



Analisis Komparatif Model Deep Learning untuk Klasifikasi Gambar

Studi Kasus: Face Mask Detection Dataset

Sumber Data :

<https://www.kaggle.com/datasets/omkargurav/face-mask-dataset>

Evaluasi Kinerja Custom CNN, MobileNetV2, dan VGG16

Nama Kelompok :

- Arnold Oktafianto
- Cahyo Anggoro Seto
- Wahyu Pratama

Latar Belakang & Tujuan Penelitian



Tujuan

Penelitian ini bertujuan membandingkan efektivitas model CNN yang dibangun dari awal dengan model Transfer Learning yang sudah mapan. Kami ingin menentukan model terbaik berdasarkan keseimbangan antara akurasi, presisi, recall, dan efisiensi komputasi yang mencakup jumlah parameter dan waktu pelatihan.

Evaluasi komprehensif ini akan memberikan wawasan praktis bagi praktisi machine learning dalam memilih arsitektur yang tepat untuk tugas klasifikasi gambar. Mengembangkan model klasifikasi gambar biner untuk mengidentifikasi dua kelas: `with_mask` dan `without_mask`.

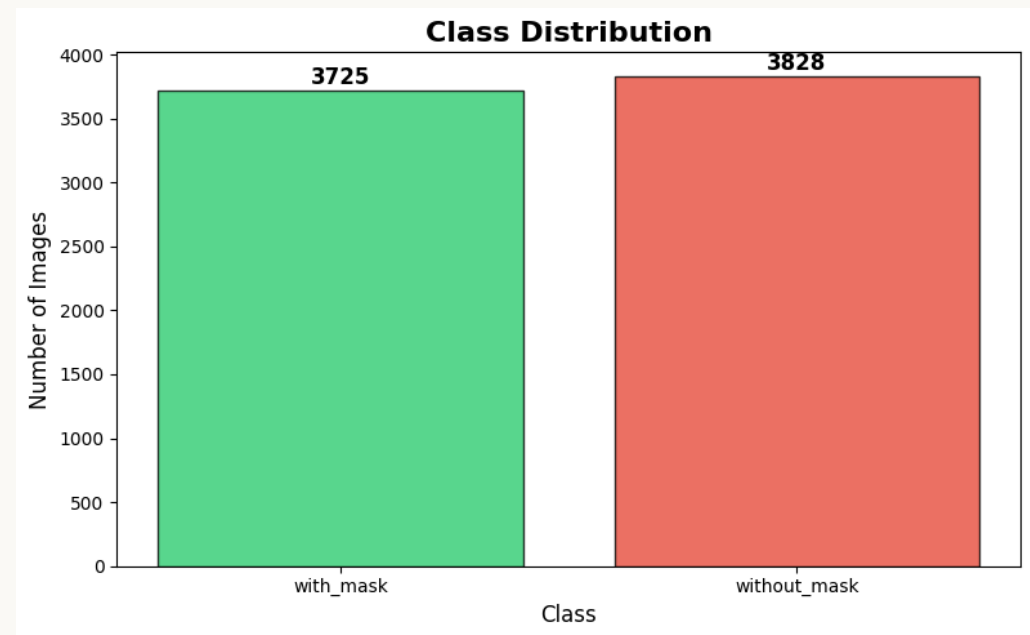


Pendekatan

Membandingkan model Custom CNN dengan teknik Transfer Learning menggunakan MobileNetV2 dan VGG16.

Penelitian ini bertujuan memahami apakah membangun model sendiri (Custom CNN) lebih baik daripada menggunakan model yang sudah dilatih sebelumnya (Transfer Learning) seperti MobileNetV2 yang ringan dan VGG16 yang lebih kompleks.

Tinjauan Dataset



Distribusi Kelas

Dataset relatif seimbang antara kelas with_mask dan without_mask, mencegah bias model dalam proses pembelajaran.

Variasi Data

Gambar mencakup berbagai sudut, pencahayaan, jenis masker, dan latar belakang untuk simulasi kondisi dunia nyata.

6043

Data Training 80%

Total gambar untuk pelatihan model

1510

Data Test 20%

Total gambar untuk pengujian model

7553

Total Dataset

Keseluruhan gambar yang digunakan



Metodologi – Model yang Dibandingkan



Custom CNN

Arsitektur Convolutional Neural Network yang dirancang dan dilatih dari nol khusus untuk dataset ini. Model ini dibangun tanpa pengetahuan sebelumnya.



MobileNetV2

Model Transfer Learning yang dirancang untuk perangkat mobile ringan dan cepat. Menggunakan bobot yang sudah dilatih sebelumnya (pre-trained weights).



VGG16

Model Transfer Learning yang lebih dalam dan "berat", dikenal dengan kemampuan ekstraksi fitur yang kuat. Juga menggunakan pre-trained weights.

Proses Pelatihan Model

```
Training Model: Custom_CNN...
c:\Users\BINLENGKAP\AppData\Local\Programs\Python\Python313\Lib\site-packages\keras\src\trainers\data_adapters\nv_dataset_adapter.py:121: UserWarning: Your `PyDataset` class should call `super().__init__(**kwargs)` in its constructor. `**kwargs` can include `workers`, `use_multiprocessing`, `max_queue_size`. Do not pass these arguments to `init()`, as they will be ignored.
  self._warn_if_super_not_called()
Epoch 1/5
63/189 — 1:02 500ms/step - accuracy: 0.6100 - loss: 0.7162c:\Users\BINLENGKAP\AppData\Local\Programs\Python\Python313\Lib\site-packages\PIL\Image.py:1047: UserWarning: Palette images with Transparency expressed in bytes should be converted to RGBA images
  warnings.warn(
189/189 — 97s 503ms/step - accuracy: 0.7690 - loss: 0.4982 - val_accuracy: 0.8901 - val_loss: 0.2981
Epoch 2/5
189/189 — 92s 488ms/step - accuracy: 0.8234 - loss: 0.4082 - val_accuracy: 0.8954 - val_loss: 0.2959
Epoch 3/5
189/189 — 91s 483ms/step - accuracy: 0.8334 - loss: 0.3937 - val_accuracy: 0.8907 - val_loss: 0.2764
Epoch 4/5
189/189 — 92s 486ms/step - accuracy: 0.8433 - loss: 0.3781 - val_accuracy: 0.8702 - val_loss: 0.2947
Epoch 5/5
189/189 — 92s 484ms/step - accuracy: 0.8440 - loss: 0.3673 - val_accuracy: 0.9166 - val_loss: 0.2412
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This file format is considered legacy. We recommend using instead the native Keras format, e.g. `model.save('my_model.keras')` or `keras.saving.save_model(model, 'my_model.keras')`.
✓ Model final disimpan di: saved_models\Custom_CNN_final.h5
✓ History training disimpan
Evaluating Custom_CNN...
48/48 — 5s 93ms/step - accuracy: 0.9166 - loss: 0.2412
48/48 — 4s 91ms/step
```

Custom CNN

```
Training Model: MobileNetV2...
Epoch 1/5
89/189 — 44s 445ms/step - accuracy: 0.7105 - loss: 0.5623c:\Users\BINLENGKAP\AppData\Local\Programs\Python\Python313\Lib\site-packages\PIL\Image.py:1047: UserWarning: Palette images with Transparency expressed in bytes should be converted to RGBA images
  warnings.warn(
189/189 — 103s 523ms/step - accuracy: 0.8984 - loss: 0.2521 - val_accuracy: 0.9709 - val_loss: 0.0959
Epoch 2/5
189/189 — 97s 513ms/step - accuracy: 0.9677 - loss: 0.0930 - val_accuracy: 0.9768 - val_loss: 0.0710
Epoch 3/5
189/189 — 98s 517ms/step - accuracy: 0.9763 - loss: 0.0718 - val_accuracy: 0.9795 - val_loss: 0.0573
Epoch 4/5
189/189 — 97s 514ms/step - accuracy: 0.9787 - loss: 0.0636 - val_accuracy: 0.9808 - val_loss: 0.0551
Epoch 5/5
189/189 — 98s 518ms/step - accuracy: 0.9803 - loss: 0.0589 - val_accuracy: 0.9801 - val_loss: 0.0511
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This file format is considered legacy. We recommend using instead the native Keras format, e.g. `model.save('my_model.keras')` or `keras.saving.save_model(model, 'my_model.keras')`.
✓ Model final disimpan di: saved_models\MobileNetV2_final.h5
✓ History training disimpan
Evaluating MobileNetV2...
48/48 — 14s 284ms/step - accuracy: 0.9801 - loss: 0.0511
48/48 — 15s 300ms/step
```

MobileNetV2

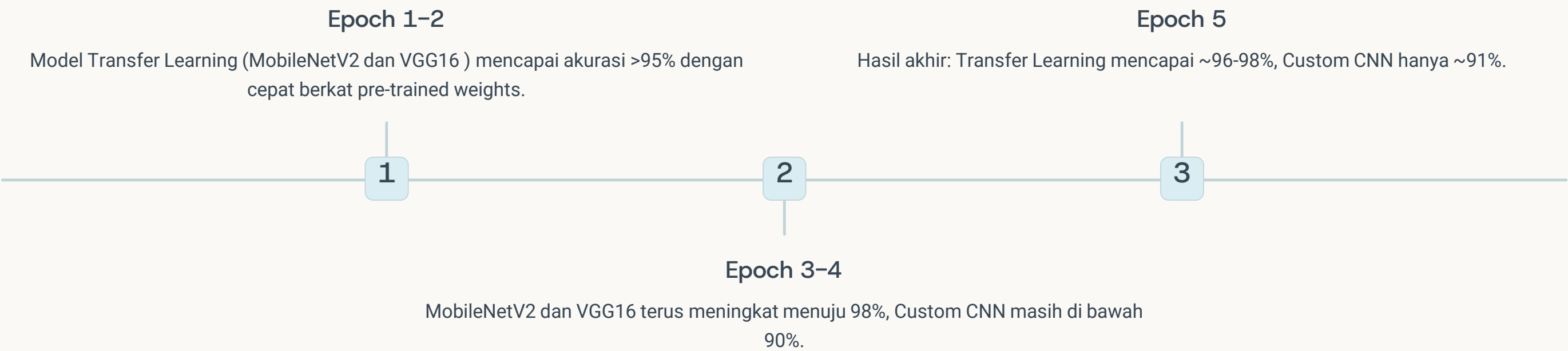
```
Training Model: VGG16...
Epoch 1/5
c:\Users\BINLENGKAP\AppData\Local\Programs\Python\Python313\Lib\site-packages\PIL\Image.py:1047: UserWarning: Palette images with Transparency expressed in bytes should be converted to RGBA images
  warnings.warn(
189/189 — 683s 4s/step - accuracy: 0.8468 - loss: 0.3391 - val_accuracy: 0.9450 - val_loss: 0.1401
Epoch 2/5
189/189 — 682s 4s/step - accuracy: 0.9346 - loss: 0.1744 - val_accuracy: 0.9530 - val_loss: 0.1238
Epoch 3/5
189/189 — 696s 4s/step - accuracy: 0.9467 - loss: 0.1457 - val_accuracy: 0.9742 - val_loss: 0.0713
Epoch 4/5
189/189 — 684s 4s/step - accuracy: 0.9527 - loss: 0.1333 - val_accuracy: 0.9662 - val_loss: 0.0830
Epoch 5/5
189/189 — 1737s 9s/step - accuracy: 0.9570 - loss: 0.1165 - val_accuracy: 0.9669 - val_loss: 0.0832
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This file format is considered legacy. We recommend using instead the native Keras format, e.g. `model.save('my_model.keras')` or `keras.saving.save_model(model, 'my_model.keras')`.
✓ Model final disimpan di: saved_models\VGG16_final.h5
✓ History training disimpan
Evaluating VGG16...
48/48 — 155s 3s/step - accuracy: 0.9669 - loss: 0.0832
48/48 — 160s 3s/step
```

VGG 16



Proses Pelatihan Model

Semua model dilatih selama 5 Epochs dengan kondisi yang sama untuk memastikan perbandingan yang adil.



📌 **Observasi Penting:** Model Transfer Learning (MobileNetV2 & VGG16) mencapai akurasi validasi tinggi sangat cepat, bahkan sejak epoch pertama. Custom CNN membutuhkan waktu lebih lama untuk konvergen dan mencapai akurasi akhir yang lebih rendah dalam jumlah epoch yang sama.

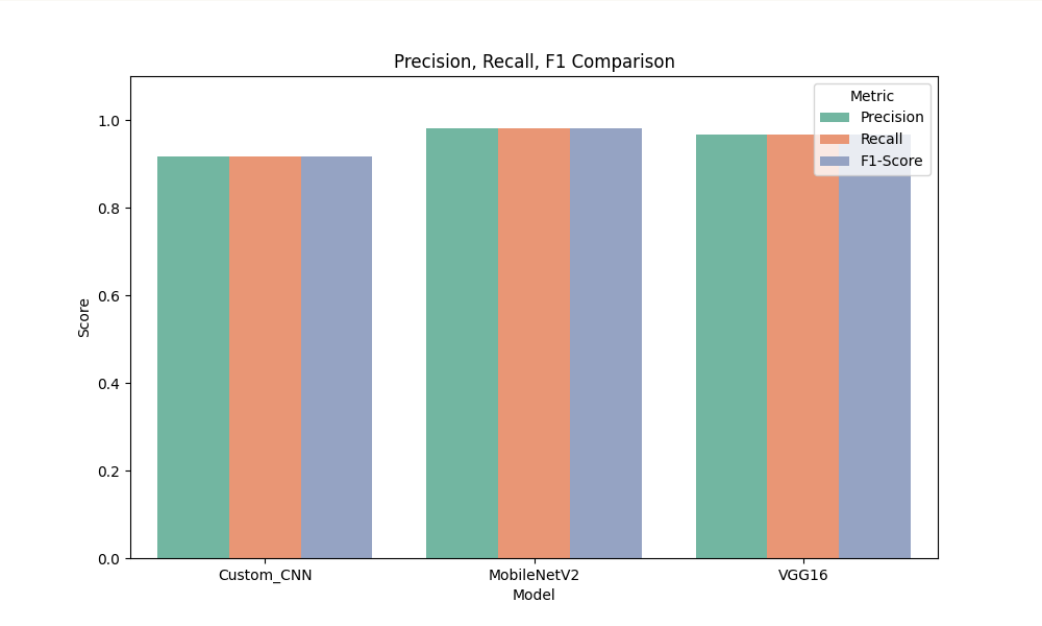
Perbandingan Metrik Kuantitatif

| Model | Accuracy | Precision | Recall | F1-Score | Parameter |
|-------------|----------|-----------|--------|----------|-----------|
| Custom CNN | 91.66% | 91.74% | 91.66% | 91.65% | 23.9M |
| MobileNetV2 | 98.01% | 98.01% | 98.01% | 98.01% | 2.4M |
| VGG16 | 96.69% | 96.77% | 96.69% | 96.69% | 17.9M |

📌 **Analisis Kunci:** MobileNetV2 adalah pemenang mutlak di semua metrik kinerja, mencapai akurasi 98%. Yang paling mengesankan, MobileNetV2 mencapai ini dengan jumlah parameter **10x lebih sedikit** dibandingkan Custom CNN dan VGG16. Ini menunjukkan efisiensi luar biasa dari transfer learning pada arsitektur modern.

Custom CNN, meskipun memiliki parameter terbanyak (23.9M), memiliki performa terendah (92%), menunjukkan bahwa "lebih besar tidak selalu lebih baik" jika arsitektur tidak optimal.

Hasil Evaluasi : Presisi, Recall, F1-Score



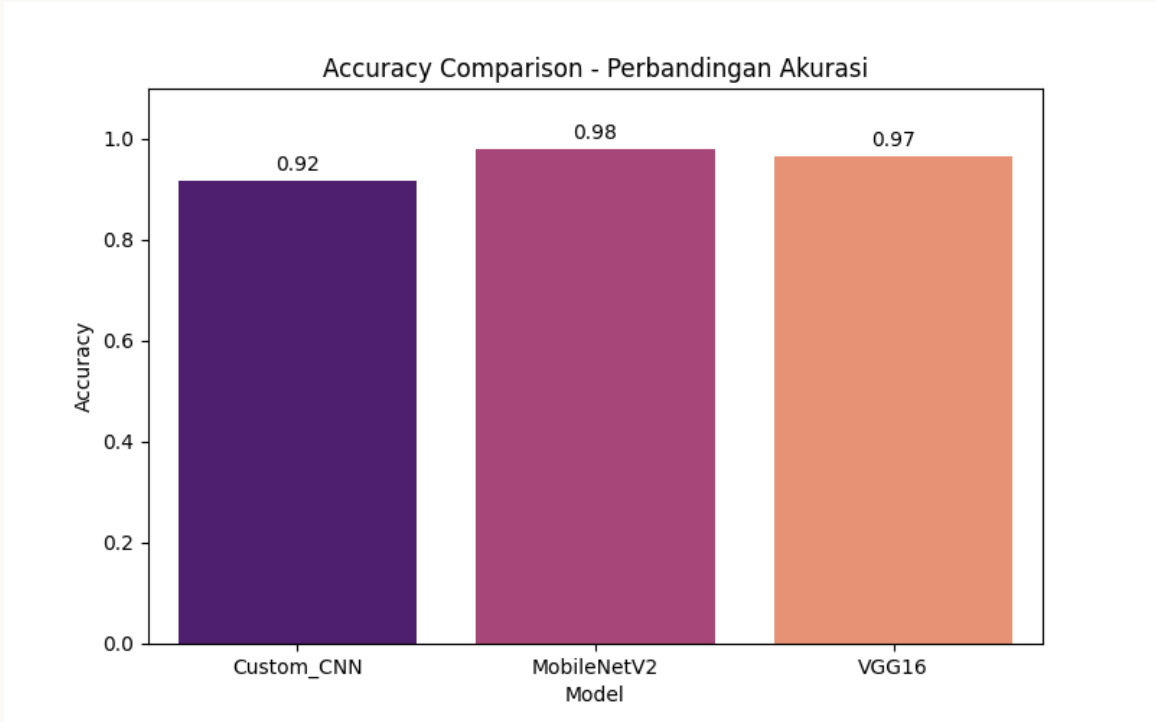
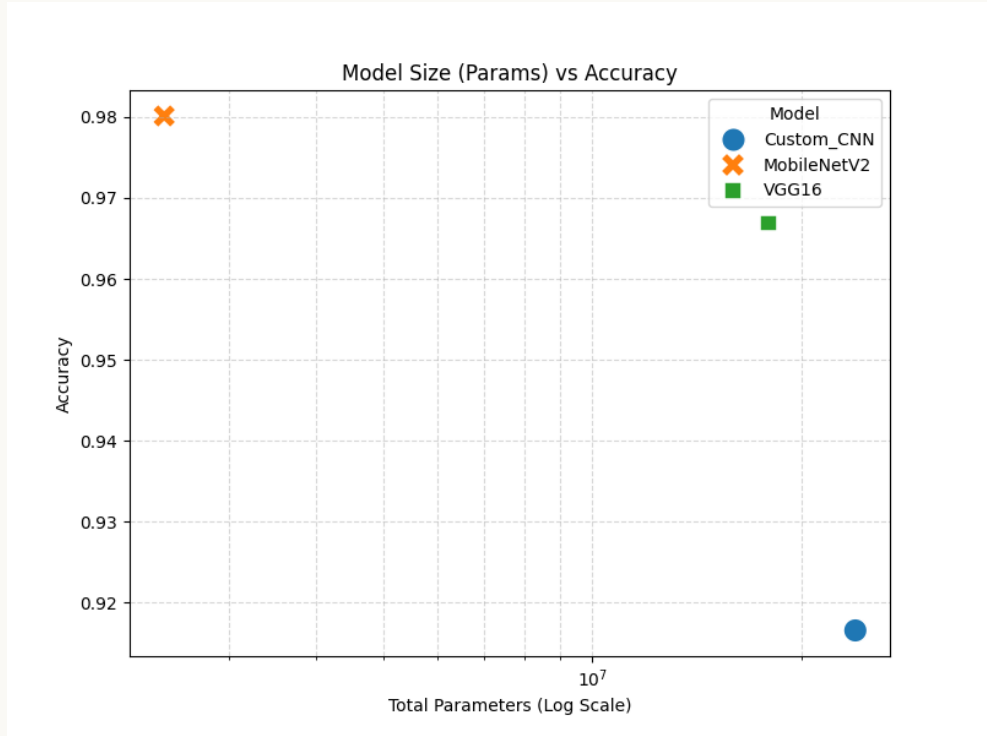
```
--- HASIL AKHIR ---
      Model Accuracy Precision Recall F1-Score Params
0 Custom_CNN 0.916556 0.917452 0.916556 0.916539 23907521
1 MobileNetV2 0.980132 0.980145 0.980132 0.980132 2422081
2 VGG16 0.966887 0.967745 0.966887 0.966882 17926209
D:\CAHYO\KULIAH\COMPUTER VISION\project\deepseek_python_20251203_819d9e.py:299: FutureWarning:
```

Keunggulan Transfer Learning

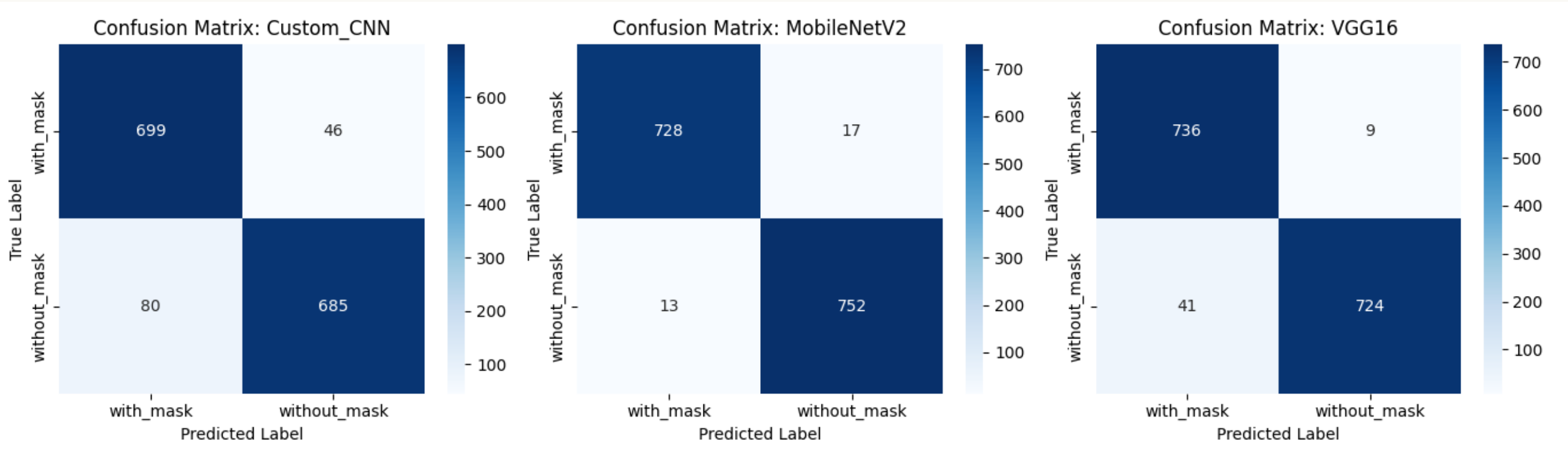
- MobileNetV2 dan VGG16 menunjukkan keseimbangan sangat baik antara Precision dan Recall (>96%)
- Jarang salah mendeteksi masker jika tidak ada (Presisi tinggi)
- Jarang melewatkan orang tanpa masker (Recall tinggi)

Konsistensi Custom CNN

Custom CNN konsisten di angka ~91% untuk semua metrik, menunjukkan performa yang stabil namun lebih rendah dibanding model Transfer Learning.



Hasil Evaluasi – Confusion Matriks



Perbandingan hasil evaluasi confusion matriks



Analisis Kesalahan (Confusion Matrix)

Confusion matrix memberikan gambaran paling jujur tentang di mana model membuat kesalahan dalam prediksi.

| | | |
|--|---|--|
| | | |
| MobileNetV2 | VGG16 | Custom CNN |
| 30 kesalahan total (17 + 13). Diagonal utama sangat tebal, kesalahan minimal. Model paling akurat. | 50 kesalahan total (9 + 41). Cenderung lebih banyak False Negative pada kelas 'without mask'. | 126 kesalahan total (46 + 80). Kesalahan terbanyak, membuktikan keterbatasan model yang dibangun dari nol. |

Insight Penting: Visualisasi confusion matrix secara jelas membuktikan keunggulan MobileNetV2 dengan kesalahan prediksi paling sedikit dibanding dua model lainnya.



Efisiensi Model dan Kesimpulan Teknis

23.9M

Custom CNN

Parameter (Sangat Berat,
Akurasi Terendah)

17.9M

VGG16

Parameter (Berat, Akurasi
Tinggi)

2.4M

MobileNetV2

Parameter (Sangat Ringan,
Akurasi Tertinggi)



Temuan Utama

Jumlah parameter besar tidak menjamin akurasi lebih baik. Custom CNN memiliki parameter terbanyak namun kinerja terburuk.



Efisiensi Luar Biasa

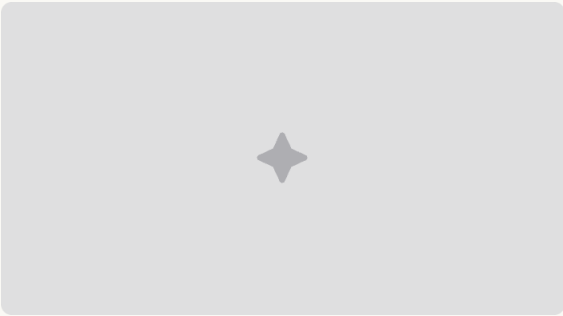
MobileNetV2 hanya 2,4 juta parameter—hampir 10x lebih kecil dari Custom CNN—namun memberikan akurasi tertinggi 98%.



Transfer Learning Efektif

Pendekatan Transfer Learning terbukti sangat efektif untuk dataset deteksi masker wajah ini.

Kesimpulan Akhir



01

Akurasi Tertinggi

Mencapai 98.01%, unggul dibanding VGG16 (96.68%) dan Custom CNN (91.65%)

03

Efisiensi Komputasi

Jauh lebih ringan (2,4M parameter) dibanding model lain, ideal untuk penerapan real-time pada CCTV atau perangkat mobile

Pemenang: MobileNetV2

MobileNetV2 adalah model terbaik untuk tugas deteksi masker wajah berdasarkan evaluasi komprehensif terhadap akurasi, efisiensi, dan stabilitas kinerja.

02

Stabilitas Metrik

Paling stabil di semua metrik evaluasi: Precision, Recall, dan F1-Score konsisten di atas 96%

04

Kesalahan Minimal

Hanya 30 kesalahan prediksi total, terendah dibanding VGG16 (50) dan Custom CNN (126)



Terima Kasih

Sesi Tanya Jawab