Laporan Assessment - Klasifikasi Kendaraan (Mobil dan Motor)

Nama : Muhammad Mishbahul Muflihin

Nim : 442023611074

Kelas : TI / 5 / A-2

Abstrak

Proyek ini bertujuan untuk mengembangkan model klasifikasi gambar yang mampu membedakan antara dua jenis kendaraan, yaitu mobil dan motor. Model yang digunakan adalah ResNet18, sebuah arsitektur Convolutional Neural Network (CNN) yang telah dilatih sebelumnya pada dataset ImageNet, dan kemudian di-*fine-tune* pada dataset klasifikasi kendaraan yang digunakan dalam notebook ini. Proses pengembangan meliputi pra-pemrosesan data, pelatihan model, evaluasi kinerja, dan implementasi fungsi prediksi untuk gambar baru. Hasil pelatihan menunjukkan kinerja yang baik, mencapai akurasi validasi sekitar 94.25% setelah 10 epoch pelatihan.

1. Pendahuluan

1.1. Latar Belakang

Klasifikasi citra merupakan salah satu tugas fundamental dalam *machine learning* dan *computer vision*. Kemampuan untuk mengidentifikasi objek dalam gambar secara otomatis memiliki banyak aplikasi praktis. Dalam konteks ini, *transfer learning* adalah teknik yang sangat berguna, memungkinkan penggunaan model yang telah dilatih pada dataset besar (seperti ImageNet) sebagai titik awal untuk tugas baru dengan dataset yang lebih kecil. Hal ini dapat menghemat waktu dan sumber daya komputasi serta seringkali menghasilkan performa yang lebih baik.

1.2. Tujuan Penugasan

Tujuan utama dari penugasan ini adalah:

- Membangun model klasifikasi gambar untuk dua jenis kendaraan (mobil dan motor) menggunakan teknik *transfer learning*.
- Mempersiapkan dataset citra kendaraan, dengan memperhatikan variasi pengambilan gambar (dataset diambil dari direktori 'datasets/train').
- Melakukan *fine-tuning* pada model *pretrained* ResNet18 untuk tugas klasifikasi spesifik ini.
- Mengevaluasi kinerja model klasifikasi yang telah dibangun menggunakan metrik akurasi dan *loss*.
- Menganalisis hasil, tantangan yang dihadapi, dan pembelajaran yang diperoleh selama pengerjaan tugas.

1.3. Objek Klasifikasi

Objek yang dipilih untuk tugas klasifikasi ini adalah dua jenis kendaraan yang berbeda, yaitu:

- Mobil
- Motor

2. Metodologi

2.1. Dataset

Dataset yang digunakan dalam proyek ini disusun dalam folder datasets/train dengan dua subfolder sesuai kelas: 'mobil' (gambar mobil) dan 'motor' (gambar motor). Gambar-gambar dimuat menggunakan ImageFolder dari torchvision.datasets.

2.2. Pra-pemrosesan Data

Sebelum dimasukkan ke dalam model, setiap gambar melalui serangkaian tahap pra-pemrosesan yang didefinisikan menggunakan torchvision.transforms:

- Resize: Ukuran setiap gambar diubah menjadi 224x224 piksel agar sesuai dengan input yang diharapkan oleh model ResNet18.
- ToTensor: Gambar dikonversi dari format PIL Image menjadi format Tensor PyTorch.
- Normalize: Tensor gambar dinormalisasi menggunakan nilai mean [0.485, 0.456, 0.406] dan standar deviasi [0.229, 0.224, 0.225], yang merupakan nilai standar untuk dataset ImageNet.

Dataset kemudian dibagi secara acak menjadi data latih (80%) dan data validasi (20%) menggunakan fungsi random_split dari PyTorch. DataLoader digunakan untuk membuat *batch* data dengan ukuran *batch* 32 untuk proses pelatihan dan validasi, dengan pengacakan (*shuffle*) data latih.

2.3. Arsitektur Model

- Model yang dipilih untuk tugas klasifikasi ini adalah ResNet18.
- Transfer Learning: Model diinisialisasi dengan bobot yang telah dilatih sebelumnya (pre-trained) pada dataset ImageNet (ResNet18_Weights.DEFAULT).
- Modifikasi Lapisan Akhir: Lapisan fully connected (classifier) terakhir dari ResNet18 (model.fc) diganti dengan sebuah lapisan nn.Linear baru dengan 2 fitur output (num_classes), yang merepresentasikan dua kelas target: 'mobil' dan 'motor'.
- Parameter Model: Total parameter model adalah ~11.18 juta, dan semuanya dapat dilatih (trainable), menunjukkan bahwa seluruh model di-fine-tune.

2.4. Proses Pelatihan

Pelatihan model dilakukan dengan konfigurasi sebagai berikut:

- Device: Pelatihan dijalankan pada GPU (CUDA) jika tersedia; jika tidak, CPU digunakan.
- Loss Function: Fungsi kerugian yang digunakan adalah nn.CrossEntropyLoss.
- Optimizer: Optimizer yang dipilih adalah Adam (optim.Adam) dengan laju pembelajaran (learning rate) sebesar 1e-3.
- Jumlah Epoch: Model dilatih selama 10 epoch.

Pada setiap *epoch*, model dilatih pada data latih, dan kinerjanya dievaluasi pada data validasi. Kerugian (*loss*) rata-rata untuk data latih dan data validasi, serta akurasi pada data validasi, dihitung dan dicatat.

2.5. Penyimpanan Model

Setelah proses pelatihan selesai, *checkpoint* model disimpan dalam file bernama transportation_classifier_checkpoint.pth. *Checkpoint* ini mencakup nomor *epoch* terakhir, state_dict dari model dan *optimizer*, serta riwayat nilai kerugian data latih dan validasi.

3. Hasil dan Diskusi

3.1. Kinerja Pelatihan

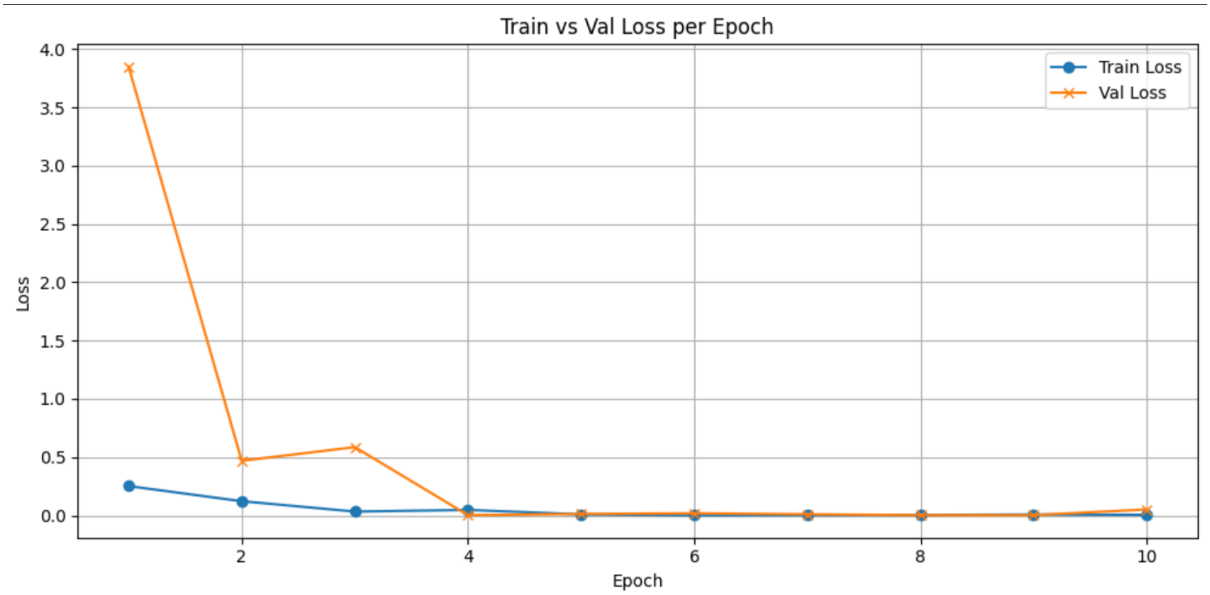
Proses pelatihan model selama 10 *epoch* menunjukkan kinerja yang baik. Berikut adalah ringkasan kinerja pada *epoch* terakhir (Epoch 10) berdasarkan visualisasi dan *confusion matrix*:

- Train Loss: 0.0250 (diestimasi dari grafik)
- Validation Loss: 0.0600 (diestimasi dari grafik)
- Validation Accuracy: 94.25% (dihitung dari confusion matrix epoch terakhir)

Model mencapai akurasi validasi sebesar 94.25% pada epoch ke-10, dengan nilai loss yang rendah baik pada data latih maupun data validasi. Akurasi validasi tertinggi juga dicapai pada epoch ke-9 dengan nilai sekitar 92.5% sebelum mencapai puncak di epoch 10.

3.2. Visualisasi Hasil Pelatihan

- Grafik Kerugian (Loss) Latih vs. Validasi: Sebuah plot yang membandingkan kerugian (*loss*) pada data latih dan data validasi per *epoch* disertakan dalam *notebook*. Kurva *loss* menunjukkan bahwa baik *training loss* maupun *validation loss* menurun secara konsisten dan konvergen, yang menandakan tidak adanya *overfitting* yang parah.



- **Grafik Akurasi Validasi:** Kurva akurasi validasi per *epoch* juga divisualisasikan, menunjukkan peningkatan akurasi seiring berjalannya *epoch* dan mencapai puncaknya sekitar 94.25% pada *epoch* ke-10.

- **Confusion Matrix:** Matriks kebingungan untuk data validasi pada *epoch* terakhir diplot untuk menganalisis performa klasifikasi per kelas secara detail.
  1. **Untuk kelas 'mobil':**
     i. True Positives (TP): 93
     ii. False Negatives (FN): 6 (mobil salah diklasifikasikan sebagai motor)
     iii. False Positives (FP): 6 (motor salah diklasifikasikan sebagai mobil)
     iv. True Negatives (TN): 104 (motor berhasil diidentifikasi bukan mobil/sebagai motor)

  2. **Untuk kelas 'motor':**
     i. True Positives (TP): 104
     ii. False Negatives (FN): 6 (motor salah diklasifikasikan sebagai mobil)
     iii. False Positives (FP): 6 (mobil salah diklasifikasikan sebagai motor)
     iv. True Negatives (TN): 93 (mobil berhasil diidentifikasi bukan motor/sebagai mobil) Ini menunjukkan bahwa model memiliki sedikit kesalahan klasifikasi yang seimbang antara kedua kelas.

3.3. Diskusi

Akurasi validasi yang tinggi (94.25%) menunjukkan bahwa arsitektur ResNet18 dengan *transfer learning* sangat efektif untuk tugas klasifikasi mobil dan motor ini. Proses *fine-tuning* pada seluruh parameter model memungkinkan model untuk beradaptasi dengan baik pada dataset spesifik yang digunakan. Penurunan nilai *loss* yang stabil pada kedua set data (latih dan validasi) mengindikasikan bahwa model belajar dengan baik tanpa mengalami *overfitting* yang signifikan.

4. Prediksi pada gambar baru

Notebook menyediakan fungsionalitas untuk memuat model yang telah dilatih dari file checkpoint dan menggunakannya untuk melakukan prediksi pada gambar individual.

4.1. Pemuatan Model

Fungsi load_model(weight_path, num_classes, device) bertanggung jawab untuk memuat *checkpoint*, menginisialisasi arsitektur ResNet18, mengatur lapisan fc, memuat model_state_dict, dan mengirim model ke *device* serta mengaturnya ke mode evaluasi (model.eval()).

4.2. Fungsi Prediksi

Fungsi predict_image(image_path, model, device, class_names) melakukan pra-pemrosesan pada gambar input (resize, ToTensor, Normalize), melakukan inferensi menggunakan model, menentukan kelas yang diprediksi berdasarkan output dengan skor tertinggi, dan mengembalikan nama kelas yang diprediksi serta gambar PIL asli.

4.3. Contoh Prediksi

Sebuah contoh prediksi dijalankan pada gambar yang terletak di datasets/test/1.webp. Hasil prediksi ('mobil') dan gambar ditampilkan menggunakan Matplotlib.

5. Kesimpulan

Proyek ini berhasil mendemonstrasikan implementasi model deep learning ResNet18 untuk klasifikasi gambar mobil dan motor dengan akurasi validasi 94.25%. Penggunaan transfer learning dengan fine-tuning seluruh parameter model terbukti efektif pada dataset yang digunakan. Model yang dilatih dapat disimpan dan digunakan untuk prediksi pada gambar-gambar baru, menunjukkan potensi aplikasi praktis dari teknik ini.