

**Nama : Arfara Yema Samgusdian**

**NIM : 1103202004**

**Kelas : TK-44-04**

## **Zero to Mastery Learn PyTorch for Deep Learning**

### **06. PyTorch Transfer Learning**

#### **0. Persiapan**

- Mengimpor/download modul yang diperlukan.
- Mengunduh direktori going\_modular dari repositori pytorch-deep-learning.
- Menginstal paket torchinfo jika belum tersedia.

#### **1. Mengambil Data**

- Mengunduh dataset pizza\_steak\_sushi.zip dari GitHub kursus.
- Memastikan dataset tidak diunduh ulang jika sudah ada.

#### **2. Membuat Dataset dan DataLoaders**

- Menggunakan skrip data\_setup.py dari direktori going\_modular untuk mempersiapkan DataLoaders.
- Menyiapkan transformasi khusus untuk model torchvision.models.

##### **2.1 Transformasi Manual untuk torchvision.models**

- Pentingnya mempersiapkan data dengan cara yang sama seperti data pelatihan asli.
- Sebelum torchvision v0.13+, transformasi untuk model pretrained dibuat secara manual.

##### **2.2 Transformasi Otomatis untuk torchvision.models**

- Mulai dari torchvision v0.13+, fitur pembuatan transformasi otomatis ditambahkan.
- Memilih berat pretrained dan arsitektur model menggunakan torchvision.models.EfficientNet\_B0\_Weights.DEFAULT.

#### **3. Mendapatkan Model Pretrained**

- Menggunakan transfer learning untuk meningkatkan kinerja model dengan mengambil model yang sudah ada dan menyesuaikannya dengan kasus penggunaan khusus.
- Mencari model klasifikasi pretrained di torchvision.models.

#### **4. Melatih Model**

- Membuat fungsi kerugian dan pengoptimal.
- Menggunakan nn.CrossEntropyLoss() sebagai fungsi kerugian.
- Menggunakan torch.optim.Adam() sebagai pengoptimal dengan lr=0.001.
- Melatih model selama 5 epoch menggunakan fungsi train() dari skrip engine.py.

#### **5. Evaluasi Model dengan Plot Kurva Kerugian**

- Performa model terlihat baik, dan untuk memahami pelatihannya, kita akan membuat plot kurva kerugian menggunakan fungsi plot\_loss\_curves.
- Fungsi ini diambil dari helper\_functions.py dan diunduh jika belum tersedia.
- Plot kurva kerugian menggunakan fungsi plot\_loss\_curves(results).

#### **6. Prediksi pada Gambar dari Set Uji**

- Model menunjukkan performa kuantitatif yang baik, tetapi bagaimana dengan kualitatif?
- Membuat prediksi pada gambar dari set uji dan menampilkannya.

- Perlu memastikan gambar memiliki format yang sesuai dengan gambar yang digunakan untuk melatih model.
- Membuat fungsi `pred_and_plot_image()` untuk prediksi dan plotting.
- Fungsi `pred_and_plot_image()`:
- Mengambil model terlatih, daftar nama kelas, path gambar, ukuran gambar, transformasi, dan perangkat target.
- Membuka gambar dengan `PIL.Image.open()`.
- Membuat transformasi untuk gambar, default ke `manual_transforms` atau dapat menggunakan transformasi dari `weights.transforms()`.
- Memastikan model berada pada perangkat target.
- Mengaktifkan mode evaluasi model dan mode inferensi.
- Melakukan transformasi dan menambah dimensi batch ke gambar.
- Membuat prediksi pada gambar dengan dimensi tambahan dan mengirimkannya ke perangkat target.
- Mengonversi logits model ke probabilitas prediksi dengan `torch.softmax()`.
- Mengonversi probabilitas prediksi model ke label prediksi dengan `torch.argmax()`.
- Menampilkan gambar dengan label dan probabilitas prediksi.

### 6.1 Prediksi pada Gambar Kustom

- Model tampaknya berhasil secara kualitatif pada data dari set uji. Kini kita akan menguji model pada gambar kustom.
- Mengunduh gambar kustom (`pizza-dad.jpeg`) dan memprediksi dengan fungsi `pred_and_plot_image`.