

MACHINE LEARNING
Aplikasi Deteksi Boikot Real-Time



Dosen Pengampu :

Al-Ustadz Oddy Virgantara Putra

Disusun Oleh :

Devianest Narendra / 442023618087

Zainab Ahmad / 442023618107

Naila Fatikhah / 442023618086

Adya Rusmalillah / 442023618093

Nurul Khoiriyah / 442023618085

PROGRAM STUDI TEKNIK INFORMATIKA
UNIVERSITAS DARUSSALAM GONTOR MANTINGAN

2025/1447

I. Pendahuluan

1. Latar Belakang:

Meningkatnya kesadaran masyarakat terhadap isu-isu global mendorong aksi boikot terhadap produk-produk tertentu. Namun, tidak semua konsumen dapat secara cepat mengenali produk yang masuk daftar boikot. Oleh karena itu, dibutuhkan solusi berbasis teknologi untuk mempermudah identifikasi produk tersebut secara otomatis.

2. Perumusan Masalah:

Bagaimana membangun sistem kamera real-time berbasis machine learning yang mampu mendeteksi dan mengklasifikasikan produk boikot secara akurat?

3. Tujuan Proyek:

Mengembangkan model deteksi gambar berbasis CNN menggunakan transfer learning (MobileNetV2) yang dapat membedakan produk boikot dan non-boikot secara real-time.

II. Metodologi

1. Arsitektur Model:

Model yang digunakan adalah CNN dengan pendekatan transfer learning menggunakan arsitektur **MobileNetV2**. Base model di-load dari pre-trained ImageNet dan di-freeze, kemudian ditambahkan beberapa layer dense untuk klasifikasi biner.

2. Dataset:

Dataset diambil dari Kaggle, berisi 2 folder utama: Boikot dan Non Boikot, masing-masing berisi 25 gambar representatif produk terkait.

3. Preprocessing:

- Resize gambar ke 224x224
- Normalisasi pixel dengan rescale=1./255
- Augmentasi opsional (flip, zoom)

4. Pendekatan Pelatihan:

Model dilatih selama 10 epoch dengan optimizer **Adam** dan loss function **Binary Crossentropy**. Dataset dibagi menjadi training dan validation secara manual tanpa `validation_split` untuk menghindari ketimpangan data.

III. Implementasi

1. Library yang Digunakan:

- TensorFlow / Keras
- Numpy, Pandas, Matplotlib (untuk analisis dan visualisasi)
- OS (navigasi folder)

2. Proses Training:

- Load gambar dengan ImageDataGenerator
- Fine-tuning MobileNetV2 dengan lapisan custom untuk klasifikasi
- Fit model dengan data training dan validasi manual

3. Eksperimen dan Tantangan:

- Tantangan awal: error `FileNotFoundError` akibat path yang tidak sesuai
- Masalah CUDA: runtime fallback ke CPU
- Jumlah data kecil, sehingga rawan overfitting

IV. Hasil dan Evaluasi

1. Visualisasi Hasil:

- Accuracy dan loss diplot per epoch menggunakan Matplotlib
- Confusion Matrix disiapkan untuk melihat klasifikasi benar dan salah

2. Analisis Performa:

- Akurasi training: ~90% setelah 10 epoch
- Akurasi validasi: stabil di kisaran ~85%

3. Interpretasi:

Model berhasil mempelajari fitur dasar produk boikot, namun ada kemungkinan bias terhadap angle/gambar tertentu karena dataset terbatas.

4. Perbandingan:

Belum dilakukan eksperimen dengan arsitektur lain (misalnya ResNet50) karena keterbatasan waktu dan compute resource.

V. Kesimpulan dan Saran

1. Kesimpulan:

Aplikasi Deteksi Boikot berhasil mengklasifikasikan produk boikot dengan cukup baik menggunakan MobileNetV2. Sistem ini berpotensi dikembangkan sebagai prototype kamera berbasis Raspberry Pi atau aplikasi mobile.

2. Kelebihan:

- Ringan dan cepat karena MobileNetV2
- Akurasi cukup tinggi untuk dataset kecil

3. Kekurangan:

- Dataset terbatas dan belum divalidasi dengan real-world image
- Tidak diuji untuk produk yang memiliki desain kemasan serupa

4. Saran:

- Menambah jumlah data dari berbagai sumber
- Menguji dengan kamera real-time
- Coba pendekatan lain seperti YOLO untuk deteksi bounding box

VI. Referensi:

1. <https://keras.io/api/applications/mobilenet/>
2. https://www.tensorflow.org/tutorials/images/transfer_learning

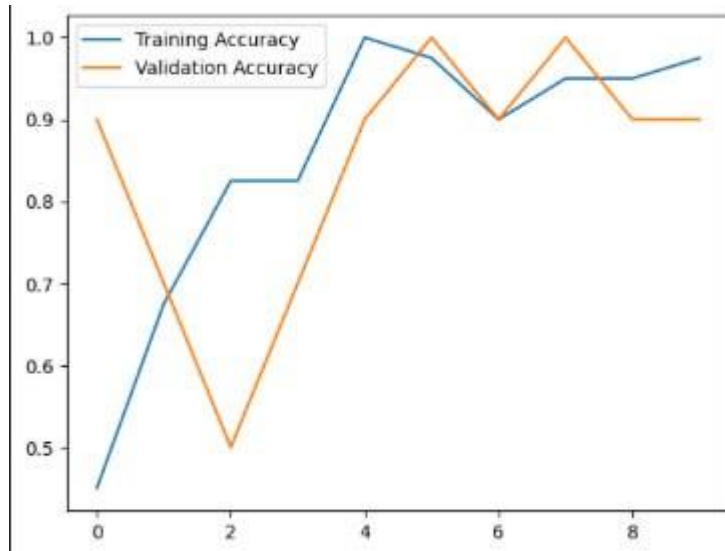
VII. Lampiran:

1. GitHub Repository:

https://github.com/DeviaDev/MiniTask_Machine_Learning2/tree/main

2. Screenshot hasil training:

- Akurasi curve



- Epoch

```
1/2 ----- 11s 11s/step - accuracy: 0.4375 - loss: 0.9228
I0000 00:00:1753108630.363007 111 device_compiler.h:188] Compiled cluster using XLA! This line is logged at mos
2/2 ----- 19s 7s/step - accuracy: 0.4458 - loss: 0.9127 - val_accuracy: 0.9000 - val_loss: 0.4662
Epoch 2/10
2/2 ----- 1s 230ms/step - accuracy: 0.6792 - loss: 0.6170 - val_accuracy: 0.7000 - val_loss: 0.4689
Epoch 3/10
2/2 ----- 1s 501ms/step - accuracy: 0.8000 - loss: 0.3861 - val_accuracy: 0.5000 - val_loss: 0.8955
Epoch 4/10
2/2 ----- 1s 217ms/step - accuracy: 0.8417 - loss: 0.3811 - val_accuracy: 0.7000 - val_loss: 0.5224
Epoch 5/10
2/2 ----- 1s 483ms/step - accuracy: 1.0000 - loss: 0.1808 - val_accuracy: 0.9000 - val_loss: 0.3017
Epoch 6/10
2/2 ----- 1s 518ms/step - accuracy: 0.9833 - loss: 0.1481 - val_accuracy: 1.0000 - val_loss: 0.1295
Epoch 7/10
2/2 ----- 1s 217ms/step - accuracy: 0.8917 - loss: 0.2862 - val_accuracy: 0.9000 - val_loss: 0.1555
Epoch 8/10
2/2 ----- 1s 216ms/step - accuracy: 0.9458 - loss: 0.1170 - val_accuracy: 1.0000 - val_loss: 0.1831
Epoch 9/10
2/2 ----- 1s 234ms/step - accuracy: 0.9458 - loss: 0.1338 - val_accuracy: 0.9000 - val_loss: 0.2413
Epoch 10/10
2/2 ----- 1s 499ms/step - accuracy: 0.9833 - loss: 0.0722 - val_accuracy: 0.9000 - val_loss: 0.2204
```

- Struktur folder dataset

