Real Time Face Mask Detection Using MobileNetV2

Tim Mahasiswa:
Fradyta Bagust Ananta / 2702239782
Nicky Aryan Gunawan / 2702371376
Albertus Edbert Chandrajaya / 2702345440

WA PIC: 089674508888 - Nicky Aryan Gunawan

I. Introduction (Pendahuluan)

1.1. Latar Belakang

Image classification merupakan salah satu bidang utama dalam computer vision yang berfokus pada kemampuan sistem untuk mengenali dan mengelompokkan citra ke dalam kelas tertentu berdasarkan karakteristik visual. Teknologi ini telah banyak diterapkan dalam berbagai sektor, seperti healthcare, transportation, security, dan smart city, karena mampu membantu proses pengambilan keputusan secara otomatis dan efisien.

Pada kasus Face Mask Detection, tantangan semakin kompleks karena bentuk, warna, dan posisi masker dapat berbeda-beda pada setiap individu. Selain itu, adanya occlusion pada area wajah serta perbedaan kondisi lingkungan pengambilan gambar juga turut mempengaruhi hasil klasifikasi. Kesalahan prediksi, seperti mendeteksi *no mask* pada citra yang sebenarnya menggunakan masker, menunjukkan pentingnya sistem klasifikasi citra yang robust dan mampu beradaptasi terhadap variasi data.

Oleh karena itu, pengembangan sistem image classification yang andal menjadi hal yang penting untuk mendukung kebutuhan analisis visual secara otomatis. Sistem yang baik diharapkan mampu mengenali pola visual secara konsisten, memiliki tingkat akurasi yang tinggi, serta dapat diintegrasikan ke dalam aplikasi interaktif sehingga mudah digunakan oleh pengguna.

1.2. Tujuan

Adapun tujuan dari pengembangan sistem ini adalah sebagai berikut:

- Mengembangkan sistem **image classification** yang mampu mengklasifikasikan citra wajah ke dalam tiga kelas, yaitu *mask*, *no mask*, dan incorrect mask berdasarkan karakteristik visual yang terdapat pada citra.
- Menerapkan konsep **computer vision** dan **deep learning based image processing** untuk mengekstraksi fitur visual dari citra wajah secara otomatis tanpa memerlukan proses ekstraksi fitur manual.

- Mengintegrasikan sistem image classification ke dalam sebuah aplikasi interaktif berbasis web sehingga hasil klasifikasi dapat ditampilkan secara real-time dan mudah digunakan oleh pengguna.

1.3. Ruang Lingkup

Kami memiliki 5 tahapan utama dalam membangun dan mengembangkan model kami, berikut adalah tahapannya:

1. **EDA**

   Pada tahap awal ini, kami melakukan analisis mengenai dataset yang akan kami gunakan. Dataset kami terdiri dari 3 kelas, yaitu Mask, Incorrect Mask, dan Not Mask. Masing-masing kelas, sudah memiliki persebaran data yang sama banyak, sehingga kami tidak perlu lagi melakukan rebalance dataset saat proses preprocessing nanti.

2. **Preprocessing**

   Pada tahap kedua ini, kami hanya melakukan preprocessing simpel saja, seperti:

   - **Resize**
   - **Normalization**
   - **Data Augmentation**

3. **Modeling**

   Kami melakukan pengembangan model di Visual Studio Code dengan menggunakan library Tensorflow dan Keras. Ada beberapa catatan/perubahan penting yang kami lakukan agar sesuai dengan tema dan dataset yang kami pakai. Berikut adalah detailnya:

   1. **Transfer Learning**

      Menggunakan model MobileNetV2 yang sudah dilatih dengan dataset ImageNet sebagai base model guna mempercepat konvergensi dan meningkatkan akurasi.

   2. **Custom Layers**

      Menambahkan lapisan-lapisan custom diakhir guna menyesuaikan kebutuhan dari tema dan dataset yang kami gunakan, seperti menambahkan lapisan Average Pooling, Flatten, Dense, dan Dropout.

4. **Evaluasi & Validation**

   Model yang sudah di train, diuji dengan categorical crossentropy untuk pengukuran loss dan menggunakan accuracy sebagai metricsnya. hasilnya adalah model kami mendapatkan akurasi sebesar 94% dengan loss yang sangat rendah + tidak ada indikasi overfitting maupun underfitting.

5. **Deployment**

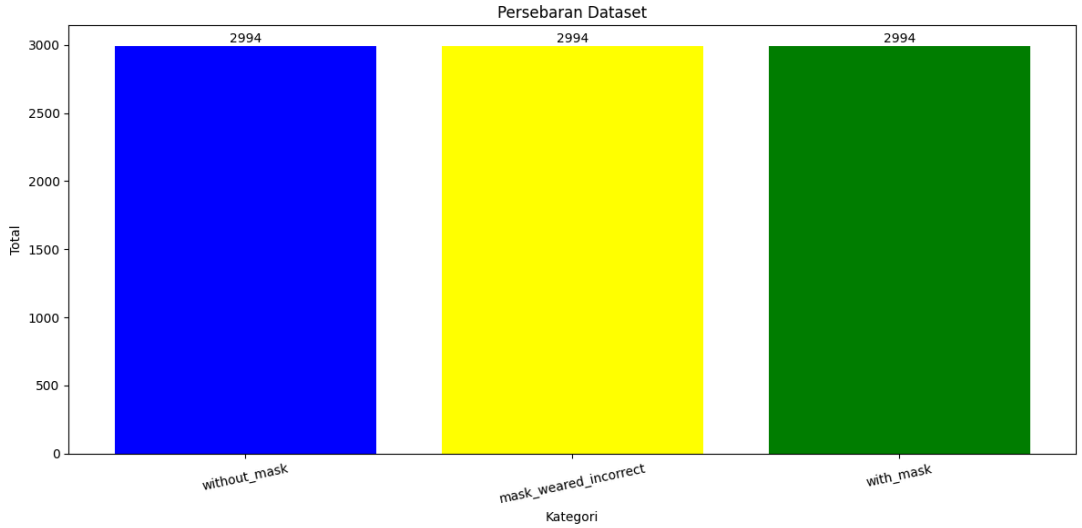
   Pada tahap akhir ini, kami mendeploy model kami dengan menggunakan framework streamlit. Dikarenakan Streamlit sangat mudah, cepat, ringan, dan bisa diakses lokal maupun global, kami memutuskan menggunakan framework Streamlit sebagai platform untuk mendemokan model yang sudah kami buat.

II. Dataset

2.1. Data Explanation

Kami menggunakan dataset yang sudah disediakan dari website kaggle. Dataset yang kami gunakan sudah terbagi menjadi 3 kelas yaitu mask, with out mask, dan mask weared incorrect. Total gambar dari dataset kami ada di 8982 dengan masing-masing kelas terdiri dari 2994 gambar. berikut dataset yang kelompok kami gunakan:
https://www.kaggle.com/datasets/vijaykumar1799/face-mask-detection

2.2. Eksplorasi (EDA)

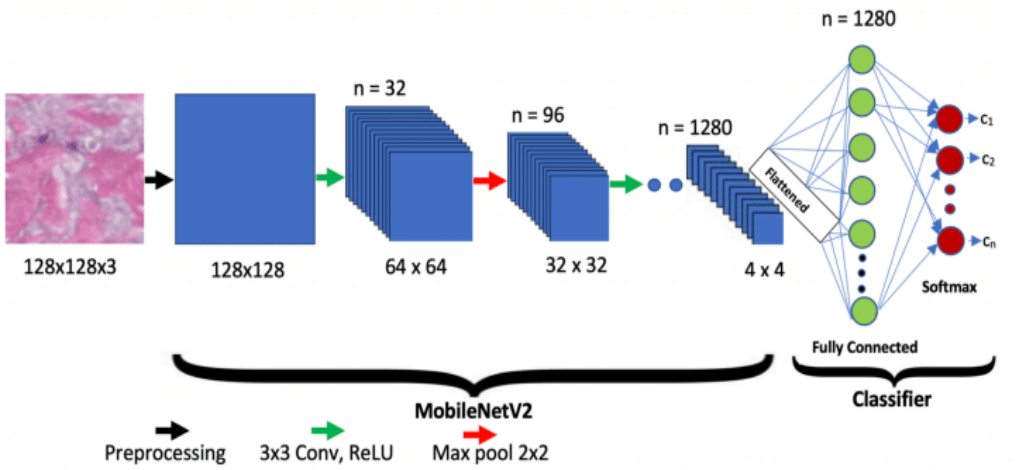


Persebaran Dataset

Dataset yang kami gunakan sudah memiliki struktur yang sangat bagus dari awal. Setiap kelas sudah mempunyai persebaran data yang sama banyak, sehingga kami tidak perlu melakukan rebalance dataset dan bisa langsung masuk ke preprocessing.

III. Modeling (Pemodelan)

3.1. Arsitektur Model



MobileNetV2 Architecture

Model yang kami gunakan dalam projek ini adalah MobileNetV2. MobileNetV2 adalah suatu model DCNN yang memiliki struktur utama yang terdiri dari lapisan konvolusi yang unik, yaitu Depthwise Separable Convolution dan Inverted Residual Blocks. Selain itu, kami juga melakukan transfer learning, dimana model kami telah dilatih sebelumnya menggunakan dataset ImageNet. Alasan kami menggunakan model ini adalah karena:

1. **Kebutuhan Real Time**
   Dikarenakan projek kami akan mendeteksi secara real time, jadi kami membutuhkan model yang tidak begitu berat dan mampu memproses frame vidio dari webcam secara cepat dan tetap memiliki akurasi yang baik. Sehingga kami dapat meminimalisir adanya lag atau patah-patah saat proses pendekteksian secara real time berlangsung.
2. **Efisiensi Komputasi yang Tinggi**
   Berkat adanya Depthwise Separable Convolution yang berfungsi sebagai pengurang jumlah parameter dan operasi matematika, ini menjadikan MobileNetV2 adalah model yang jauh lebih ringan ketimbang arsitektur CNN konvensional.
3. **Transfer Learning**
   MobileNetV2 juga mendukung teknik transfer learning yang membuat kita tidak perlu melakukan feature extraction secara manual karena model sudah belajar dan mempunyai bobot yang dilatih dari ImageNet. Hal ini membuat proses training jadi lebih cepat dan akurasi model yang didapat sangat tinggi.

3.2. Skenario Pelatihan

Kami menggunakan platform Visual Studio Code. Dari segi transfer learning, kami melakukan freezing pada lapisan awal MobileNetV2 agar bobot yang telah dilatih pada ImageNet tetap terjaga. Kami melakukan penambahan layer custom dibagian akhir untuk

menyesuaikan tema dan dataset yang kami gunakan. Berikut adalah konfigurasi parameter dan hyperparameter yang kami gunakan:

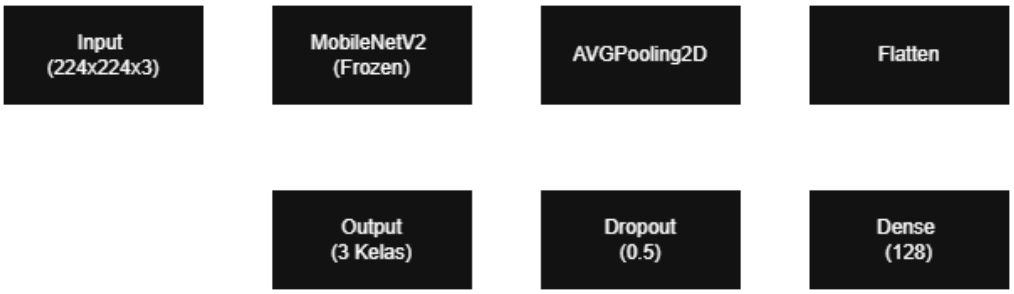
1. **Optimizer**
   Kami menggunakan optimizer Adam dengan learning rate sebesar $1x10^{-4}$ karena learning rate yang relatif kecil ini bertujuan agar perubahan weight terjadi secara halus, mengingat kita menggunakan transfer learning.
2. **Batch Size**
   Kami menggunakan batch size 32 selama pelatihan untuk menyeimbangkan stabilitas gradien dan efisiensi GPU.
3. **Epochs**
   Model dilatih dengan 5 epoch dan sudah menunjukan hasil yang sangat baik tanpa adanya indikasi overfitting ataupun underfitting.
4. **Input Shape**
   Semua data citra yang kami gunakan dan masukkan memiliki resolusi 224x224x3 agar sesuai dengan persyaratan input dari model MobileNetV2.

3.3. Diagram Arsitektur

Kami melakukan sedikit modifikasi seperti menghapus lapisan atas (top layers) dan digantikan dengan lapisan baru yang kami custom sendiri agar menyesuaikan dengan tema dan dataset yang kami pakai. Berikut adalah layers yang kami tambahkan:

1. **Average Pooling 2D**
   Kami mengurangi dimensi spasial dari fitur yang dihasilkan oleh base model dengan ukuran pooling 7x7.
2. **Flatten**
   Kami juga mengubah fitur multidimensi menjadi satu dimensi (vektor).
3. **Dense**
   Kami menambahkan 128 lapisan fully connected dengan aktivasi relu untuk mempelajari pola klasifikasi yang lebih kompleks.
4. **DropOut**
   Kami juga menambahkan teknik regularisasi DropOut sebesar 0.5 untuk mencegah terjadinya overfitting dengan cara menonaktifkan 50% neuron secara acak selama proses training berlangsung
5. **Output Layer**
   Kami men-set lapisan terakhir dengan aktivasi softmax untuk menghasilkan 3 kategori prediksi, yaitu mask, with out mask, mask weared incorrectly.

Setelah memodifikasi base model, kami juga melakukan freezing layers pada semua lapisan base model MobileNetV2 dengan tujuan agar bobot (weight) yang sudah pintar tidak mengalami kerusakan dalam mengenali objek umum. Sehingga proses training akan fokus pada penyesuaian bobot dilapisan yang kami tambahkan secara custom tersebut.
Dikarenakan visualisasi dari output “model.summary()” kami yang terlalu panjang, jadi kami menggantinya dengan visualisasi simpel seperti gambar dibawah ini:



Visualisasi Model

IV. Evaluation (Evaluasi)

4.1. Metrik Evaluasi

```
Epoch 5/5
225/225 [==============================] - 115s 511ms/step - loss: 0.1627 - accuracy: 0.9449 - val_loss: 0.1371 - val_accuracy: 0.9532
```

Acc & Loss Indicator

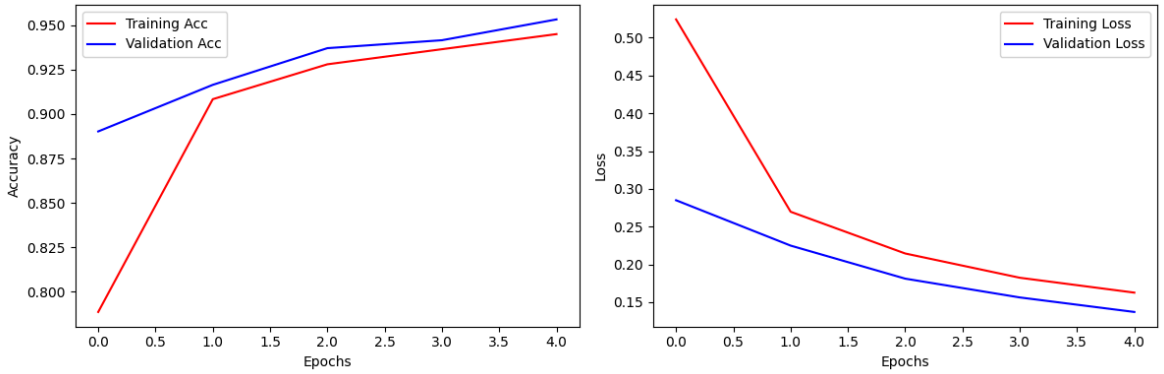
Kami mendapatkan akurasi yang sangat tinggi, yaitu di angka 94% dengan validation accuracy di 95%, karena perbedaan selisih yang sangat sedikit yaitu sekitar 1%, ini membuktikan bahwa model kami tidak mengalami overfitting.

4.2. Comparison (Perbandingan)

Berikut adalah hasil analisis perbandingan kami:

1. **Augmentasi**
   Pada dasarnya, Model hanya akan menghafal posisi wajah yang tegak lurus sesuai dengan posisi wajah yang ada di dataset. dengan augmentasi seperti rotation, zoom, dan horizontal flip. Model tetap mampu mendeteksi wajah didepan kamera walaupun posisi wajah tidak tegak lurus dengan kamera, terlalu jauh dengan kamera, dan terlalu dekat dengan kamera. Hal ini dibuktikan dari nilai validation loss yang terus menurun seiring dengan akurasi yang terus naik.
2. **Transfer Learning**
   Berkat teknik transfer learning, kami jadi tidak perlu melakukan feature extraction manual dari awal yang dimana itu akan membutuhkan waktu yang lumayan lama ditambah resiko adanya underfitting yang tinggi. Penggunaan transfer learning terbukti mampu menciptakan model MobileNetV2 dengan akurasi yang sangat tinggi yaitu di 94% hanya dengan 5 epochs saja.

4.3. Visualisasi Hasil



Acc & Loss Plot

Kami berhasil menghasilkan model yang sangat baik selama proses training. Seperti yang dapat dilihat pada plot diatas, hasilnya menunjukan bahwa model kami mampu belajar dengan sangat baik tanpa adanya indikasi overfitting ataupun underfitting.

V. Deployment (Implementasi)

5.1. Aplikasi

Kami menggunakan framework streamlit untuk mendeploy model yang sudah kami buat dan kami latih. Alasan kami menggunakan streamlit adalah karna streamlit bisa deploy secara lokal maupun global, cepat dan interaktif. Dengan menggunakan streamlit, kami juga tidak perlu harus membuat website atau mengerjakan web development (HTML, CSS, dan JS). Berikut adalah fitur antarmuka yang kami buat dan kembangkan:

1. **Upload Images**
    - User dapat mengupload image (JPG, JPEG, dan PNG) dari mana saja (galeri, internet, dll).
    - Model kami akan secara otomatis memprediksi gambar yang diupload oleh user dengan menambahkan border berwarna menyesuaikan hasil prediksi, label prediksinya, dan confidence score.
2. **Real Time Live Webcam**
    - User bisa menggunakan camera webcam pada device untuk bisa melakukan pemindaian wajah mereka secara real time
    - Model akan memproses setiap frame vidio untuk menangkap wajah user lalu diklasifikasi dengan model MobileNetV2 kita.

5.2. Alur Kerja:

Berikut ini adalah semua proses yang terjadi dibalik interface Streamlit kami:

1. **Input Handling**

   Model akan menerima masukan images berupa citra (RGB) jika user menggunakan fitur upload image. Sementara jika user menggunakan mode real time, maka model akan menerima setiap frame vidio.
2. **Face Detection**

Pada tahap ini, sistem tidak melakukan proses deteksi wajah secara eksplisit. Citra hasil upload maupun setiap frame video dari mode real-time diproses secara langsung sebagai masukan ke dalam model klasifikasi.

3. **Preprocessing & Prediction**

   Citra wajah yang sudah dipotong akan diubah ukuranya menjadi 224x224 pixel, lalu di normalisasi, dan baru bisa dimasukan ke dalam model.

4. **Visual Feedback**

   Setelah semuanya selesai, model kami akan menampilkan kembali images atau vidio tersebut ke user dengan tambahan border yang berwarna mengikuti hasil prediksinya, confidence, dan hasil predik.

VI. Reflection (Refleksi)

6.1. Analisis Kritis

Setelah dilakukanya evaluasi secara mendalam, kami menemukan beberapa temuan setelah kami mengembangkan dan menyelesaikan projek ini. Berikut adalah detailnya:

1. **Kesesuaian Model MobileNetV2**

   Model yang kami gunakan sebenarnya sudah sangat tepat untuk tujuan deployment secara real time maupun tidak real time. Meskipun model-model seperti ResNet atau VGG16 memiliki lapisan yang jauh lebih dalam, tapi model MobileNetV2 sudah mampu menghasilkan akurasi yang sudah sangat tinggi yaitu sekitar 94%.

2. **Keterbatasan Sistem**

   Disinilah letak masalah kami. Kami menemukan beberapa kendala teknis yang mempengaruhi performa deteksi:

   1. **Kesulitan Mempertahankan Hasil Klasifikasi Ketika Real Time**

      Model masih cukup kesulitan dalam mempertahankan hasil klasifikasi ketika mengaktifkan menggunakan real time detector, meskipun memiliki confidence score atau akurasi yang sangat tinggi. Model juga masih sering berganti-ganti klasifikasi jika user bergerak sedikit. Posisi wajah (seperti danga keatas atau menunduk atau tegak lurus) juga sangat menentukan hasil klasifikasi model.

3. **Potensi Pengembangan**

   Untuk meningkatkan hasil yang jauh lebih baik lagi, kami mempunyai beberapa solusi yang dapat kami pertimbangkan untuk membuat model kami jadi jauh lebih baik lagi. Dengan melakukan hyperparameter tuning misal pada learning rate dan dropout, dan mungkin menambah jumlah epoch, kami percaya model akan menjadi lebih baik lagi.

6.2. Kendala & Pembelajaran

Selama pengerjaan proyek ini, ditemukan beberapa hambatan teknis dalam pembelajaran implementasi Deep Learning. Berikut adalah ringkasan kendala dan solusi:

1. **Pembelajaran Utama**

   Pembelajaran terpenting dari proyek ini adalah bahwa akurasi model yang sangat tinggi selama tahap pelatihan (94**%**) tidak serta-merta menjamin performa yang baik

pada tahap *deployment*. Keberhasilan sistem classification secara *real-time* sangat bergantung pada kekuatan algoritma object detector (seperti detektor wajah) yang digunakan untuk menyuplai data ke model utama. Tanpa deteksi wajah yang stabil, model *Deep Learning* tercanggih sekalipun tidak dapat memberikan hasil yang maksimal di lingkungan produksi.

2. **Kendala versi library**

   Salah satu kendala yang dihadapi dalam pengembangan dan deployment aplikasi Face Mask Detection adalah ketidaksesuaian versi antar library yang digunakan, khususnya antara TensorFlow, NumPy, dan OpenCV. TensorFlow memiliki ketergantungan versi yang cukup ketat terhadap NumPy, di mana versi TensorFlow tertentu hanya kompatibel dengan rentang versi NumPy tertentu. Di sisi lain, OpenCV cenderung menarik versi NumPy yang lebih baru, sehingga menyebabkan konflik dependensi yang berujung pada error atau crash saat aplikasi dijalankan di Streamlit. Permasalahan ini berdampak langsung pada stabilitas sistem dan menghambat proses pengujian serta deployment aplikasi. Untuk mengatasi kendala tersebut, dilakukan penyesuaian versi library secara manual melalui pengaturan file requirements, dengan memilih versi NumPy yang masih kompatibel dengan TensorFlow namun tetap dapat berjalan dengan OpenCV, sehingga aplikasi dapat dijalankan secara stabil di lingkungan lokal.

VII. References (Referensi)

[1] S. Mohan, A. Kumar, and A. Kushwaha, “Face Mask Detection using Deep Learning and Computer Vision,” *International Journal of Engineering Research & Technology*, vol. 10, no. 12, Jan. 2022, doi: 10.17577/IJERTV10IS120206.

[2] I. M. D. P. Asana, G. A. Pradana, I. P. S. Handika, and S. I. Murpratiwi, “Mask Detection System Using Convolutional Neural Network Method on Surveillance Camera,” *Telematika: Jurnal Telematika dan Teknologi Informasi*, vol. 19, no. 2, pp. 201–214, June 2022, doi: 10.31315/TELEMATIKA.V19I2.7246.

[3] A. Younesi, R. Afrouzian, and Y. Seyfari, “A transfer learning approach with convolutional neural network for Face Mask Detection,” Oct. 2023, doi: 10.22034/JASP.2022.48447.1167.

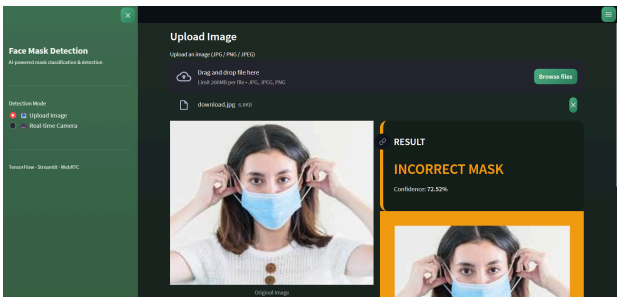
[4] R. Chinnaiyan, I. M, A. R. S. A, K. Sai, M. B. M, and P. Bharath, “Deep Learning based CNN Model for Classification and Detection of Individuals Wearing Face Mask,” Nov. 2023, Accessed: Dec. 28, 2025. [Online]. Available: https://arxiv.org/pdf/2311.10408

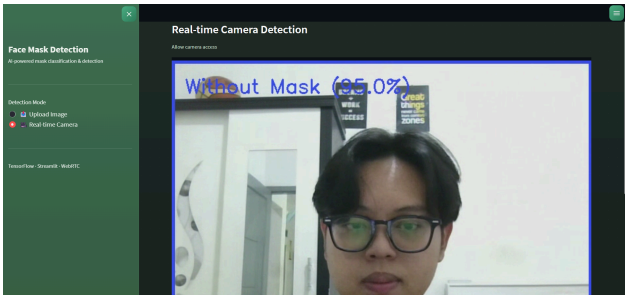
[5] S. F. Abbas, S. H. Shaker, and Firas. A. Abdullatif, “Face Mask Detection Based on Deep Learning: A Review,” *Journal of Soft Computing and Computer Applications*, vol. 1, no. 1, p. 7, June 2024, doi: 10.70403/3008-1084.1006.

[6] J. N. Rathod, V. R. Andodariya, and S. R. Garasia, “Deep Learning Approaches for Face Mask Detection: A Comprehensive Review,” *Journal of Computational Analysis and Applications (JoCAAA)*, vol. 33, no. 05, pp. 1911–1915, May 2024, doi: 10.2139/SSRN.4883405.

VIII. LAMPIRAN

***Screenshot* Aplikasi**





Link Repository (*Source Code*)

https://github.com/Nickskra/FaceMaskDetection.git

Link Deploy

https://facemaskdetection-dl.streamlit.app/

Link Demo Video

Face Mask Detection Demo

Link Video Presentation

AOL DL - Group 6.mp4

Link Power Point

https://www.canva.com/design/DAG7frFknZc/ms_IHoGl7f_ly5SlBRgwzg/edit?utm_content=DAG7frFknZc&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton