

Deteksi Objek Masker Menggunakan EfficientDet-Lite3

Ferdian Nursulistio
Program Studi Informatika
Universitas Islam Indonesia
Yogyakarta, Indonesia
18523149@students.uii.ac.id

Arrie Kurniawardhani
Program Studi Informatika
Universitas Islam Indonesia
Yogyakarta, Indonesia
arrie.kurniawardhani@uui.ac.id

Dhomas Hatta Fudholi
Program Studi Informatika
Universitas Islam Indonesia
Yogyakarta, Indonesia
hatta.fudholi@uui.ac.id

Abstrak— Masker merupakan benda yang digunakan untuk menutupi mulut dan hidung untuk mencegah menghirup dan melepaskan cairan pernafasan. Jenis masker dibagi dua, yaitu masker medis dan non-medis yang biasa dibuat dari tekstil atau bahan lainnya yang dapat dipakai berulang kali. Baik menggunakan masker medis maupun masker kain keduanya dapat menyaring dengan baik partikel atau cairan dengan tingkat efisiensi 86.4% untuk masker kain dan 99.9% untuk masker medis 3M. Masker digunakan untuk mencegah penyakit menular seperti pada pandemi COVID-19 serta juga digunakan untuk melindungi petugas kesehatan untuk mencegah Infeksi Nosokomial di rumah sakit. Maka dibuatlah model *object detection* untuk mendeteksi apakah seseorang menggunakan masker medis, masker kain, masker scuba, tidak menggunakan masker dengan benar, atau tidak menggunakan masker. Metode yang digunakan adalah TensorFlow Lite Model Maker dengan model dasar EfficientDet-Lite3. Model yang dihasilkan sebanyak dua jenis dengan rata-rata presisi sebesar 75.82%, 75.84% dan rata-rata recall 82.61%, 82.25% untuk model pertama dan kedua berturut-turut.

Kata Kunci— Masker Object Detection, TensorFlow Lite Model Maker, Object Detection

I. PENDAHULUAN

Masker wajah adalah istilah luas yang digunakan untuk benda apapun yang dikenakan di atas mulut dan hidung untuk mencegah menghirup cairan penyakit menular atau pelepasan cairan penyakit menular yang berasal dari sistem pernafasan. Cairan tersebut dihasilkan Ketika bernafas, berbicara, batuk, dan bersin. Salah satu jenis masker adalah masker medis dan masker non-medis. Masker medis adalah masker sekali pakai yang biasa digunakan pada petugas Kesehatan sedangkan masker non medis adalah berbagai macam bentuk masker yang dapat dibuat sendiri termasuk masker yang dapat digunakan berkali-kali yang dibuat dari kain atau tekstil lainnya [1].

Masker sangatlah direkomendasikan karena dapat mencegah cairan pernafasan penderita lepas ke lingkungan sekitar dan dihirup oleh orang sehat lainnya [1]. Baik menggunakan masker medis maupun masker kain keduanya dapat menyaring dengan baik partikel atau cairan dengan tingkat efisiensi 86.4% untuk masker kain dan 99.9% untuk masker medis 3M [2] sehingga dapat menurunkan tingkat penularan penyakit menular.

Sebagai contoh, pada tahun 2019 sampai 24 Mei 2022, WHO mencatat ada lebih dari 523 juta kasus positif dengan lebih dari enam juta orang meninggal di seluruh dunia. Di antaranya, 6,053,109 kasus positif berada di Indonesia dengan 156 ribu orang meninggal [3]. COVID-19 dapat menular dengan beberapa cara, yaitu melalui kontak langsung seperti batuk, bersin, atau saat berbicara dan secara tidak langsung

melalui benda yang sudah terkena cairan pernafasan penderita COVID-19 dan melalui udara [4]. Selain pada kasus Covid, masker juga penting digunakan sebagai alat pelindung diri pada petugas kesehatan untuk mencegah Infeksi Nosokomial di rumah sakit [5].

Object detection memiliki kemampuan dalam mendeteksi sebuah objek dalam suatu gambar seperti wajah, tumbuhan, kendaraan, melacak pergerakan orang, atau objek lainnya sehingga automasi sebuah pekerjaan dapat ditingkatkan [6]. Dengan menggunakan *EfficientDet-Lite3* yang merupakan sebuah model dengan arsitektur yang mengedepankan efisiensi, proses automasi ini dapat berjalan dengan lancar [7].

Memakai masker sangat penting, tetapi terdapat kesulitan serta memerlukan usaha lebih dalam memantau satu persatu dan mengingatkan apakah seseorang menggunakan masker sesuai standar. Maka dari itu dibuatlah model *object detection* untuk mendeteksi apakah seseorang mengenakan masker beserta jenis maskernya, tidak memakai masker, atau tidak mengenakan masker dengan tepat.

II. KAJIAN PUSTAKA

A. TensorFlow

Tensorflow adalah *library open source* yang dibuat oleh Google untuk melakukan pekerjaan seputar *deep learning*. *Tensorflow* menggunakan grafik aliran data komputasi untuk merepresentasikan arsitektur jaringan saraf tiruan yang rumit. Perlahan-lahan *TensorFlow* menjadi *library* yang sering digunakan untuk penelitian seputar *deep learning* dan implementasi pada proyek tertentu serta digunakan untuk proyek yang membutuhkan implementasi di *cloud* [8].

B. Object Detection

Object detection atau dalam bahasa Indonesia disebut deteksi objek, adalah sebuah sistem yang bertujuan untuk menemukan dan mengklasifikasi sebuah objek dalam sebuah gambar [6].

C. Efficient-Det

Efficient-Det merupakan salah satu model yang dapat digunakan untuk melatih sistem. Model ini didasarkan pada jaringan saraf tiruan yang menggunakan algoritma *Convolutional Neural Network (CNN)*. Model ini dapat digunakan untuk mendeteksi objek menggunakan sekumpulan data berupa gambar. Dalam perkembangannya, model ini dibuat untuk meningkatkan efisiensi pelatihan dengan menggunakan sejumlah kecil parameter, namun tetap mencapai akurasi yang baik [9].

EfficientDet-Lite sendiri merupakan sebuah versi *Efficient-Det* yang sudah dioptimalkan untuk perangkat

bergerak dengan menggunakan ukuran *input* yang lebih kecil dan fungsi aktivasi yang lebih sederhana tetapi masih memiliki arsitektur dan pemrosesan yang sama dengan *Efficient-Det* [7].

D. Precision dan Recall

Precision adalah proporsi dari hasil prediksi positif dengan jawaban yang *real positive*. Sedangkan *recall* adalah kebalikan dari *precision*, yaitu proporsi *real positive* yang terprediksi positif [10]. Dari hasil komputasi kedua nilai tersebut akan ditentukan nilai AP (*Average Precision*) dengan menghitung rata-rata nilai presisi dari nilai keseluruhan recall. Perhitungan *precision*, *recall*, dan AP dapat ditulis dengan persamaan sebagai berikut:

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (1)$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (2)$$

$$AP = \frac{1}{11} \sum_{Recall_{i-11}} Precision(Recall_i) \quad (3)$$

True Positive adalah ketika model berhasil melakukan deteksi objek dan hasil prediksi sesuai dengan label, *False Positive* adalah ketika model berhasil melakukan deteksi objek namun hasil prediksi berbeda dengan label, *False Negative* adalah model tidak dapat mendeteksi objek sama sekali [10].

E. Penelitian sebelumnya

Sudah ada beberapa penelitian pendeteksi masker yang sudah dilakukan sebelumnya dengan menggunakan berbagai macam metode. Walaupun topik yang diteliti sama, dari ketiga penelitian yang dianalisis semuanya memiliki perbedaan seperti penggunaan *dataset*, pemilihan metode atau *pre-trained* model, dan hasil yang didapatkan.

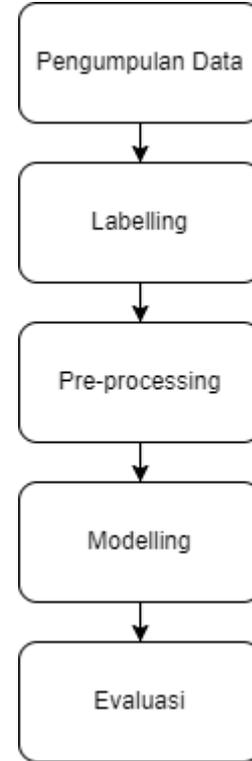
1. Penelitian yang dilakukan oleh Suressh, K dan kawan-kawan yang membuat model deteksi masker *real time* menggunakan *Convolution Neural Network* (CNN). Model yang dibuat dapat diimplementasikan pada kamera pengawas sekolah, universitas, pusat perbelanjaan, dan lain-lain. Penelitian ini memonitor seseorang dan mendeteksi apakah mereka menggunakan masker [11].
2. Pada penelitian yang dilakukan oleh Yadav, S. Sistem yang dibuat ditujukan untuk memastikan kesehatan masyarakat yang berada di area publik dengan cara memantau jarak aman antar orang dan mendeteksi individu yang memakai atau tidak memakai masker. Sistem ini menggunakan metode *transfer learning* dengan menggunakan arsitektur *MobileNetV2* dan *SSD MultiBox* sebagai model inti pendeteksi [12].
3. Penelitian yang dilakukan oleh Rahman dan kawan-kawan yang membuat model untuk mendeteksi orang yang tidak memakai masker melalui kamera CCTV. Arsitektur yang digunakan adalah CNN [13].

Berbeda dengan penelitian yang sebelumnya yang hanya memiliki dua label yaitu memakai masker dan tidak memakai masker, penelitian ini menggunakan lima label yaitu tidak memakai masker, kurang tepat dalam memakai masker, dan

menggunakan masker beserta jenis masker yang digunakan seperti masker medis, masker kain, dan masker scuba.

III. METODOLOGI

Dalam penelitian ini, alur penelitian yang harus dilakukan terbagi menjadi beberapa tahap. Tahapan tersebut adalah pengumpulan dataset, pelabelan data, *modeling*, dan evaluasi yang dapat divisualisasikan pada gambar 1.



Gambar 1. Alur penelitian

A. Pengumpulan Dataset

Data yang digunakan diambil dari dataset terbuka yang tersedia di internet. Fokus utama dalam pengumpulan dataset ini adalah menemukan gambar wajah seseorang yang memakai masker dengan tiga jenis masker yaitu masker medis, scuba, dan kain, serta gambar wajah seseorang tidak memakai masker dengan benar dan tidak memakai masker sama sekali. Contoh gambar untuk setiap kelas tersebut dapat dilihat pada gambar 2. Untuk melakukan penelitian ini, dataset yang dibutuhkan lebih dari satu jenis karena satu dataset tidak memenuhi untuk setiap kelas yang akan dideteksi serta untuk menambah variasi data. Dataset yang digunakan adalah *Masked Face (MAFA)* [14], *Flickr-Faces-HQ* [15], *MaskedFace-Net* [16], *Face Mask Detection Dataset* [17].

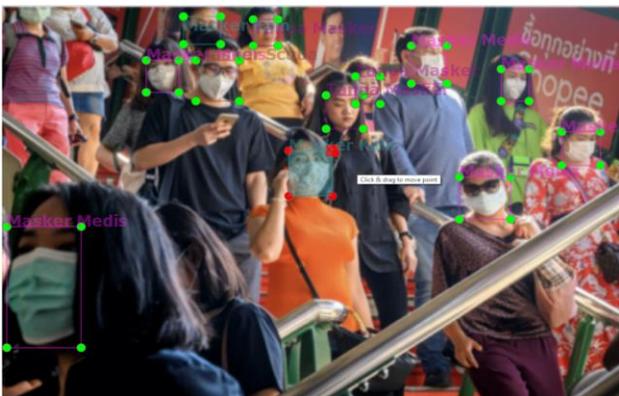


Gambar 2. Contoh gambar dari dataset yang digunakan

Jumlah foto pada keempat dataset tersebut sangat besar sehingga tidak semua foto yang tersedia pada dataset tersebut digunakan. Meski begitu, terdapat kelas yang sulit ditemukan pada keempat dataset tersebut sehingga ditambahkan gambar yang diambil dari *Instagram* dengan cara *scraping* menggunakan *library Instaloader* [18]. Pada penelitian ini, jumlah gambar yang digunakan dibagi menjadi dua jenis, yaitu dataset bias dengan sebanyak 986 foto dengan jumlah total 1310 label dengan masing-masing 333 untuk masker medis, 212 masker kain, 201 masker scuba, 202 penggunaan masker salah, dan 362 tanpa masker. Serta dataset yang sudah disamaratakan dengan jumlah total 1274 foto dengan jumlah label 1638 dengan pembagian 336 masker medis, 303 masker kain, 314 masker scuba, 313 penggunaan masker salah dan 372 tanpa masker. Sedangkan untuk data validasi dan data *test* berisi gambar yang berbeda dari data *train* dan berisi sekitar 30 sampai 40 label untuk masing-masing kelas.

B. Labeling

Proses *labeling* merupakan proses pembuatan *bounding-box* untuk menandakan setiap objek yang ingin dideteksi [19]. Proses ini menggunakan *LabelImg* yang merupakan sebuah *tool* untuk membuat anotasi dan *bounding-box* pada sebuah gambar seperti yang terlihat pada gambar 3. Aplikasi ini dijalankan dengan bahasa *python* dan menggunakan *Qt* sebagai antarmuka grafis [20]. Aplikasi ini memudahkan dalam proses *labelling* karena adanya *graphical user interface (GUI)* sehingga dapat membuat *bounding-box* dan mengganti ke gambar selanjutnya dengan cepat. Luaran yang dihasilkan pada aplikasi ini adalah file *xml* dengan jenis label *PASCAL Visual Object Classes (VOC)* [19].



Gambar 3. Contoh Pemberian Label

C. Pre-Processing

Input size pada model ini beragam tergantung *pre-trained* model yang dipilih. Untuk *EfficientDet-Lite3* ukurannya adalah 512x512 pixel. Gambar pada dataset memiliki ukuran yang beragam, namun pada *library tensorflow lite model maker* sudah melakukan *resize* untuk menyesuaikan ukuran gambar *input* menjadi sesuai dengan model *input*.

Meski begitu *library* ini memiliki kekurangan yaitu tidak dapat menerima gambar dengan ekstensi selain 'jpeg' sehingga *pre-processing* tetap perlu dilakukan untuk mengubah gambar yang memiliki ekstensi 'png' dan 'jpg' menjadi 'jpeg' dengan melakukan *re-encoding* dengan menggunakan *library cv2*. Selanjutnya adalah menyesuaikan nama foto pada tag 'filename' pada file label (.xml) dengan mengganti tulisan ekstensi 'png' atau 'jpg' menjadi 'jpeg' menggunakan *library lxml*.

D. Modeling

Proses *modeling* ini menggunakan bantuan *library TensorFlow Lite Model Maker*. *Library* ini menggunakan *transfer learning* untuk menyederhanakan proses pembuatan model dengan dataset sendiri. Metode ini membuat proses *training* berjalan singkat dan hanya membutuhkan data yang sedikit. Pada saat penelitian ini dibuat, *TensorFlow Model Lite Model Maker* hanya menyediakan satu jenis model arsitektur yang dibagi menjadi lima ukuran [21] dengan pilihan model seperti pada tabel 1.

Tabel 1. Pilihan model *Tensorflow Lite Model Maker*

Model	Size (MB)	Latency (ms)	Average Precision
<i>EfficientDet-Lite0</i>	4.4	37	25.69%
<i>EfficientDet-Lite1</i>	5.8	49	30.55%
<i>EfficientDet-Lite2</i>	7.2	69	33.97%
<i>EfficientDet-Lite3</i>	11.4	116	37.70%
<i>EfficientDet-Lite4</i>	19.9	260	41.96%

Pemilihan model berdasarkan ukuran, presisi, dan latensi. Model *EfficientDet-Lite3* dipilih karena model tersebut memiliki akurasi yang tinggi namun memiliki ukuran dan latensi yang cukup rendah dibanding dengan *EfficientDet-Lite4*.

E. Evaluasi

Model yang sudah dilatih selanjutnya akan dievaluasi menggunakan data yang belum pernah dilihat oleh model tersebut. Proses evaluasi berguna untuk mengetahui performa model dengan melihat tingkat presisi model tersebut [22]. Matriks penilaian yang digunakan adalah matriks *Average Precision* seperti pada persamaan 3. Model yang akan dievaluasi adalah kedua model dengan dataset bias dan dataset yang sudah disamaratakan.

IV. HASIL DAN PEMBAHASAN

Di bawah ini adalah hasil dari penelitian deteksi objek masker menggunakan *EfficientDet-Lite3* dengan bantuan *library TensorFlow Lite Model Maker*. Pengujian dibagi menjadi dua skenario yaitu dataset bias disebut sebagai

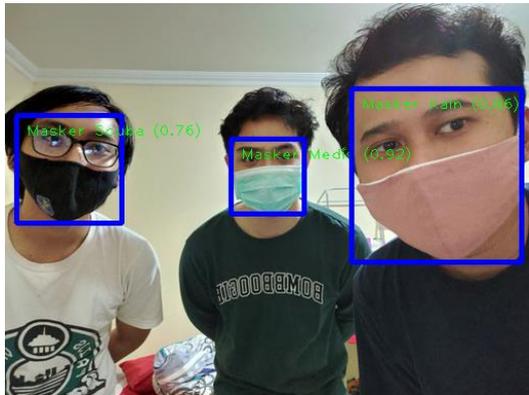
skenario pertama dan dataset yang sudah disamaratakan disebut sebagai skenario kedua.

Pada skenario pertama menggunakan dataset bias dengan jumlah label untuk beberapa kelas tidak merata. Pada percobaan pertama *training* dengan *hyperparameter* sebagai berikut:

- *Epoch*: 35
- *Batch size*: 8
- *Train whole model*: True

Jumlah *epoch* ini merupakan nilai pertama yang dicobakan lalu akan dievaluasi nilai *loss* serta perbandingan antara besar presisi pada data *train* dan data validasi. *Batch size* dipilih berdasarkan kemampuan komputasi komputer yang terbatas sehingga tidak dapat memilih ukuran *batch* yang lebih besar.

Setelah melakukan training selama enam puluh menit, model pertama memiliki rata-rata presisi sebesar 82.4% pada data *train* dan 73.1% pada data validasi, namun memiliki *loss* yang seimbang, yaitu 0.2791 pada data *train* dan 0.2740 pada data *test*. Karena besarnya perbandingan antara presisi data *test* dan data *train*, maka dilakukan *training* lagi tetapi dengan *epoch* yang lebih kecil yaitu sebanyak 25 *epoch*. Percobaan kedua ini menghasilkan presisi sebesar 80.3% pada data *train* dan 70% pada data validasi. Selisih nilai presisi pada percobaan kedua kurang lebih sama dengan selisih presisi percobaan pertama, sehingga model yang akan diujikan adalah model dari percobaan pertama karena nilai presisi data *train* dan data validasi lebih besar. Adapun contoh deteksi dari model skenario pertama dapat dilihat pada gambar 4.



Gambar 4. Contoh hasil deteksi model pada skenario pertama

Model dari skenario pertama ini memiliki rata-rata presisi sebesar 75.82% dan rata-rata *recall* 82.61%. Adapun presisi untuk masing-masing kelas dapat dilihat pada tabel 2.

Tabel 2. Presisi untuk setiap kelas pada model skenario pertama

Kelas	Presisi
Masker Medis	83.4%
Masker Kain	74.1%
Masker Scuba	74.9%
Penggunaan Masker Salah	68.1%
Tanpa Masker	78.4%

Selain pada data *test*, kemampuan model juga diuji dalam mendeteksi kelima kelas tersebut dari berbagai jarak. Hasil pengujian ini pada skenario pertama mendapat hasil yang dapat dilihat pada tabel 3.

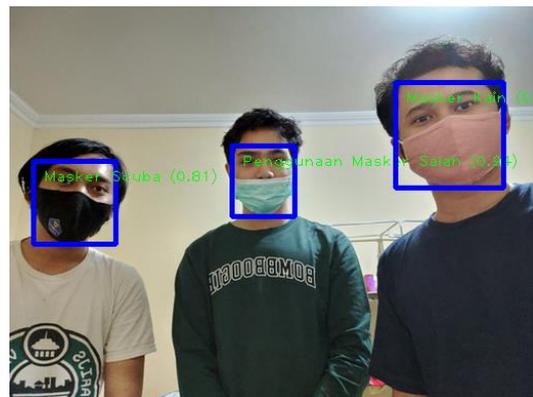
Tabel 3. Hasil deteksi dalam berbagai jarak pada model skenario pertama

Kelas	Jarak		
	Dekat	Sedang	Jauh
Masker Medis	Benar	Benar	Benar
Masker Kain	Benar	Benar	Benar
Masker Scuba	Benar	Benar	Benar
Penggunaan Masker Salah	Benar	Benar	Salah
Tanpa Masker	Benar	Benar	Benar

Pada skenario kedua menggunakan dataset yang sudah diseimbangkan labelnya untuk setiap kelas. Dengan menggunakan *hyperparameter* yang sama seperti skenario pertama, model pada skenario kedua mendapat rata-rata presisi sebesar 75.84% dan rata-rata *recall* 82.25%. Adapun presisi untuk masing-masing kelas dapat dilihat pada tabel 4 dan contoh hasil deteksi dapat dilihat pada gambar 5.

Tabel 4. Presisi untuk setiap kelas pada model skenario kedua

Kelas	Presisi
Masker Medis	68.7%
Masker Kain	82.7%
Masker Scuba	76.7%
Penggunaan Masker Salah	72.5%
Tanpa Masker	78.4%



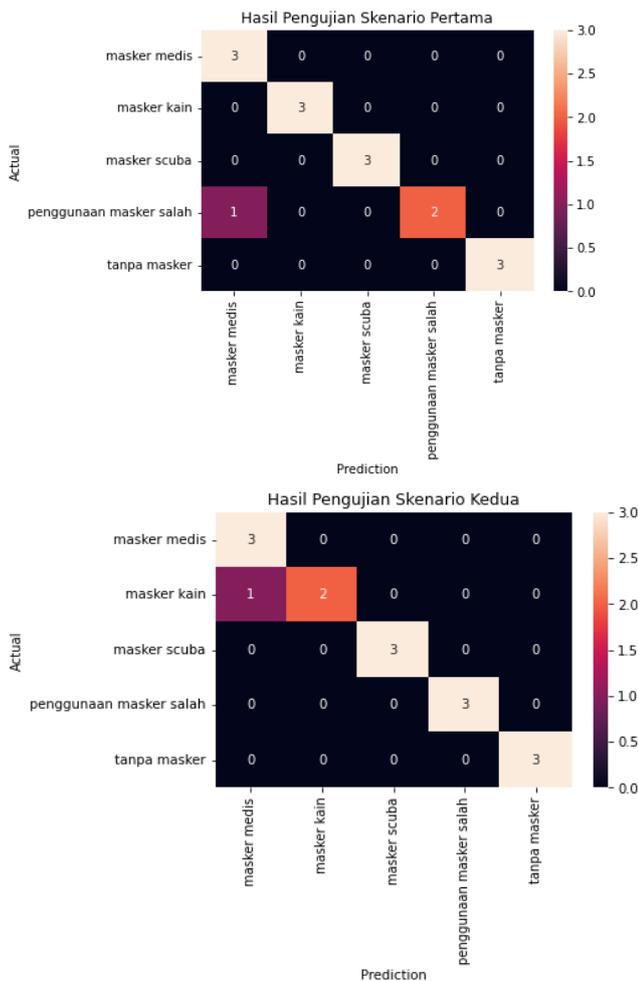
Gambar 5. Contoh hasil deteksi model pada skenario kedua

Adapun hasil pengujian untuk berbagai jarak pada skenario kedua dapat dilihat pada tabel 5.

Tabel 5. Hasil deteksi dalam berbagai jarak pada model skenario kedua

Kelas	Jarak		
	Dekat	Sedang	Jauh
Masker Medis	Benar	Benar	Benar
Masker Kain	Benar	Benar	Salah
Masker Scuba	Benar	Benar	Benar
Penggunaan Masker Salah	Benar	Benar	Benar
Tanpa Masker	Benar	Benar	Benar

Pada kedua hasil pengujian ini dapat dibuat perbandingan *confusion matrix* seperti gambar 6 untuk melihat hasil prediksi setiap kelasnya.



Gambar 6. Confusion matrix untuk kedua pengujian

Pada gambar 6 dapat dilihat bahwa skenario pertama dan kedua model berhasil membuat *bounding box* pada semua objek, namun salah memberi nama kelas untuk objek tersebut. Objek yang dimaksud adalah penggunaan masker salah terdeteksi sebagai masker medis pada skenario pertama dan pada skenario dua masker kain salah terdeteksi sebagai masker medis.

Pada tabel 2 dan 4, nilai presisi pada setiap kelas mengalami kenaikan dan penurunan, untuk kenaikan presisi mungkin disebabkan oleh bertambahnya dataset sehingga varian untuk kelas tertentu bertambah, sedangkan penurunan presisi mungkin disebabkan varian tambahan tersebut memiliki fitur yang mirip dengan kelas lainnya.

V. KESIMPULAN

Pada penelitian ini dilakukan deteksi objek masker yang dibagi menjadi lima kelas, yaitu masker medis, masker scuba, masker kain, penggunaan masker salah, dan tanpa masker dengan menggunakan *EfficientDet-Lite3*. Model dari skenario pertama ini memiliki rata-rata presisi sebesar 75.82% dan rata-rata *recall* 82.61%, sedangkan model pada skenario kedua mendapat rata-rata presisi sebesar 75.84% dan rata-rata *recall* 82.25%. Dapat dikatakan model dengan dataset seimbang memiliki nilai rata-rata presisi 0.02% lebih tinggi dari dataset bias namun memiliki penurunan 0.36% pada nilai *recall*. Setelah dilakukan pengujian lebih lanjut, kedua model berhasil mendeteksi 14 dari 15 gambar yang berisi kelima kelas dari berbagai jarak. Meski begitu terdapat kenaikan dan

penurunan presisi pada kelas tertentu sehingga pada penelitian selanjutnya disarankan untuk menambah dataset atau mencoba arsitektur lain.

REFERENSI

- [1] European Centers for Disease Control (ECDC). (2022). Considerations for the use of face masks in the community in the context of the SARS-CoV-2 Omicron variant of concern. [online]. Available: <https://www.ecdc.europa.eu/en/publications-data/using-face-masks-community-reducing-covid-19-transmission>
- [2] Ho, K. F., Lin, L. Y., Weng, S. P., & Chuang, K. J. (2020). Medical mask versus cotton mask for preventing respiratory droplet transmission in micro environments. *Science of the Total Environment*, 735.
- [3] Covid19.who.int. WHO Coronavirus (COVID-19) Dashboard. 2022. [online]. Available: <https://covid19.who.int/>
- [4] Lotfi, M., Hamblin, M. R., & Rezaei, N. (2020). COVID-19: Transmission, prevention, and potential therapeutic opportunities. In *Clinica Chimica Acta* (Vol. 508, pp. 254–266). Elsevier B.V.
- [5] Salawati, L. (2012). Pengendalian infeksi nosokomial di ruang intensive care unit rumah sakit. *Jurnal Kedokteran Syiah Kuala*, 12(1), 47–52.
- [6] Zhao, Z. Q., Zheng, P., Xu, S. T., & Wu, X. (2019). Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems*, 30(11), 3212–3232.
- [7] Tomas, R. (2021). Fast object detection on mobile platforms using neural networks.
- [8] Scarpino, M. (2018). TensorFlow For Dummies. In *Climate Change 2013 - The Physical Science Basis*.
- [9] Tan, M., Pang, R., & Le, Q. V. (2020). Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 10781–10790).
- [10] Powers, D. M. (2020). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061*.
- [11] Suresh, K., Palangappa, M. B., & Bhuvan, S. (2021). Face Mask Detection by using Optimistic Convolutional Neural Network. *Proceedings of the 6th International Conference on Inventive Computation Technologies, ICICT 2021*, 1084–1089.
- [12] Yadav, S. (2020). Deep Learning based Safe Social Distancing and Face Mask Detection in Public Areas for COVID-19 Safety Guidelines Adherence. *International Journal for Research in Applied Science and Engineering Technology*, 8(7), 1368–1375.
- [13] Rahman, M. M., Manik, M. M. H., Islam, M. M., Mahmud, S., & Kim, J. H. (2020, September 1). An automated system to limit COVID-19 using facial mask detection in smart city network. *IEMTRONICS 2020 - International IOT, Electronics and Mechatronics Conference, Proceedings*.
- [14] Ge, S., Li, J., Ye, Q., & Luo, Z. (2017). Detecting masked faces in the wild with LLE-CNNs. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, 2017-January*.
- [15] Karras NVIDIA, T., & Laine NVIDIA, S. (2019). #StyleGAN - A Style-Based Generator Architecture for Generative Adversarial Networks Timo Aila NVIDIA. *Cvpr 2019*.
- [16] Cabani, A., Hammoudi, K., Benhabiles, H., & Melkemi, M. (2021). MaskedFace-Net–A dataset of correctly/incorrectly masked face images in the context of COVID-19. *Smart Health*, 19, 100144.
- [17] Larxel. Face Mask Detection. 2020. [online]. Available: <https://www.kaggle.com/andrewmvd/face-mask-detection>
- [18] Alexander Graf. Instaloader. 2022. [online]. Available: <https://instaloader.github.io/>
- [19] Yadav, S., & Shukla, S. (2016, February). Analysis of k-fold cross-validation over hold-out validation on colossal datasets for quality classification. In *2016 IEEE 6th International conference on advanced computing (IACC)* (pp. 78–83). IEEE.
- [20] Tzutalin. LabelImg. 2015. [online]. Available: <https://github.com/tzutalin/labelImg>
- [21] TensorFlow.org. Object Detection with TensorFlow Lite Model Maker. [online]. Available: https://www.tensorflow.org/lite/models/modify/model_maker/object_detection
- [22] Raschka, S. (2018). Model evaluation, model selection, and algorithm selection in machine learning. *arXiv preprint arXiv:1811.128*