masker tanpa perlu melakukan pendeteksian wajah terlebih dahulu.



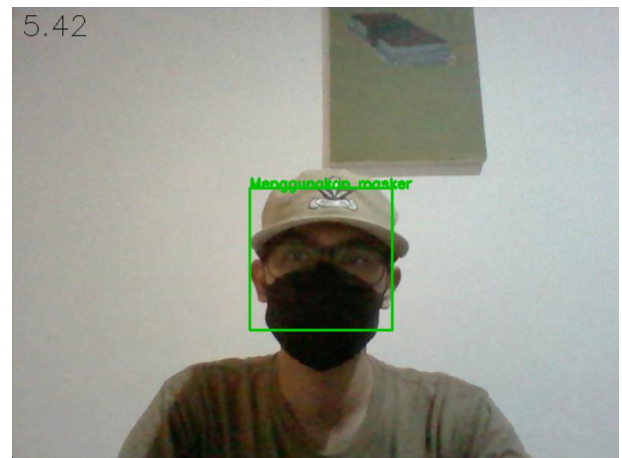
Gambar 8. Hasil eksperimen keempat. Masker dapat terdeteksi dengan baik.



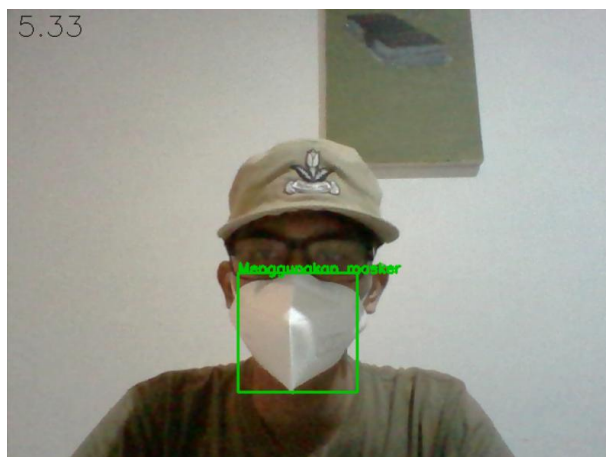
Gambar 11. Hasil eksperimen ketujuh. Masker berwarna gelap dapat terdeteksi.



Gambar 9. Hasil eksperimen kelima. Masker dapat terdeteksi ketika orang memakai kacamata.



Gambar 12. Hasil eksperimen kedelapan. Masker berwarna gelap juga dapat terdeteksi ketika aksesori wajah digunakan.



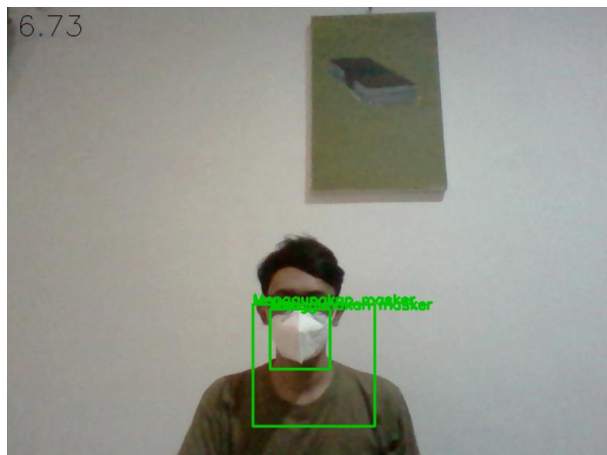
Gambar 10. Hasil eksperimen keenam. Masker tetap terdeteksi meskipun topi dan kacamata digunakan.

Selain masker warna putih, di sini penulis juga menggunakan masker warna hitam seperti pada Gambar 11 dan Gambar 12. Hasil eksperimen ketujuh dan kedelapan menunjukkan bahwa sistem juga mampu mendeteksi masker berwarna gelap.

Pengujian terkait jarak kamera terhadap objek juga dilakukan yang ditunjukkan pada Gambar 13 dan 14. Hasil menunjukkan bahwa sistem mampu mendeteksi wajah tanpa masker dengan cukup baik. Meskipun demikian, terkadang muncul beberapa false detection, yaitu objek-objek yang sebenarnya bukan merupakan sebuah masker justru terprediksi sebagai masker. Selain itu, kesalahan dalam lokalisasi juga terkadang masih dapat ditemukan.



Gambar 13. Hasil eksperimen kesembilan. Wajah non masker dapat terdeteksi pada jarak yang lebih jauh.



Gambar 14. Hasil eksperimen kesepuluh. False detection terkadang muncul meskipun secara visual sangat berbeda dari objek yang diinginkan.

Hal lain yang perlu dilihat di sini adalah bahwa sistem yang dikembangkan mampu mendeteksi kesalahan penggunaan masker. Hasil eksperimen pada Gambar 15 menunjukkan bahwa seseorang dianggap tidak menggunakan masker karena hidung yang seharusnya tertutup masih terlihat.



Gambar 15. Hasil eksperimen kesebelas. Sistem terkadang mampu mendeteksi penggunaan masker yang salah.

IV. KESIMPULAN

Berdasarkan pembahasan di atas, dapat disimpulkan bahwa sistem sudah berjalan dengan cukup baik. Meskipun demikian, ini bukan berarti bahwa sistem sudah sempurna. Terdapat beberapa error yang muncul pada kasus-kasus tertentu, yaitu seperti adanya deteksi false positive ketika jarak kamera terlalu jauh. Pada implementasinya, nilai-nilai input untuk parameter scaleFactor dan minNeighbors juga harus diatur dengan hati-hati secara eksperimental. Ini terjadi karena kedua nilai parameter tersebut, terutama scaleFactor, berkaitan erat dengan kecepatan pemrosesan. Artinya, jika ingin mendapatkan performa real-time, maka scaleFactor harus diperbesar, namun tidak boleh terlalu besar karena akan mengakibatkan turunnya akurasi.

DAFTAR PUSTAKA

- [1] D. Giancini, E. Yulia Puspaningrum, Y. Vita Via, U. Pembangunan Nasional, and J. Timur, "Seminar Nasional Informatika Bela Negara (SANTIKA) Identifikasi Penggunaan Masker Menggunakan Algoritma CNN YOLOv3-Tiny".
- [2] R. R. Mahurkar and N. G. Gadge, "Real-time Covid-19 Face Mask Detection with YOLOv4," in *Proceedings of the 2nd International Conference on Electronics and Sustainable Communication Systems, ICESC 2021*, Aug. 2021, pp. 1250–1255. doi: 10.1109/ICESC51422.2021.9533008.
- [3] N. Youssry and A. Khattab, "Accurate Real-Time Face Mask Detection Framework Using YOLOv5," in *4th IEEE International Conference on Design and Test of Integrated Micro and Nano-Systems, DTS 2022*, 2022. doi: 10.1109/DTS55284.2022.9809855.
- [4] T. Septiana, N. P. Putri, M. al Fikih, N. Setyawan, U. M. Malang, and M. Kontak, "Seminar Nasional Teknologi dan Rekayasa (SENTRA) 2020 ISSN (Cetak) 2527-6042 eISSN (Online)."
- [5] 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.
- [6] H. Goyal, K. Sidana, C. Singh, A. Jain, and S. Jindal, "A real time face mask detection system using convolutional neural network," *Multimed Tools Appl*, vol. 81, no. 11, pp. 14999–15015, May 2022, doi: 10.1007/s11042-022-12166-x.
- [7] S. Shivaprasad, M. Dinesh Sai, U. Vignasahithi, G. Keerthi, S. Rishi, and P. Jayanth, "Real Time Cnn Based Detection of Face Mask Using Mobilenetv2 to Prevent Covid-19," 2021. [Online]. Available: <http://annalsofscsb.ro>
- [8] J. Rani, S. Veluswami, S. Prakash, S. Charan B, and N. Parekh, "2 nd International Conference on IoT Based Control Networks and Intelligent Systems (ICICNIS 2021) Face mask detection using SSDNET and lightweight custom CNN." [Online]. Available: <https://ssrn.com/abstract=3882472>
- [9] VinithaV and VelantinaV, "Covid-19 Facemask Detection With Deep Learning And Computer Vision," *International Research Journal of Engineering and Technology*, 2020, [Online]. Available: www.irjet.net
- [10] P. Nagrath, R. Jain, A. Madan, R. Arora, P. Kataria, and J. Hemanth, "SSDMNV2: A real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2," *Sustain Cities Soc*, vol. 66, Mar. 2021, doi: 10.1016/j.scs.2020.102692.
- [11] G. J. Chowdary, N. S. Punna, S. K. Sonbhadra, and S. Agarwal, "Face Mask Detection using Transfer Learning of InceptionV3," Sep. 2020, doi: 10.1007/978-3-030-66665-1_6.
- [12] H. Santoso and A. Harjoko, "Haar Cascade Classifier dan Algoritma Adaboost untuk Deteksi Banyak Wajah dalam Ruang Kelas," *Jurnal Teknologi AKPRIND*, vol. 6, no. 2, pp. 108–115, 2013, [Online]. Available: http://jurtek.akprind.ac.id/sites/default/files/108-115_santoso.pdf.
- [13] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1, no. February, 2001, doi: 10.1109/cvpr.2001.990517.