

# **Laporan Proyek Deteksi Masker Wajah Menggunakan ResNet18**

442023611073

Adam Toyib Nur Wahid

Tanggal: 27 Mei 2025

## **Abstrak**

Proyek ini bertujuan untuk mengembangkan model klasifikasi gambar yang mampu mendeteksi keberadaan masker pada wajah seseorang dalam sebuah gambar. Model yang digunakan adalah ResNet18, sebuah arsitektur Convolutional Neural Network (CNN) yang telah dilatih sebelumnya pada dataset ImageNet, dan kemudian di-fine-tune pada dataset deteksi masker wajah. Proses pengembangan meliputi pra-pemrosesan data, pelatihan model, evaluasi kinerja, dan implementasi fungsi prediksi untuk gambar baru. Hasil pelatihan menunjukkan akurasi yang sangat tinggi, mencapai 99.92% pada data uji setelah 10 epoch.

## **1. Pendahuluan**

### **1.1. Latar Belakang**

Penggunaan masker wajah telah menjadi praktik umum, terutama dalam konteks kesehatan masyarakat untuk mencegah penyebaran penyakit menular melalui udara. Dengan meningkatnya kesadaran akan pentingnya masker, teknologi untuk mendeteksi penggunaan masker secara otomatis menjadi relevan dalam berbagai skenario, seperti di tempat umum, fasilitas kesehatan, atau area kerja.

### **1.2. Tujuan Proyek**

Tujuan utama dari proyek ini adalah:

Mengimplementasikan dan melatih model deep learning untuk klasifikasi gambar masker wajah.

Menggunakan arsitektur ResNet18 dengan transfer learning untuk mempercepat proses pelatihan dan meningkatkan kinerja.

Mengevaluasi kinerja model yang telah dilatih berdasarkan metrik akurasi dan loss.

Menyediakan fungsionalitas untuk melakukan prediksi pada gambar baru.

## **2. Metodologi**

### **2.1. Dataset**

Dataset yang digunakan dalam proyek ini adalah "Face Mask Detection Dataset" yang diperoleh dari Kaggle (/kaggle/input/face-mask-detection/Dataset). Dataset ini terdiri dari gambar-gambar orang dengan dan tanpa masker wajah, yang dibagi ke dalam dua direktori kelas: with\_mask dan without\_mask. Gambar-gambar dalam dataset memiliki format .jpg, .jpeg, atau .png. Sebuah kelas kustom FaceMaskDataset ditulis untuk memuat gambar dan label dari direktori tersebut.

### **2.2. Pra-pemrosesan Data**

Sebelum dimasukkan ke dalam model, setiap gambar melalui serangkaian tahap pra-pemrosesan yang didefinisikan menggunakan torchvision.transforms:

Resize: Ukuran setiap gambar diubah menjadi 224x224 piksel agar sesuai dengan input yang diharapkan oleh model ResNet18.

ToTensor: Gambar dikonversi dari format PIL Image menjadi format Tensor PyTorch.

Normalize: Tensor gambar dinormalisasi menggunakan nilai mean [0.485, 0.456, 0.406] dan standar deviasi [0.229, 0.224, 0.225], yang merupakan nilai standar untuk dataset ImageNet.

Dataset kemudian dibagi secara acak menjadi data latih (80%) dan data uji (20%) menggunakan fungsi random\_split dari PyTorch. DataLoader digunakan untuk membuat batch data dengan ukuran batch 32 untuk proses pelatihan dan pengujian, dengan pengacakan (shuffle) data latih.

### **2.3. Arsitektur Model**

Model yang dipilih untuk tugas klasifikasi ini adalah ResNet18.

Transfer Learning: Model diinisialisasi dengan bobot yang telah dilatih sebelumnya (pre-trained) pada dataset ImageNet (ResNet18\_Weights.DEFAULT).

Modifikasi Lapisan Akhir: Lapisan fully connected (classifier) terakhir dari ResNet18 (model.fc) diganti dengan sebuah lapisan nn.Linear baru dengan 2 fitur output, yang merepresentasikan dua kelas target: 'with\_mask' dan 'without\_mask'.

Parameter Model: Total parameter model adalah 11.18 juta, dan semuanya dapat dilatih (trainable).

### **2.4. Proses Pelatihan**

Pelatihan model dilakukan dengan konfigurasi sebagai berikut:

Device: Pelatihan dijalankan pada GPU (CUDA) jika tersedia; jika tidak, CPU digunakan.

Loss Function: Fungsi kerugian yang digunakan adalah nn.CrossEntropyLoss.

Optimizer: Optimizer yang dipilih adalah Adam dengan laju pembelajaran (learning rate) sebesar  $1e-4$ .

Jumlah Epoch: Model dilatih selama 10 epoch.

Pada setiap epoch, model dilatih pada data latih, dan kinerjanya dievaluasi pada data uji.

Kerugian (loss) rata-rata untuk data latih dan data uji, serta akurasi pada data uji, dihitung dan dicatat.

### **2.5. Penyimpanan Model**

Setelah proses pelatihan selesai, checkpoint model disimpan dalam file bernama `face_mask_checkpoint.pth`. Checkpoint ini mencakup nomor epoch terakhir, state dictionary dari model dan optimizer, serta riwayat nilai kerugian data latih dan data uji.

### **3. Hasil dan Diskusi**

#### **3.1. Kinerja Pelatihan**

Proses pelatihan model selama 10 epoch menunjukkan kinerja yang sangat baik. Berikut adalah ringkasan kinerja pada epoch terakhir:

Epoch 10:

Train Loss: 0.0003

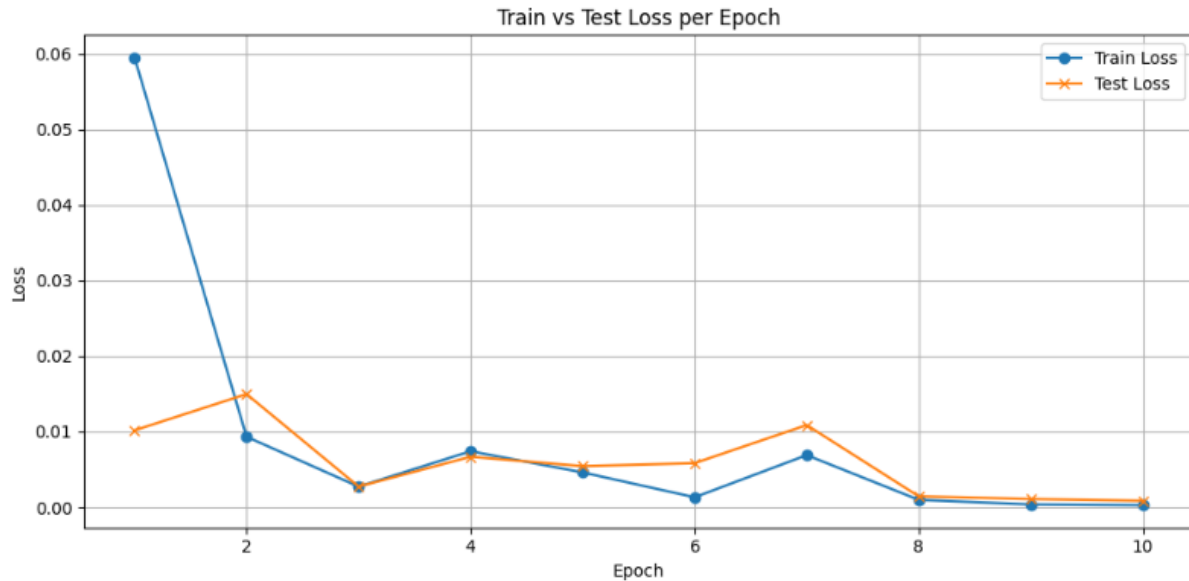
Test Loss: 0.0009

Test Accuracy: 0.9992

Model mencapai akurasi pengujian sebesar 99.92% pada epoch ke-10, dengan nilai loss yang sangat rendah baik pada data latih maupun data uji.

#### **3.2. Visualisasi Kerugian (Loss)**

Sebuah plot yang membandingkan kerugian (loss) pada data latih dan data uji per epoch disertakan dalam notebook.



)

Kurva loss menunjukkan bahwa baik training loss maupun test loss menurun secara signifikan dan konvergen, yang menandakan tidak adanya overfitting yang parah.

### 3.3. Diskusi

Akurasi yang sangat tinggi (99.92%) menunjukkan bahwa arsitektur ResNet18 dengan transfer learning sangat efektif untuk tugas deteksi masker wajah ini. Proses fine-tuning pada lapisan terakhir memungkinkan model untuk beradaptasi dengan baik pada dataset spesifik.

## 4. Prediksi pada Gambar Baru

Notebook menyediakan fungsionalitas untuk memuat model yang telah dilatih dari file checkpoint dan menggunakannya untuk melakukan prediksi pada gambar individual.

#### 4.1. Pemuatan Model

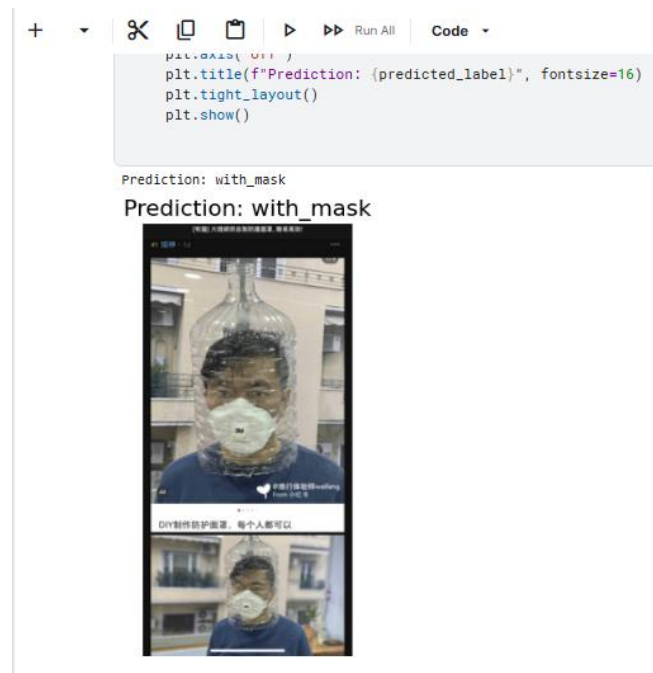
Fungsi `load_model(weight_path, num_classes, device)` bertanggung jawab untuk memuat checkpoint, menginisialisasi arsitektur ResNet18, mengatur lapisan fc, memuat `model_state_dict`, dan mengirim model ke device serta mengaturnya ke mode evaluasi (`model.eval()`).

#### 4.2. Fungsi Prediksi

Fungsi `predict_image(image_path, model, device, class_names)` melakukan pra-pemrosesan pada gambar input, melakukan inferensi, menentukan kelas yang diprediksi, dan mengembalikan nama kelas serta gambar PIL asli.

#### 4.3. Contoh Prediksi

Sebuah contoh prediksi dijalankan pada gambar yang terletak di `/kaggle/input/d/andrewmvd/face-mask-detection/images/maksskskss15.png`. Hasil prediksi dan gambar ditampilkan menggunakan Matplotlib.



## **5. Kesimpulan**

Proyek ini berhasil mendemonstrasikan implementasi model deep learning ResNet18 untuk deteksi masker wajah dengan akurasi 99.92%. Penggunaan transfer learning terbukti efektif. Model yang dilatih dapat disimpan dan digunakan untuk prediksi pada gambar-gambar baru.