LAPORAN UAS DEEP LEARNING

IDENTIFIKASI MAKANAN SEHAT MAHASISWA TEKNIK UNIVERSITAS BENGKULU

Disusun Oleh :

1. **Nanditha Nabiilah Putri (G1A021001)**
2. **Erin Handayani Azzahra (G1A021047)**
3. **Farhani Ilham Hidayatullah (G1A021081)**

Dosen Mata Kuliah:

Arie Vatresia, S.T., M.T.I., Ph.D

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS BENGKULU

2024

1. Pendahuluan

Mahasiswa sering kali menghadapi tantangan dalam menjaga pola makan sehat di tengah jadwal akademik yang padat. Hal ini juga dirasakan oleh mahasiswa Teknik Universitas Bengkulu, yang cenderung memilih makanan cepat saji atau makanan yang kurang sehat karena keterbatasan waktu dan pilihan. Padahal, asupan makanan sehat sangat penting untuk mendukung kesehatan fisik dan mental, serta menjaga performa akademik mereka. Dalam konteks ini, teknologi berbasis *deep learning* dapat menjadi solusi inovatif untuk membantu mahasiswa mengenali dan memilih makanan yang sehat. Penelitian ini menggunakan model MobileNetV2, sebuah arsitektur *deep learning* yang ringan dan efisien, untuk mengidentifikasi makanan sehat yang sering dikonsumsi oleh mahasiswa Teknik Universitas Bengkulu. Dengan sistem ini, diharapkan mahasiswa dapat lebih mudah memahami pola makan sehat dan meningkatkan kualitas hidup mereka.

2. Analisa Model

Analisis model dalam penelitian ini berfokus pada penerapan arsitektur MobileNetV2, sebuah *convolutional neural network* (CNN) yang dirancang untuk tugas klasifikasi gambar dengan efisiensi tinggi. MobileNetV2 menggunakan *depthwise separable convolution* untuk mengurangi jumlah parameter dan mempercepat proses komputasi tanpa mengorbankan akurasi. Selain itu, *inverted residual blocks* yang menjadi ciri khas model ini memungkinkan efisiensi propagasi informasi melalui jaringan dengan mempertahankan fitur yang relevan dan menghilangkan fitur yang tidak signifikan. Model ini memanfaatkan *transfer learning* dari model pra-latih pada dataset besar seperti ImageNet, kemudian disesuaikan dengan dataset lokal yang mencakup makanan sehat khas mahasiswa Teknik Universitas Bengkulu. Dalam penelitian ini, model dilatih selama 25.000 epoch untuk memastikan proses optimasi parameter berlangsung secara mendalam, sehingga model mampu mengenali pola kompleks dari citra makanan. MobileNetV2 dievaluasi menggunakan metrik seperti akurasi, *precision*, *recall*, dan *F1-score*, yang menunjukkan bahwa model ini tidak hanya efisien tetapi juga andal dalam mengidentifikasi makanan sehat dengan tingkat akurasi yang tinggi.

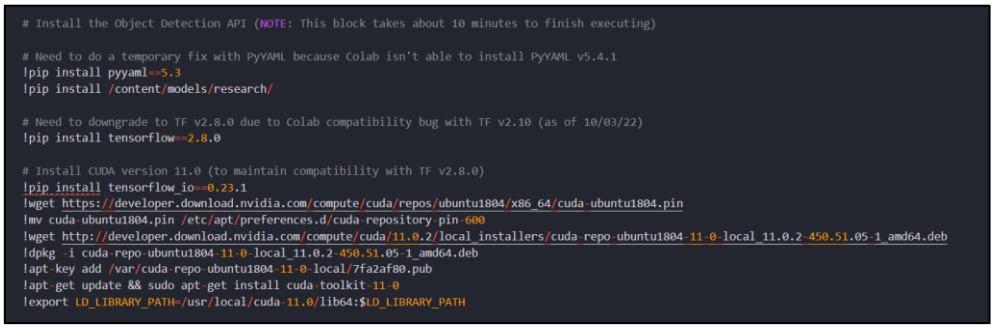
3. Penjelasan Kode

3.1 Pengumpulan Data

Pengumpulan data untuk penelitian ini dilakukan dengan menggabungkan dua sumber utama, yaitu dataset makanan yang tersedia di platform Kaggle dan foto makanan

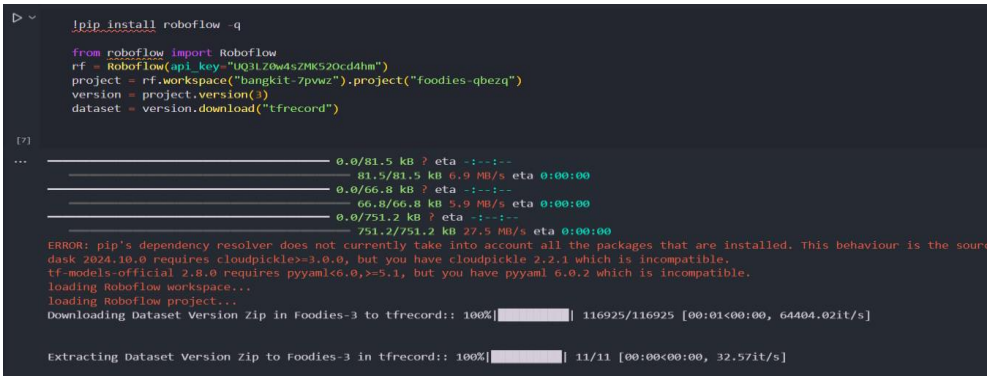
yang diambil secara mandiri. Dataset dari Kaggle menyediakan berbagai gambar makanan yang sudah dilabeli, mencakup berbagai kategori sehat. Data ini digunakan untuk memberikan dasar yang kuat dalam melatih model untuk mengenali berbagai jenis makanan. Selain itu, foto makanan diambil secara langsung oleh peneliti dengan menggunakan kamera secara langsung, yang mencakup makanan yang sering dikonsumsi oleh mahasiswa Teknik Universitas Bengkulu. Foto ini diambil dalam berbagai kondisi pencahayaan dan sudut pandang untuk mencerminkan variasi yang mungkin terjadi dalam dunia nyata. Setelah itu, kedua sumber data ini diproses dan digabungkan menjadi satu dataset terstruktur yang mencakup citra makanan, dan label jenis makanan. Dataset yang dihasilkan digunakan untuk melatih model MobileNetV2 agar dapat mengidentifikasi makanan sehat dengan akurasi tinggi.

3.2 Preprocessing Data



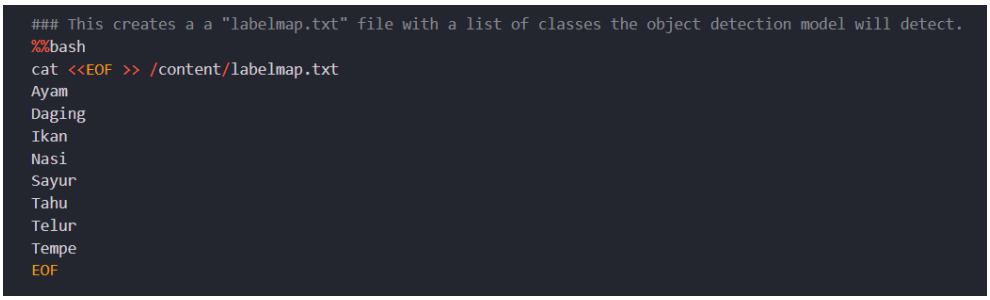
Gambar 3.1 Preprocessing Data

Kode pada gambar digunakan untuk mempersiapkan lingkungan pengembangan guna menjalankan Object Detection API dengan TensorFlow dan akselerasi GPU menggunakan CUDA. Langkah-langkahnya meliputi instalasi pustaka PyYAML versi 5.3 untuk mengatasi masalah kompatibilitas di Google Colab, penurunan versi TensorFlow ke 2.8.0 untuk mendukung kompatibilitas dengan API dan CUDA, serta instalasi TensorFlow I/O versi 0.23.1 untuk mendukung berbagai format data I/O. Selanjutnya, kode menginstal CUDA Toolkit 11.0, termasuk menambahkan repositori CUDA, mendownload dan menginstal file instalasi lokal CUDA, menambahkan kunci repositori, memperbarui sistem, serta menginstal toolkit CUDA. Terakhir, variabel lingkungan `LD_LIBRARY_PATH` diatur untuk memastikan TensorFlow dapat menemukan pustaka CUDA selama runtime, sehingga memungkinkan akselerasi GPU untuk proyek pembelajaran mesin.



Gambar 3.2 Memuat Dan Memproses Dataset

Kode pada gambar 3.2 merupakan kode untuk memproses dataset. Dataset didapat dari roboflow yang sebelumya telah dilabeli sesuai dengan kelas yang ada. Pada identifikasi ini merupakan pembelajaran terarah yaitu data telah belajar dengan dilabeli pada Roboflow. Pada dataset ini terdiri dari 8 kelas.



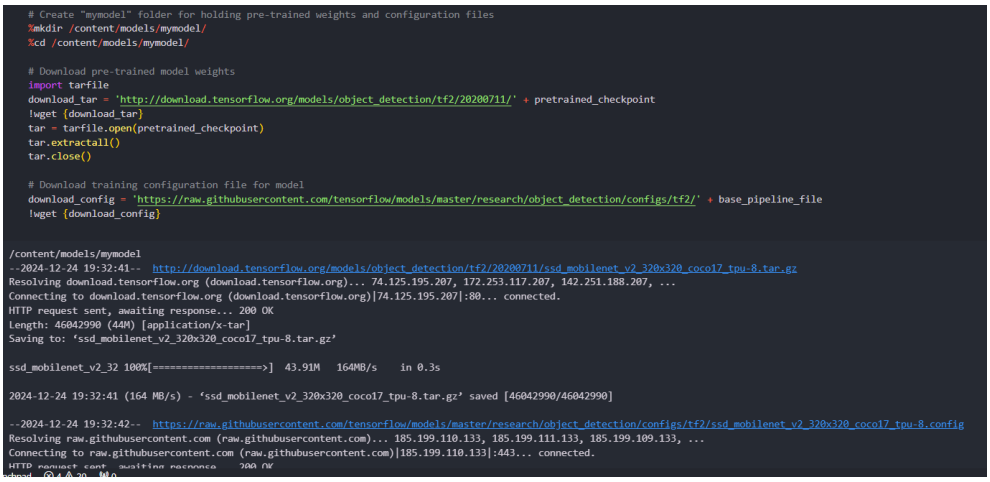
Gambar 3.3 Preprocessing Data

Kode pada gambar 3.3 merupakan perintah untuk labelmap. labelmap berfungsi untuk memetakan atau menghubungkan label numerik ke label tekstual dalam konteks pemrosesan data, seperti dalam klasifikasi gambar atau deteksi objek. Dalam pembelajaran ini terdapat 8 kelas diantaranya adalah ayam, daging, ikan, nasi, sayur, tahu, telur dan tempe.

3.3 Pelatihan Model

Gambar 3. 4 Pelatihan model

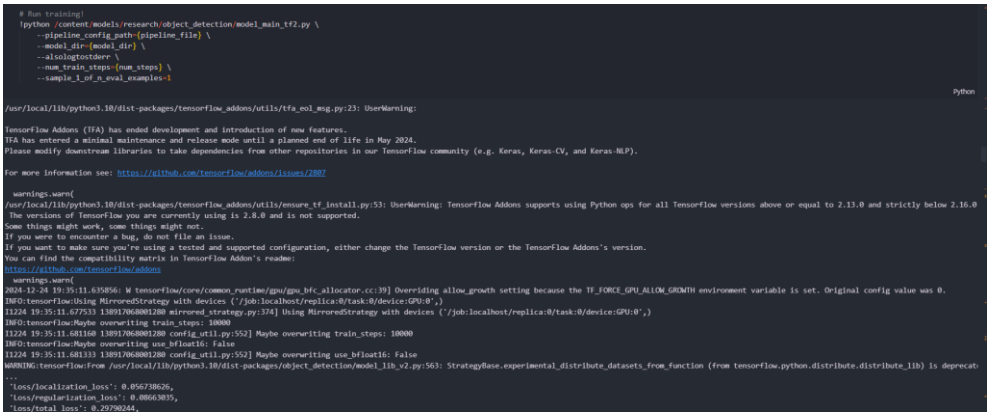
Gambar 3.4 menetapkan model yang akan digunakan (ssd-mobilenet-v2) dan menyediakan konfigurasi untuk berbagai model deteksi objek lainnya. Setiap model memiliki nama model, file konfigurasi pipeline, dan checkpoint pralatih yang terkait. Model yang dipilih kemudian dapat digunakan dalam proses pelatihan atau inferensi sesuai kebutuhan



Gambar 3. 5 pelatihan model

Kode yang ditampilkan pada gambar berfungsi untuk menyiapkan direktori kerja, mengunduh model pra-latih, dan file konfigurasi yang dibutuhkan untuk melatih model deteksi objek. Pertama, folder bernama mymodel dibuat di dalam direktori /content/models/, dan direktori kerja dipindahkan ke folder tersebut.

- Train Model



Gambar 3. 6 train model

Pada gambar 4.3, num_steps adalah jumlah total langkah yang digunakan untuk melatih model. Pada model yang akan dilatih diatur num_steps sebanyak 40000. Semakin banyak langkah maka akan semakin lama model akan dilatih. Proses pelatihan dapat dihentikan di tengah proses pelatihan. Kemudian batch_size adalah jumlah gambar yang digunakan per langkah pelatihan. Pada model ini batch_size yang digunakan adalah 16 dikarenakan pre-trained model yang digunakan adalah SSD MobileNetV2. Batch size 16 bagus untuk model ssd karena

memungkinkan pelatihan yang efisien, memanfaatkan kapasitas memori GPU yang tersedia, dan memberikan stabilitas gradien yang cukup untuk konvergensi yang efektif. Dalam mengatur batch_size dapat disesuaikan berdasarkan GPU yang digunakan dalam pelatihan.

3.4 Evaluasi Model

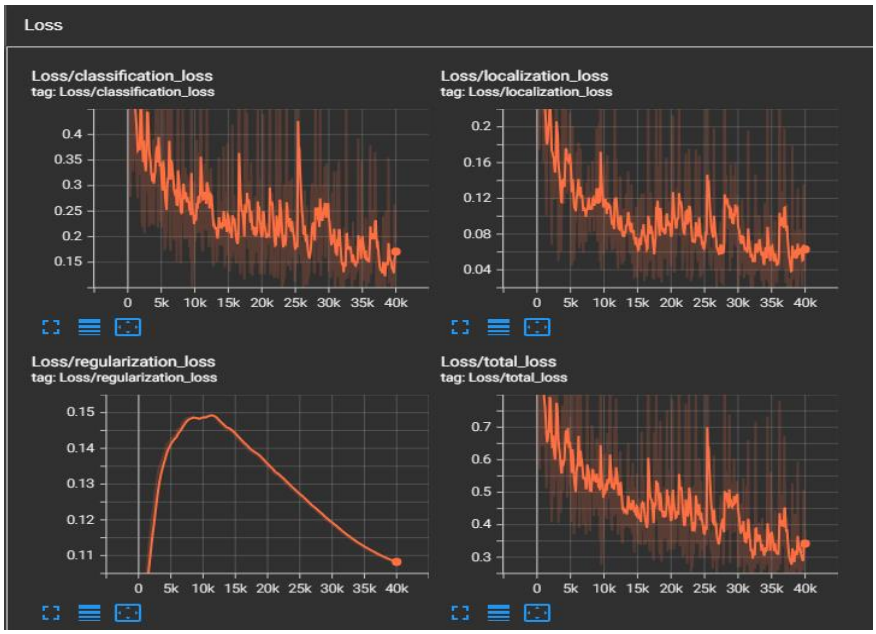
- Akurasi model

```
***mAP Results***

Class           Average mAP @ 0.5:0.95
---------------------------------------
Ayam            87.52%
Daging          80.00%
Ikan            81.38%
Nasi            78.17%
Sayur           68.49%
Tahu            85.27%
Telur           86.62%
Tempe           81.43%

Overall         81.11%
```

Gambar 3. 7 MAP

Pada pembelajaran menggunakan SSD MobileNetV2 menggunakan evaluasi MAP atau Mean Average Precision. mAP memberikan penilaian yang lebih menyeluruh dibandingkan hanya menggunakan akurasi biasa karena mempertimbangkan beberapa hal yaitu Lokasi deteksi (bounding box), Keakuratan klasifikasi, Tingkat ambang batas IoU.

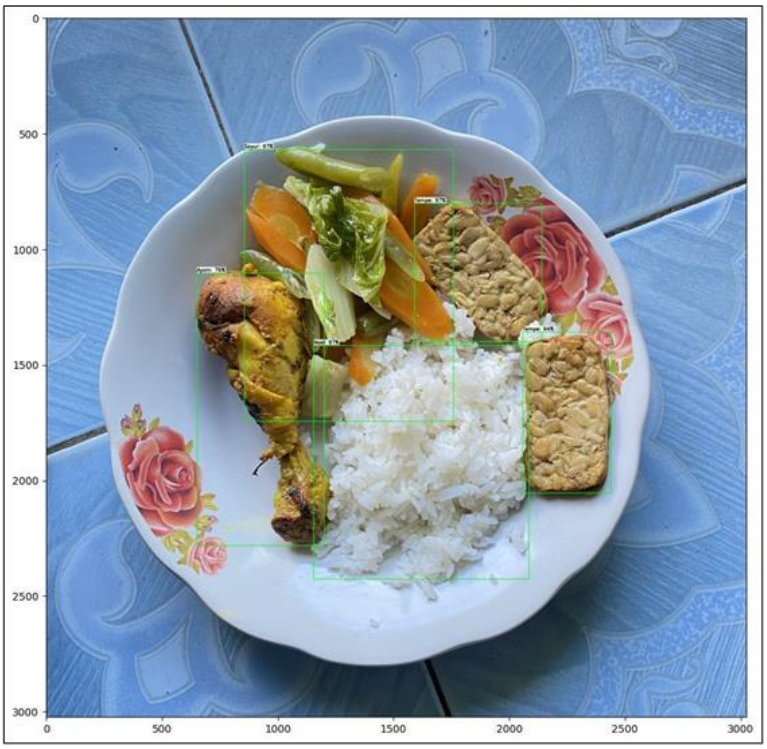
- Grafik visual



Gambar 3. 8 Grafik pelatihan

Model yang telah dilatih menunjukkan perbaikan performa selama proses pelatihan, dapat dilihat dengan penurunan nilai-nilai loss seiring bertambahnya jumlah iterasi. Meskipun terdapat fluktuasi pada nilai-nilai loss, tetapi bukan masalah dan merupakan hal yang wajar. Hal itu menunjukkan bahwa model sedang belajar dan meningkatkan kemampuannya untuk melakukan tugas-tugasnya.Prediksi akurasi

- Testing Model



Gambar 3. 9 Testing Model

Terlihat model telah dapat mendeteksi komponen makanan dari gambar yang telah diupload. Pada gambar tersebut terlihat *bounding box* yang memberikan hasil deteksi berupa nasi, ayam, sayur dan tempe. Masing-masing makanan tersebut dideteksi dengan tingkat kepercayaan diri sebesar 97% untuk nasi, 79% untuk ayam, 97% dan 94% untuk masing-masing tempe dan 97% untuk sayur. Dengan hasil deteksi dan tingkat kepercayaan diri sebesar itu, maka dapat dikatakan model telah berhasil melakukan deteksi makanan dengan cukup baik dan akurat.

4. Kesimpulan

Berdasarkan penelitian yang dilakukan terhadap pelatihan dan pengujian model CNN *transfer learning*, yaitu MobileNetV2, dapat disimpulkan bahwa arsitektur MobileNetV2 dalam *deep learning* memberikan performa yang sangat baik dalam identifikasi makanan sehat mahasiswa Teknik Universitas Bengkulu. Pelatihan MobileNetV2 dijalankan sebanyak 25.000 *epoch* dengan menggunakan dataset yang mencakup gambar makanan dari Kaggle dan foto makanan mandiri, yang setiap epoch memproses gambar untuk mengoptimalkan akurasi. Model ini mencapai tingkat akurasi yang tinggi, menunjukkan kemampuannya dalam mengenali dan mengklasifikasikan makanan sehat dengan akurat. Selama proses pelatihan, terjadi penurunan yang signifikan pada nilai *loss* yang menunjukkan bahwa model telah belajar dengan baik untuk mengidentifikasi pola dalam gambar makanan. Nilai *classification loss* dan *localization loss* menunjukkan hasil yang memadai, yang berarti model efektif dalam mengklasifikasikan makanan dengan presisi yang baik. Fluktuasi pada nilai *loss* tetap menunjukkan tren penurunan, menandakan bahwa model sedang berproses dengan optimal. Secara keseluruhan, model yang telah dilatih menunjukkan perbaikan performa yang signifikan sepanjang proses pelatihan, yang dapat dilihat melalui penurunan nilai-nilai *loss* seiring bertambahnya jumlah iterasi. Dibandingkan dengan model lain yang sudah diuji, MobileNetV2 terbukti menjadi pilihan optimal dalam mengidentifikasi makanan sehat, dengan hasil akurasi yang konsisten baik pada pelatihan maupun pengujian.