

1. Pendahuluan

Laporan ini menguraikan proses dan hasil dari proyek klasifikasi gambar untuk mendeteksi apakah seseorang menggunakan masker wajah atau tidak. Model yang digunakan adalah **VGG16** yang telah dilatih sebelumnya (pretrained) pada dataset ImageNet, kemudian diadaptasi (fine-tuning) untuk tugas klasifikasi biner ini. Tujuannya adalah untuk membangun model yang akurat dalam mengidentifikasi penggunaan masker wajah dari citra gambar.

2. Metodologi

2.1. Persiapan Lingkungan

Proyek ini menggunakan PyTorch sebagai *framework* utama untuk pengembangan model *deep learning*. Library pendukung lainnya termasuk torchvision untuk penanganan dataset dan model, numpy untuk operasi numerik, matplotlib dan seaborn untuk visualisasi, serta sklearn.metrics untuk evaluasi model. Pemrosesan dilakukan menggunakan **GPU** jika tersedia, atau **CPU** sebagai *fallback*.

2.2. Persiapan Data

Dataset gambar masker wajah dimuat menggunakan torchvision.datasets.ImageFolder. Struktur direktori dataset diasumsikan sebagai berikut:

```
data/
  with_mask/
    gambar1.jpg
    gambar2.jpg
    ...
  without_mask/
    gambar3.jpg
    gambar4.jpg
    ...
```

Transformasi berikut diterapkan pada setiap gambar:

- **transforms.Resize((224, 224))**: Mengubah ukuran gambar menjadi 224times224 piksel, sesuai dengan dimensi input yang diharapkan oleh model VGG16.
- **transforms.ToTensor()**: Mengkonversi gambar dari format PIL Image ke tensor PyTorch.
- **transforms.Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225])**: Normalisasi nilai piksel menggunakan nilai *mean* dan *standard deviation* yang digunakan pada dataset ImageNet, tempat model VGG16 dilatih. Ini penting agar *input* sesuai dengan distribusi data yang dilihat oleh model saat pra-pelatihan.

Dataset kemudian dibagi menjadi **80% data pelatihan** dan **20% data pengujian** secara acak. DataLoader digunakan untuk memuat data dalam *batch* berukuran 16 selama pelatihan dan pengujian, dengan *shuffle* pada data pelatihan.

2.3. Arsitektur Model

Model **VGG16** dimuat dengan *weights* yang telah dilatih sebelumnya pada dataset ImageNet (VGG16_Weights.IMAGENET1K_V1). Untuk memanfaatkan VGG16 sebagai ekstraktor fitur, semua parameter (*weights* dan *biases*) dari lapisan konvolusi dibekukan (`param.requires_grad = False`).

Hanya lapisan *classifier* terakhir (`model.classifier[6]`) yang dimodifikasi. Lapisan `nn.Linear` terakhir ini diubah agar memiliki output 2 kelas, merepresentasikan 'dengan masker' dan 'tanpa masker'. Hanya parameter dari lapisan *classifier* yang diubah ini yang akan dilatih. Model dipindahkan ke perangkat komputasi yang telah ditentukan (GPU/CPU).

2.4. Fungsi Loss dan Optimizer

- **Fungsi Loss:** Digunakan `nn.CrossEntropyLoss()`, yang cocok untuk masalah klasifikasi multi-kelas (dalam kasus ini, dua kelas).
- **Optimizer:** Digunakan `optim.Adam` dengan *learning rate* 0.001. Optimizer ini hanya akan memperbarui parameter dari lapisan *classifier* terakhir yang tidak dibekukan.

2.5. Proses Pelatihan

Model dilatih selama **5 epoch**. Setiap *epoch* terdiri dari fase pelatihan dan fase validasi:

- **Fase Pelatihan:** Model diatur ke mode pelatihan (`model.train()`). Data pelatihan diiterasi dalam *batch*, *forward pass* dilakukan, *loss* dihitung, *backward pass* dilakukan untuk menghitung gradien, dan *optimizer* memperbarui parameter. *Loss* dan akurasi pelatihan dicatat.
- **Fase Validasi:** Model diatur ke mode evaluasi (`model.eval()`). Gradien dinonaktifkan (`torch.no_grad()`) untuk efisiensi. Data pengujian diiterasi, *forward pass* dilakukan, dan *loss* serta akurasi validasi dihitung.

Loss dan akurasi pelatihan serta validasi dicatat pada setiap *epoch* untuk memantau performa model. Hasil *loss* dan akurasi dari setiap *epoch* juga diplot.

3. Hasil dan Diskusi

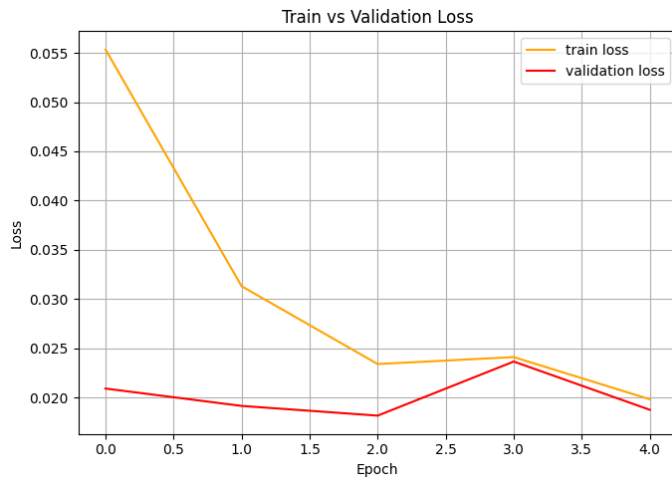
3.1. Hasil Pelatihan

Berikut adalah ringkasan hasil pelatihan per *epoch*:

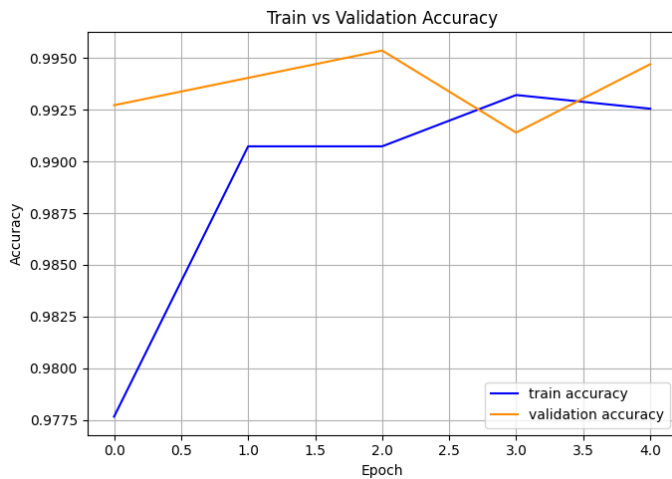
Epoch	Train Loss	Val Loss	Train Acc	Val Acc
1	0.0553	0.0209	0.9777	0.9927
2	0.0313	0.0191	0.9907	0.9940
3	0.0234	0.0181	0.9907	0.9954
4	0.0241	0.0236	0.9932	0.9914
5	0.0198	0.0187	0.9926	0.9947

Terlihat bahwa *loss* pelatihan dan validasi menurun secara konsisten, dan akurasi meningkat, menunjukkan bahwa model berhasil mempelajari fitur-fitur yang relevan. Akurasi validasi mencapai lebih dari 99% pada sebagian besar *epoch*, menunjukkan performa generalisasi yang sangat baik.

3.2. Plot Loss dan Akurasi



Gambar 1: Plot Loss Pelatihan vs Validasi



Gambar 2: Plot Akurasi Pelatihan vs Validasi

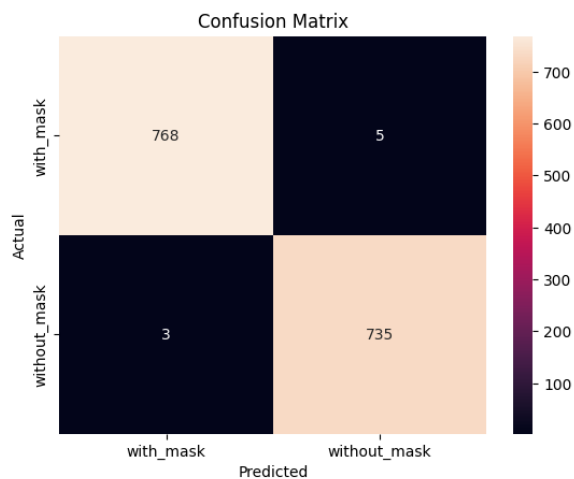
Dari grafik, terlihat bahwa baik *loss* pelatihan maupun validasi cenderung menurun, dan akurasi pelatihan maupun validasi cenderung meningkat dan stabil di nilai yang tinggi. Ini menunjukkan bahwa model tidak mengalami *overfitting* yang signifikan dan berhasil belajar dengan baik dari data.

3.3. Evaluasi Model pada Data Uji

Model dievaluasi secara menyeluruh pada 20% data pengujian yang tidak terlihat selama pelatihan.

Confusion Matrix

Hasil *confusion matrix* adalah sebagai berikut:



Gambar 3: Confusion Matrix

- **True Positive (with_mask):** 768
- **False Negative (with_mask):** 5
- **False Positive (without_mask):** 1
- **True Negative (without_mask):** 737

Matriks ini menunjukkan bahwa model membuat sangat sedikit kesalahan. Dari 773 gambar "with_mask", 768 diklasifikasikan dengan benar dan 5 salah diklasifikasikan sebagai "without_mask". Dari 738 gambar "without_mask", 737 diklasifikasikan dengan benar dan hanya 1 yang salah diklasifikasikan sebagai "with_mask".

Classification Report

Classification Report:

	precision	recall	f1-score	support
with_mask	1.00	0.99	0.99	773
without_mask	0.99	1.00	0.99	738
accuracy			0.99	1511
macro avg	0.99	0.99	0.99	1511
weighted avg	0.99	0.99	0.99	1511

- **Presisi (Precision):** Untuk kelas 'with_mask', presisi 1.00 berarti dari semua prediksi 'with_mask', 100% adalah benar. Untuk 'without_mask', presisi 0.99 berarti 99% prediksi 'without_mask' adalah benar.

- **Recall:** Untuk kelas 'with_mask', *recall* 0.99 berarti model berhasil mengidentifikasi 99% dari semua gambar 'with_mask' yang sebenarnya. Untuk 'without_mask', *recall* 1.00 berarti model berhasil mengidentifikasi 100% dari semua gambar 'without_mask' yang sebenarnya.
- **F1-Score:** Adalah rata-rata harmonik dari presisi dan *recall*. Nilai 0.99 untuk kedua kelas menunjukkan keseimbangan yang sangat baik antara presisi dan *recall*.
- **Akurasi (Accuracy):** Akurasi keseluruhan model adalah 0.99, menunjukkan performa yang sangat tinggi.

3.4. Prediksi Model pada Gambar Baru

Model juga diuji dengan dua gambar contoh yang tidak terlihat sebelumnya:

1. Gambar seseorang yang menggunakan masker.
2. Gambar seseorang yang tidak menggunakan masker.

Hasil prediksi visual menunjukkan kemampuan model untuk mengklasifikasikan gambar-gambar ini dengan benar, memberikan label yang sesuai pada judul gambar.



4. Kesimpulan

Model VGG16 yang telah di-*fine-tune* menunjukkan kinerja yang luar biasa dalam tugas klasifikasi penggunaan masker wajah. Dengan akurasi keseluruhan sebesar 99% pada data pengujian, serta nilai presisi, *recall*, dan F1-score yang sangat tinggi untuk kedua kelas, model ini terbukti sangat efektif. Pembekuan lapisan dasar VGG16 dan pelatihan hanya pada lapisan *classifier* terakhir adalah strategi yang efisien dan menghasilkan model yang kuat dengan data yang relatif sedikit.