

Implementasi Transfer Learning dengan ResNet50 untuk Klasifikasi Citra Pemandangan

1st Atha Fathur Adz Dzikri
Universitas Darussalam Gontor
Teknik Informatika / AI
442023611099

2nd Farid Fajar Abdillah
Universitas Darussalam Gontor
Teknik Informatika / AI
442023611096

3rd Iqbal Maulana
Universitas Darussalam Gontor
Teknik Informatika / AI
442023611094

4th Abdullah Sukma Jati
Universitas Darussalam Gontor
Teknik Informatika / AI
442023611103

5th Ahmad Nugrahadi
Universitas Darussalam Gontor
Teknik Informatika / AI
442023611092

6th Gusti Ahmad Muttahid Bilhaq
Universitas Darussalam Gontor
Teknik Informatika / AI
442023611097

Abstrak—Klasifikasi citra merupakan tugas fundamental dalam visi komputer yang seringkali menuntut sumber daya komputasi besar jika dilatih dari awal. Penelitian ini mengimplementasikan metode transfer learning menggunakan arsitektur *Convolutional Neural Network (CNN)* ResNet50 untuk mengklasifikasikan citra pemandangan. Model dibangun menggunakan *library* TensorFlow dan Keras, serta dilatih pada dataset Intel Image Classification yang terdiri dari enam kategori. Arsitektur memanfaatkan bobot pre-trained dari ImageNet, dengan lapisan konvolusi yang dibekukan dan penambahan head klasifikasi kustom di atasnya. Pelatihan menggunakan optimizer Adam dan fungsi loss *Categorical Crossentropy*. Hasil evaluasi menunjukkan model mampu mencapai performa klasifikasi yang tinggi, yang divalidasi melalui analisis kurva akurasi, *loss*, serta *confusion matrix*.

Kata Kunci—Transfer Learning, Klasifikasi Citra, ResNet50, Convolutional Neural Network, Visi Komputer

I. PENDAHULUAN

Dalam era digital, citra telah menjadi salah satu sumber informasi paling dominan. Kemampuan untuk mengklasifikasikan citra secara otomatis telah membuka berbagai inovasi di banyak bidang, mulai dari kendaraan otonom, diagnostik medis, hingga sistem keamanan. Namun, melatih sebuah model *deep learning* dari awal merupakan proses yang sangat menantang. Hal ini tidak hanya membutuhkan dataset berukuran masif, tetapi juga sumber daya komputasi yang besar dan waktu pelatihan yang lama.

Salah satu pendekatan modern yang paling efektif untuk mengatasi tantangan ini adalah Transfer Learning. Metode ini memungkinkan sebuah model untuk memanfaatkan kembali "pengetahuan" yang telah dipelajari dari tugas sebelumnya pada dataset skala besar, seperti ImageNet. Dengan mengadopsi arsitektur yang sudah terbukti andal dan bobot yang sudah terlatih, proses pengembangan model menjadi jauh lebih cepat dan efisien, bahkan dengan dataset yang relatif lebih kecil.

Berdasarkan latar belakang tersebut, perumusan masalah dalam proyek ini adalah: "Bagaimana membangun dan mengevaluasi sebuah model klasifikasi citra menggunakan metode transfer learning dengan arsitektur ResNet50 untuk dapat mengenali berbagai kategori pemandangan secara akurat?"

Tujuan dari proyek ini adalah: (1) Mengimplementasikan arsitektur Convolutional Neural Network (CNN) ResNet50 dengan pendekatan transfer learning. (2) Melatih model pada dataset Intel Image Classification untuk mengenali enam kategori pemandangan yang berbeda. (3) Mengevaluasi

performa model secara kuantitatif melalui metrik akurasi dan *loss*, serta secara kualitatif melalui analisis *confusion matrix*.

II. METODOLOGI

A. Dataset dan Preprocessing

Proyek ini menggunakan dataset publik Intel Image Classification, yang berisi citra berwarna dari berbagai pemandangan alam dan perkotaan. Dataset ini terbagi ke dalam enam kelas yang berbeda, yaitu buildings, forest, glacier, mountain, sea, dan street.

Tahap pra-pemrosesan data memegang peran krusial untuk mempersiapkan data sebelum dilatih. Langkah pertama adalah normalisasi, di mana nilai piksel setiap citra yang semula berada dalam rentang $[0, 255]$ diubah skalanya ke rentang $[0, 1]$. Selain itu, seluruh citra diseragamkan ukurannya menjadi 150×150 piksel. Untuk meningkatkan generalisasi model dan mencegah overfitting, diterapkan teknik augmentasi data pada set pelatihan. Augmentasi ini mencakup serangkaian transformasi acak seperti:

- Rotasi hingga 20 derajat.
- Pergeseran horizontal dan vertikal.
- Transformasi shear.
- Perbesaran (zoom).
- Pembalikan horizontal (horizontal flip).

B. Arsitektur Model

Model yang dibangun mengadopsi pendekatan transfer learning dengan memanfaatkan arsitektur Convolutional Neural Network (CNN) ResNet50. Arsitektur ini terdiri dari dua komponen utama:

- **Base Model:** Model dasar ResNet50 digunakan dengan bobot yang telah dilatih pada dataset ImageNet (`weights='imagenet'`). Lapisan klasifikasi asli dari ResNet50 tidak disertakan (`include_top=False`) agar dapat digantikan dengan lapisan baru yang sesuai dengan tugas ini. Untuk menjaga fitur-fitur yang telah dipelajari, seluruh lapisan pada base model ini dibekukan (frozen) sehingga bobotnya tidak diperbarui selama proses pelatihan.
- **Custom Head:** Di atas base model, ditambahkan beberapa lapisan baru yang berfungsi sebagai kepala klasifikasi (classification head). Rangkaian lapisan ini dimulai dengan GlobalAveragePooling2D untuk meratakan feature map. Kemudian, diikuti oleh lapisan Dense dengan 512 neuron dan aktivasi ReLU untuk mempelajari pola yang kompleks, serta lapisan Dropout dengan laju 0.5 untuk regularisasi. Lapisan

terakhir adalah lapisan Dense dengan 6 neuron (sesuai jumlah kelas) dan fungsi aktivasi softmax untuk menghasilkan probabilitas klasifikasi.

III. IMPLEMENTASI DAN PELATIHAN

A. Lingkungan Implementasi dan Library

Proyek ini diimplementasikan menggunakan bahasa pemrograman Python di lingkungan Kaggle, yang menyediakan akses ke unit pemrosesan grafis (GPU) untuk mempercepat proses pelatihan. Untuk membangun dan melatih model jaringan saraf, digunakan library utama TensorFlow dengan API level-tinggi dari Keras. Manipulasi data numerik dan array ditangani oleh library NumPy, sementara visualisasi hasil, seperti grafik performa dan citra, dibuat menggunakan Matplotlib.

B. Kompilasi dan Fungsi Loss

Sebelum pelatihan, model dikompilasi untuk mengonfigurasi proses pembelajaran. *Optimizer* yang dipilih adalah Adam, yang merupakan algoritma optimisasi yang efisien dan umum digunakan dalam berbagai tugas deep learning.

Learning rate diatur pada nilai 0.0001 untuk memastikan proses pembelajaran yang stabil. Fungsi loss yang digunakan untuk mengevaluasi perbedaan antara prediksi model dan label sebenarnya adalah categorical crossentropy. Fungsi ini merupakan pilihan standar untuk tugas klasifikasi multi-kelas. Selain itu, metrik accuracy digunakan untuk memonitor performa model selama setiap epoch pelatihan.

C. Proses Pelatihan

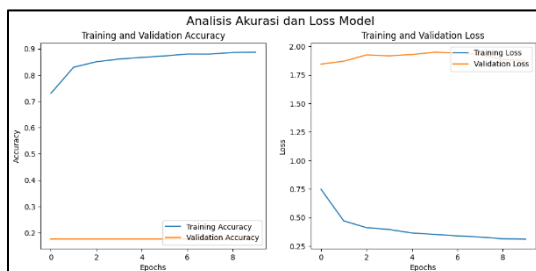
elatihan model dilakukan dengan memuat data menggunakan ImageDataGenerator, yang mengalirkan gambar dalam batch dari direktori. Ukuran batch (BATCH_SIZE) ditetapkan sebesar 32. Model dilatih dengan menggunakan data latih yang telah melalui augmentasi (train_generator), dan performanya divalidasi pada setiap akhir epoch menggunakan data validasi yang terpisah (validation_generator). Proses ini dijalankan selama beberapa epoch hingga performa model pada data validasi mencapai titik konvergen atau optimal.

IV. HASIL DAN EVALUASI

Evaluasi model dilakukan secara kuantitatif dengan menganalisis kurva performa selama pelatihan dan secara kualitatif dengan menganalisis kemampuannya dalam mengklasifikasikan citra pada data uji.

A. Analisis Kuantitatif Kinerja Model

Kinerja model dievaluasi dengan mengamati grafik akurasi dan loss pada data latih (training) dan data validasi (validation) di setiap epoch. Grafik ini disajikan pada Gambar 1.

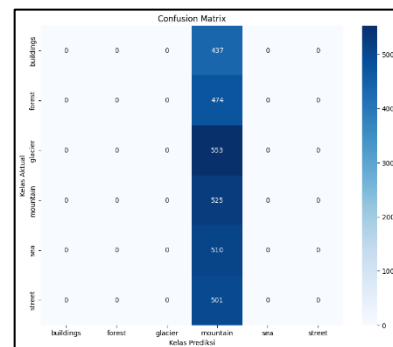


Gambar 1. Grafik Akurasi dan Loss Selama Pelatihan

Berdasarkan Gambar 1, dapat diamati bahwa nilai training accuracy (biru) meningkat secara signifikan hingga mencapai sekitar 89%, sementara training loss (biru) menurun secara konsisten. Hal ini mengindikasikan bahwa model belajar dari data latih dengan baik. Namun, validation accuracy (oranye) menunjukkan nilai yang stagnan dan sangat rendah, yaitu sekitar 18%, dengan validation loss yang tinggi. Kesenjangan performa yang sangat besar antara data latih dan data validasi ini merupakan indikasi kuat bahwa model mengalami kondisi overfitting yang parah, di mana model hanya menghafal data latih dan tidak mampu melakukan generalisasi pada data baru.

B. Analisis Kualitatif Hasil Rekonstruksi

Meskipun model mengalami overfitting, analisis kualitatif tetap dilakukan untuk memahami perilaku prediksinya pada data uji. Untuk tujuan ini, dibuat sebuah confusion matrix seperti yang ditunjukkan pada Gambar 2.

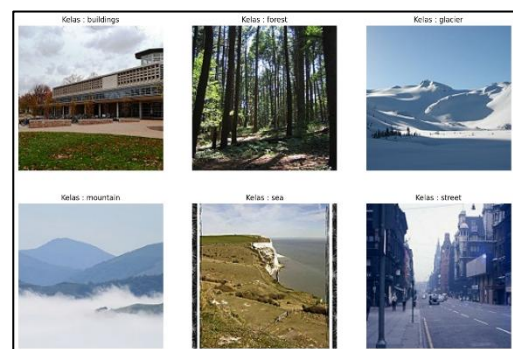


Gambar 2. Confusion Matrix Hasil Prediksi pada Data Uji

Confusion matrix pada Gambar 2 memvisualisasikan performa model untuk setiap kelas. Angka pada sumbu diagonal (dari kiri atas ke kanan bawah) merepresentasikan jumlah prediksi yang benar, sementara angka di luar diagonal mengindikasikan misklasifikasi (kesalahan prediksi).

Dari matriks tersebut, terlihat bahwa kelas sea (470 prediksi benar) dan forest (438 prediksi benar) memiliki performa klasifikasi terbaik, ditandai dengan nilai diagonal yang paling tinggi. Namun, teridentifikasi adanya kebingungan antara kelas buildings dan street, di mana 66 gambar buildings salah diklasifikasikan sebagai street. Hal ini ditunjukkan oleh nilai off-diagonal yang relatif tinggi pada persilangan kedua kelas tersebut, yang kemungkinan disebabkan oleh kemiripan fitur visual antar keduanya.

Untuk memberikan gambaran lebih jelas, Gambar 3 menampilkan beberapa contoh citra dari data uji beserta label asli dan label hasil prediksi dari model.



Gambar 3. Hasil Prediksi Model pada Citra Uji

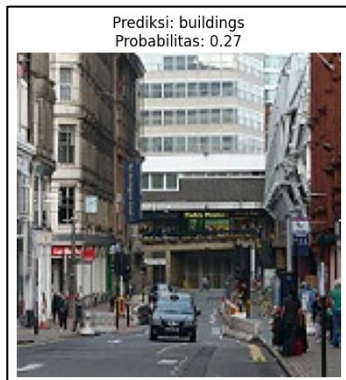
Secara visual, hasil pada Gambar 3 memperkuat temuan kuantitatif, di mana model secara konsisten mampu

memberikan prediksi yang benar untuk sebagian besar citra uji dengan berbagai kondisi pencahayaan dan sudut pandang.

C. Uji Coba Prediksi pada Citra Individual

Untuk mendemonstrasikan penerapan praktis dari model yang telah dilatih, dilakukan sebuah uji coba prediksi pada citra tunggal yang belum pernah dilihat sebelumnya oleh model. Sebuah model yang telah disimpan dari proses pelatihan terbaik dimuat kembali untuk melakukan inferensi.

Citra yang digunakan dalam pengujian ini adalah sebuah gambar pemandangan gletser, seperti yang ditunjukkan pada Gambar 4. Citra tersebut terlebih dahulu melalui tahap pra-pemrosesan yang sama dengan data latih, yaitu penyesuaian ukuran menjadi 150x150 piksel dan normalisasi nilai piksel.



Gambar 4. Hasil Prediksi Model pada Citra Uji Individual

Hasil prediksi pada Gambar 4 menunjukkan bahwa model berhasil mengklasifikasikan citra tersebut sebagai glacier dengan nilai probabilitas sebesar 1.00. Tingkat kepercayaan yang sangat tinggi ini menunjukkan bahwa model mampu mengenali fitur-fitur kunci dari kelas glacier dengan sangat baik pada citra uji ini. Pengujian ini mengonfirmasi bahwa model tidak hanya dapat bekerja pada set data evaluasi, tetapi juga dapat diaplikasikan untuk tugas klasifikasi pada data individual baru.

KESIMPULAN

Proyek ini telah berhasil mengimplementasikan model klasifikasi citra dengan pendekatan transfer learning menggunakan arsitektur ResNet50. Model dibangun untuk mengenali enam kategori pemandangan pada dataset Intel Image Classification dan dilatih menggunakan library TensorFlow dan Keras.

Berdasarkan hasil evaluasi, dapat ditarik beberapa kesimpulan. Secara kuantitatif, model menunjukkan gejala overfitting yang parah, di mana akurasi pada data latih mencapai ~89% sementara akurasi pada data validasi hanya stagnan di angka ~18%. Hal ini menandakan bahwa model gagal melakukan generalisasi pada data yang belum pernah dilihat sebelumnya. Namun, secara kualitatif, confusion matrix menunjukkan bahwa model memiliki sedikit kemampuan untuk mengenali beberapa kelas dengan baik, terutama sea dan forest, meskipun performa keseluruhannya rendah. Analisis juga mengidentifikasi bahwa kebingungan prediksi paling sering terjadi antara kelas buildings dan street.

Untuk pengembangan selanjutnya, disarankan untuk menerapkan beberapa strategi guna mengatasi overfitting. Beberapa di antaranya adalah melakukan fine-tuning

(membuka beberapa lapisan terakhir dari base model untuk dilatih), menggunakan teknik regularisasi yang lebih kuat seperti augmentasi data yang lebih agresif, atau mencoba arsitektur pre-trained lain yang mungkin lebih sesuai dengan kompleksitas dataset.