Klasifikasi Pemandangan Alam Menggunakan MobileNetV3 dan Teknik Fine-tuning

Muhammad Abiya Makruf (203012420034)

Abstrak

Klasifikasi pemandangan alam secara otomatis merupakan tugas fundamental dalam visi komputer dengan aplikasi yang luas, mulai dari sistem navigasi hingga pemantauan lingkungan. Proyek ini mengatasi tantangan klasifikasi enam kategori pemandangan alam (bangunan, hutan, gletser, gunung, laut, dan jalan) dari dataset Intel Image Classification. Pendekatan utama yang digunakan adalah *deep learning* dengan arsitektur Convolutional Neural Network (CNN) MobileNetV3, memanfaatkan efisiensi dan performanya. Dua varian, MobileNetV3 Small dan MobileNetV3 Large, dieksplorasi menggunakan teknik *transfer learning*. Eksperimen dilakukan dalam dua tahap: tanpa *fine-tuning* sebagai *baseline* dan *fine-tuning* untuk adaptasi yang lebih mendalam. Hasil menunjukkan bahwa *fine-tuning* secara signifikan meningkatkan performa, dengan MobileNetV3 Large yang di-*fine-tune* mencapai akurasi tes tertinggi sebesar **93.63%**. Selain evaluasi kuantitatif, interpretabilitas model dianalisis menggunakan Grad-CAM. Proyek ini juga menghasilkan aplikasi web interaktif berbasis Streamlit untuk demonstrasi klasifikasi secara langsung.

1. Introduction

Pengenalan pemandangan alam adalah kemampuan inti dalam sistem persepsi visual, baik biologis maupun artifisial. Dengan meningkatnya volume data citra digital, kebutuhan akan sistem otomatis yang dapat memahami dan mengkategorikan konten visual dari pemandangan alam menjadi semakin penting. Aplikasi dari teknologi ini sangat beragam, mencakup sistem navigasi untuk kendaraan otonom, pemetaan penggunaan lahan, pemantauan perubahan iklim melalui analisis citra satelit, pengorganisasian koleksi foto digital, hingga sistem rekomendasi konten. Meskipun manusia dapat dengan mudah mengenali berbagai jenis pemandangan, membangun sistem komputasi yang mampu melakukan hal serupa dengan akurasi tinggi tetap menjadi tantangan karena variasi intra-kelas yang besar (misalnya, berbagai jenis bangunan atau hutan) dan kemiripan antar-kelas (misalnya, beberapa jenis gunung dan gletser).

Proyek ini bertujuan untuk mengembangkan dan mengevaluasi sistem klasifikasi pemandangan alam yang efektif menggunakan pendekatan *deep learning*. Dataset yang digunakan adalah "Intel Image Classification" yang populer dari Kaggle, terdiri dari enam kategori pemandangan umum. Fokus utama adalah pada implementasi dan evaluasi arsitektur MobileNetV3, yang dikenal karena keseimbangan antara efisiensi komputasional dan akurasi, menjadikannya kandidat yang baik untuk aplikasi potensial di perangkat dengan sumber daya terbatas atau aplikasi web *real-time*. Kami mengeksplorasi MobileNetV3 Small dan MobileNetV3 Large, menerapkan strategi *transfer learning* yang melibatkan tahap ekstraksi fitur dan *fine-tuning* secara menyeluruh.

Hasil eksperimen menunjukkan bahwa pendekatan *fine-tuning* pada model MobileNetV3 Large berhasil mencapai akurasi klasifikasi sebesar **93.63%** pada *test set*, menunjukkan efektivitas metode yang diusulkan. Laporan ini akan merinci dataset yang digunakan, metodologi pra-pemrosesan, arsitektur model, proses pelatihan dan evaluasi, serta analisis hasil termasuk interpretabilitas model menggunakan Grad-CAM. Sebuah aplikasi web interaktif juga dikembangkan untuk mendemonstrasikan fungsionalitas sistem klasifikasi ini.

2. Related Work

Klasifikasi citra, khususnya pemandangan alam, telah menjadi area penelitian aktif selama beberapa dekade. Pendekatan awal seringkali bergantung pada ekstraksi fitur manual seperti SIFT, SURF, HOG, atau deskriptor tekstur yang kemudian dimasukkan ke dalam *classifier* tradisional seperti Support Vector Machines (SVM) atau Random Forests. Meskipun metode ini memberikan dasar yang kuat, performanya seringkali terbatas oleh kualitas fitur yang diekstraksi secara manual dan ketidakmampuannya untuk menangkap representasi hierarkis yang kompleks dari data visual.

Revolusi *deep learning*, terutama dengan Convolutional Neural Networks (CNNs), telah secara dramatis mengubah lanskap klasifikasi citra. Model seperti AlexNet [1], VGG [2], GoogLeNet [3], dan ResNet [4] menunjukkan performa SOTA pada dataset skala besar seperti ImageNet [5]. Keberhasilan ini disebabkan oleh kemampuan CNN untuk belajar fitur hierarkis langsung dari data mentah, menghilangkan kebutuhan akan rekayasa fitur manual yang rumit.

Untuk tugas klasifikasi pemandangan alam, berbagai arsitektur CNN telah diterapkan. Dataset seperti Scene-15, MIT Indoor-67, dan SUN397 sering digunakan sebagai *benchmark*. Penelitian seringkali fokus pada adaptasi arsitektur CNN yang sudah ada atau pengembangan arsitektur baru yang lebih sesuai untuk menangkap karakteristik spasial dan kontekstual dari pemandangan.

Transfer learning telah menjadi teknik standar dalam visi komputer, terutama ketika dataset target relatif kecil dibandingkan dengan dataset seperti ImageNet. Dengan menggunakan bobot dari model yang telah dilatih sebelumnya (pre-trained) pada ImageNet, model dapat belajar fitur visual umum yang kemudian dapat diadaptasi untuk tugas spesifik dengan dataset yang lebih kecil, sehingga mempercepat konvergensi dan seringkali meningkatkan performa.

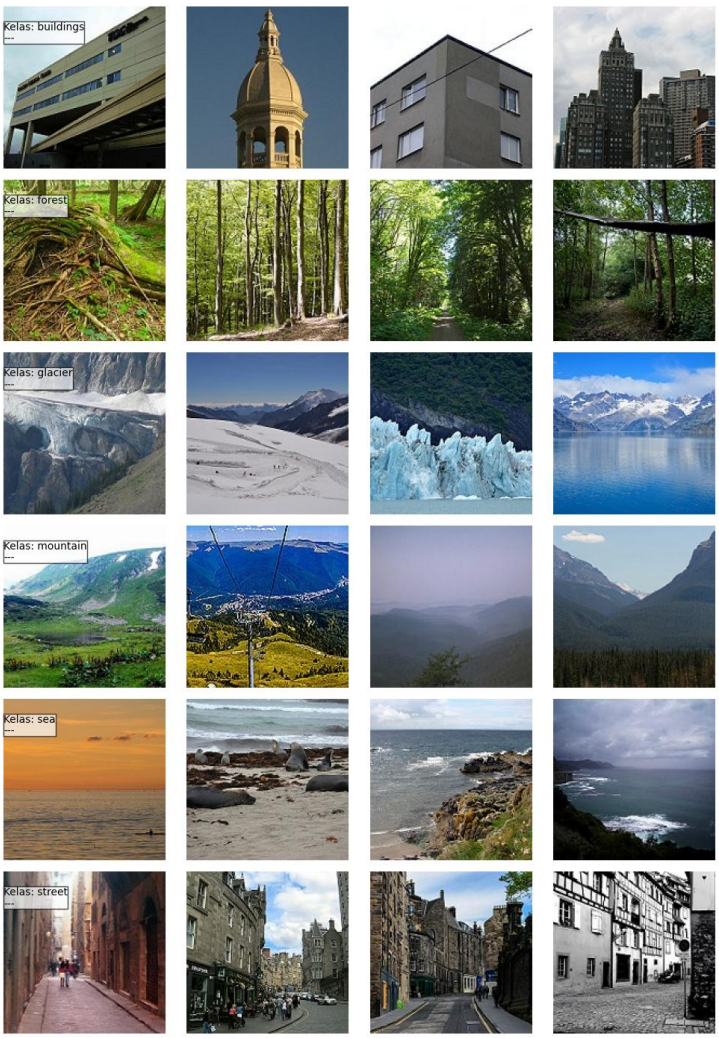
Arsitektur MobileNet [6], termasuk MobileNetV1, V2, dan V3 [7], dirancang khusus untuk aplikasi pada perangkat *mobile* dan *edge* di mana efisiensi komputasional (latensi rendah, ukuran model kecil) sangat penting. MobileNetV3, yang digunakan dalam proyek ini, menggabungkan *depthwise separable convolutions*, blok *inverted residual* dengan *linear bottleneck*, dan arsitektur yang dioptimalkan menggunakan *Network Architecture Search* (NAS) serta modifikasi seperti *Squeeze-and-Excite* (SE) block dan penggunaan aktivasi h-swish. Pendekatan ini memungkinkan MobileNetV3 mencapai akurasi yang kompetitif dengan jumlah parameter dan operasi komputasi yang jauh lebih sedikit dibandingkan arsitektur yang lebih besar.

Untuk memahami dan menginterpretasikan keputusan model CNN, teknik visualisasi seperti Gradient-weighted Class Activation Mapping (Grad-CAM) [8] telah dikembangkan. Grad-

CAM menghasilkan *heatmap* yang menyoroti daerah penting pada citra input yang paling mempengaruhi prediksi kelas tertentu, memberikan wawasan tentang "apa yang dilihat" oleh model. Ini penting untuk validasi kualitatif dan membangun kepercayaan pada sistem *deep learning*. Proyek ini mengadopsi MobileNetV3 dengan *transfer learning* dan menggunakan Grad-CAM untuk interpretabilitas, sejalan dengan praktik SOTA dalam klasifikasi citra yang efisien dan dapat dipahami.

3. Data

Proyek ini menggunakan dataset publik "Intel Image Classification" yang bersumber dari Kaggle. Dataset ini dirancang untuk tugas klasifikasi pemandangan alam dan berisi gambar-gambar yang dikategorikan ke dalam enam kelas berbeda, yaitu buildings (bangunan), forest (hutan), glacier (gletser), mountain (gunung), sea (laut), street (jalan) seperti yang terlihat pada Gambar 1.



Gambar 1. Contoh Gambar Masing Masing Kelas Beserta Variasinya.

Dataset ini terbagi menjadi tiga direktori utama: seg_train (untuk data latih), seg_test (untuk data uji), dan seg_pred (berisi gambar tambahan, tidak digunakan dalam pelatihan atau evaluasi proyek ini). Untuk keperluan proyek ini, direktori seg_train digunakan untuk melatih model,

dan seg_test digunakan untuk evaluasi akhir performa model. Distribusi gambar antar kelas relatif seimbang, meskipun ada sedikit variasi jumlah gambar per kelas seperti yang terlihat pada Tabel 1. Gambar-gambar dalam dataset ini memiliki resolusi asli 150x150 piksel.

Tabel 1. Distribusi Gambar Antar Kelas.

Kelas	Train	Test	Total
Buildings	2191	437	2628
Forest	2271	474	2745
Glacier	2404	553	2957
Mountain	2512	525	3037
Sea	2274	510	2784
Street	2382	501	2883
Total	14034	3000	17034

Pra-pemrosesan Data: Beberapa langkah pra-pemrosesan diterapkan pada data sebelum dimasukkan ke model:

1. **Pembagian Data:** Data dari direktori seg_train dibagi lagi menjadi set pelatihan (90%) dan set validasi (10%) secara acak namun terstratifikasi. Set validasi digunakan untuk memantau performa model selama pelatihan dan untuk melakukan *early stopping* atau penyesuaian *learning rate*. Direktori seg_test secara keseluruhan digunakan sebagai \ *test set* untuk evaluasi final.
2. **Pengubahan Ukuran (Resizing):** Semua gambar, baik dari set pelatihan, validasi, maupun uji, diubah ukurannya menjadi 224x224 piksel. Ukuran ini merupakan ukuran input standar yang umum digunakan untuk banyak arsitektur CNN *pre-trained*, termasuk MobileNetV3, dan memungkinkan pemanfaatan bobot *pre-trained* dari ImageNet.
3. **Normalisasi:** Nilai piksel gambar dinormalisasi menggunakan fungsi preprocess_input yang spesifik untuk model MobileNetV3. Ini adalah langkah krusial untuk memastikan input konsisten dengan apa yang diharapkan model dan membantu dalam stabilisasi proses pelatihan.
4. **Augmentasi Data (Data Augmentation):** Untuk set pelatihan, teknik augmentasi data diterapkan menggunakan ImageDataGenerator dari Keras. Augmentasi yang digunakan meliputi:
    - Rotasi acak (hingga 30 derajat)
    - Pergeseran lebar dan tinggi acak (hingga 20% dari dimensi)
    - Pergeseran shear acak (hingga 20%)
    - Zoom acak (hingga 20%)
    - Flip horizontal acak

Tujuan augmentasi data adalah untuk meningkatkan variasi data pelatihan secara artifisial, yang membantu model menjadi lebih robust terhadap variasi dalam data riil dan mengurangi risiko *overfitting,* contoh gambar setelah melewati proses augmentasi terlihat seperti pada Gambar 2. Augmentasi tidak diterapkan pada set validasi dan uji.

Gambar 2. Contoh Gambar Setelah Melewati Proses Augmentasi.

Sesuai dengan proposal awal, teknik pra-pemrosesan klasik seperti filter Gaussian, filter Median, dan *histogram equalization* (yang juga merupakan bentuk *contrast stretching*) dipertimbangkan. Fungsi untuk teknik-teknik ini dikembangkan. Namun, untuk pipeline utama dengan model MobileNetV3, fokus diberikan pada normalisasi standar dan augmentasi data yang umum digunakan dalam *deep learning*, karena CNN modern seringkali mampu belajar untuk mengatasi variasi noise dan pencahayaan tertentu, terutama dengan augmentasi.

4. Metode

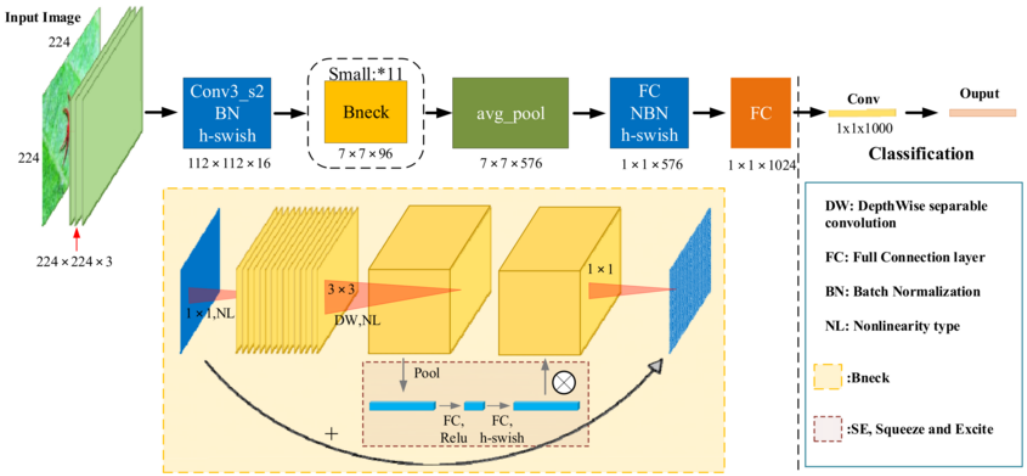
Pendekatan utama yang diadopsi dalam proyek ini adalah *deep learning* menggunakan Convolutional Neural Networks (CNNs) dengan strategi *transfer learning* untuk tugas klasifikasi pemandangan alam. Arsitektur MobileNetV3 dipilih sebagai model dasar karena reputasinya dalam memberikan keseimbangan yang baik antara akurasi dan efisiensi komputasional, sejalan dengan tujuan awal proposal untuk menghasilkan model yang efektif dan berpotensi ringan untuk aplikasi *real-time*.

4.1 Arsitektur Model: MobileNetV3

MobileNetV3 merupakan generasi ketiga dari keluarga MobileNet yang dirancang untuk efisiensi. Seperti yang terlihat pada gambar Gambar 3 arsitektur ini memperkenalkan beberapa peningkatan signifikan, termasuk:

- **Blok Inverted Residual dengan Linear Bottleneck:** Mirip dengan MobileNetV2, menggunakan proyeksi ke ruang dimensi rendah sebelum ekspansi dan kembali ke proyeksi dimensi rendah.
- **Depthwise Separable Convolutions:** Mengurangi jumlah parameter dan komputasi secara drastis dibandingkan konvolusi standar.
- **Squeeze-and-Excite (SE) Blocks:** Mengadaptasi bobot kanal secara dinamis untuk meningkatkan representasi fitur.

- **Platform-Aware Network Architecture Search (NAS):** Digunakan untuk menemukan arsitektur blok yang optimal.
- **Fungsi Aktivasi h-swish:** Modifikasi dari fungsi aktivasi swish yang lebih efisien secara komputasi.



Gambar 3. Arsitektur MobileNetV3.

Dua varian MobileNetV3 digunakan dalam proyek ini:

- **MobileNetV3 Small:** Versi yang lebih kecil dan lebih cepat, cocok untuk aplikasi dengan batasan sumber daya yang sangat ketat.
- **MobileNetV3 Large:** Versi yang lebih besar dengan kapasitas lebih tinggi, umumnya menghasilkan akurasi yang lebih baik dengan sedikit peningkatan biaya komputasi.

Kedua model dimuat dengan bobot yang telah dilatih sebelumnya (*pre-trained*) pada dataset ImageNet. Jumlah perbandingan total parameter antara MobileNetV3 *small* dan *large* dapat dilihat pada Tabel 1.

Tabel 2. Jumlah Parameter MobileNetV3Small dan MobileNetV3Large.

Model	Total Parameter
MobileNetV3Small	2,554,968
MobileNetV3Large	5,507,432

4.2 Transfer Learning Strategi *transfer learning* diterapkan dalam dua fase utama:

1. **Tahap 1: Baseline**
    - Lapisan-lapisan konvolusi dari *base model* MobileNetV3 (Small dan Large) dibekukan (*frozen*), sehingga bobotnya tidak diperbarui selama pelatihan awal.
    - *Classifier head* bawaan dari MobileNetV3 (yang dilatih untuk 1000 kelas ImageNet) dihilangkan dan digantikan dengan *head* baru yang sesuai untuk tugas klasifikasi 6 kelas pemandangan alam.

- Hanya bobot dari *classifier head* baru ini yang dilatih pada dataset pemandangan alam. Tahap ini memungkinkan *head* baru untuk belajar memetakan fitur-fitur yang diekstraksi oleh *base model pre-trained* ke kelas-kelas target baru.

2. **Tahap 2: Fine-tuning**
    - Setelah *head classifier* dilatih dan menunjukkan konvergensi yang wajar, sebagian atau seluruh lapisan dari *base model* MobileNetV3 di-*unfreeze* (dibuka sehingga bobotnya dapat diperbarui).
    - Model kemudian dilatih kembali pada dataset pemandangan alam dengan *learning rate* yang jauh lebih kecil daripada yang digunakan pada tahap ekstraksi fitur. Penggunaan *learning rate* yang kecil sangat penting selama *fine-tuning* untuk mencegah rusaknya bobot *pre-trained* yang sudah baik dan hanya melakukan penyesuaian halus agar lebih relevan dengan dataset spesifik.
    - Dalam proyek ini, seluruh *base model* di-*unfreeze* untuk memungkinkan adaptasi yang lebih komprehensif.

Pendekatan *transfer learning* ini dipilih karena dataset Intel Image Classification, meskipun cukup besar (sekitar 14.000 gambar latih), masih jauh lebih kecil dibandingkan ImageNet. *Transfer learning* memungkinkan kita memanfaatkan pengetahuan fitur visual umum yang telah dipelajari dari ImageNet dan mengadaptasinya untuk tugas spesifik pengenalan alam, yang seringkali menghasilkan performa yang lebih baik dan waktu pelatihan yang lebih cepat dibandingkan melatih model dari awal (*from scratch*).

4.3 Detail Pelatihan

- **Framework:** TensorFlow (versi 2.19.0) dengan API Keras dan Python (3.10) digunakan untuk membangun, melatih, dan mengevaluasi model.
- **Loss Function:** categorical_crossentropy dipilih sebagai fungsi kerugian, yang sesuai untuk tugas klasifikasi multi-kelas.
- **Optimizer:** Adam (Adaptive Moment Estimation) digunakan sebagai optimizer dengan *learning rate* awal yang berbeda untuk tahap ekstraksi fitur dan *fine-tuning*.
- **Metrics:** accuracy adalah metrik utama yang dipantau selama pelatihan. Metrik lain seperti precision, recall, dan F1-score dihitung pada tahap evaluasi.
- **Callbacks:** Beberapa *callback* Keras digunakan untuk mengelola proses pelatihan:
    - ModelCheckpoint: Menyimpan bobot model dengan performa terbaik (berdasarkan val_accuracy) selama pelatihan.
    - EarlyStopping: Menghentikan pelatihan jika tidak ada peningkatan signifikan pada val_accuracy setelah sejumlah epoch tertentu (patience), untuk mencegah *overfitting* dan menghemat waktu komputasi.

    - ReduceLROnPlateau: Mengurangi *learning rate* secara otomatis jika val_loss tidak membaik setelah beberapa epoch, membantu model keluar dari plato dan mencari minimum yang lebih baik.

- **Batch Size:** *Batch size* yang digunakan adalah 32.
- **Epoch:** *Epoch* yang digunakan adalah 50.

4.4 Interpretabilitas Model: Grad-CAM

Untuk mendapatkan wawasan tentang bagaimana model membuat keputusannya, teknik Gradient-weighted Class Activation Mapping (Grad-CAM) diimplementasikan. Grad-CAM menggunakan gradien dari kelas target yang mengalir ke lapisan konvolusi terakhir untuk menghasilkan *heatmap* lokalisasi kasar yang menyoroti daerah penting dalam citra input yang berkontribusi paling besar terhadap prediksi kelas tersebut. Ini membantu dalam:

- Memverifikasi apakah model memperhatikan fitur yang relevan secara semantik.
- Melakukan *debugging* kualitatif pada kasus-kasus di mana model salah klasifikasi.
- Meningkatkan kepercayaan pada keputusan model.

4.5 Alternatif yang Dipertimbangkan

Meskipun proposal awal menyebutkan komponen seperti ekstraksi fitur menggunakan analisis tekstur dan deskriptor bentuk, fokus proyek ini diarahkan pada pendekatan *deep learning SOTA* untuk "masalah pengenalan citra yang menantang". Arsitektur CNN lain seperti ResNet atau EfficientNet juga dipertimbangkan, namun MobileNetV3 dipilih karena keseimbangan yang sangat baik antara performa dan efisiensi, yang sejalan dengan potensi aplikasi *real-time* atau pada perangkat dengan sumber daya terbatas yang disebutkan dalam proposal awal.

5. Eksperimen

Bagian ini merinci pengaturan eksperimental, hasil yang diperoleh dari berbagai konfigurasi model, dan analisis terhadap hasil tersebut.

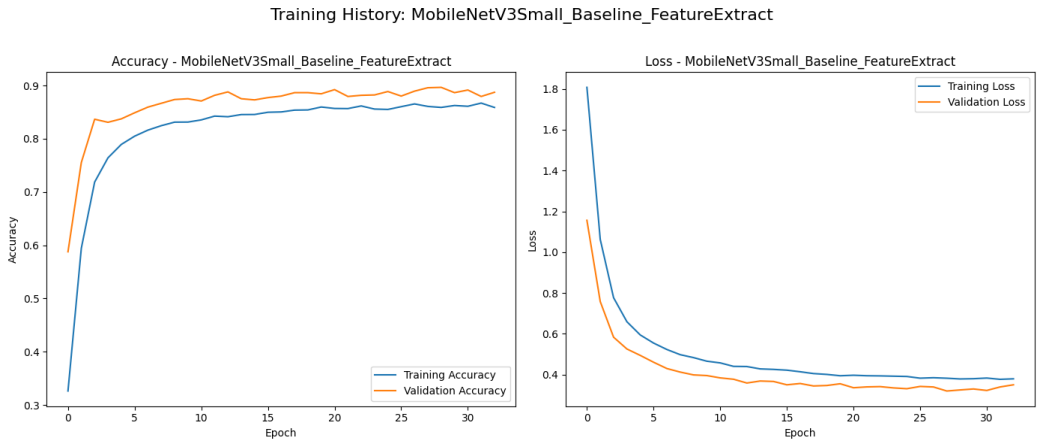
5.1 Pengaturan Eksperimental

- **Dataset:** Intel Image Classification, dibagi menjadi 90% data latih dan 10% data validasi dari direktori seg_train, dengan direktori seg_test digunakan sebagai set uji akhir yang independen.
- **Pra-pemrosesan:** Seperti yang dijelaskan di Bagian 3 (resize ke 224x224, normalisasi MobileNetV3, augmentasi data untuk set latih).
- **Perangkat Keras:** *(AMD Ryzen 8700F, GTX 1060 6GB, RAM 32GB)*.

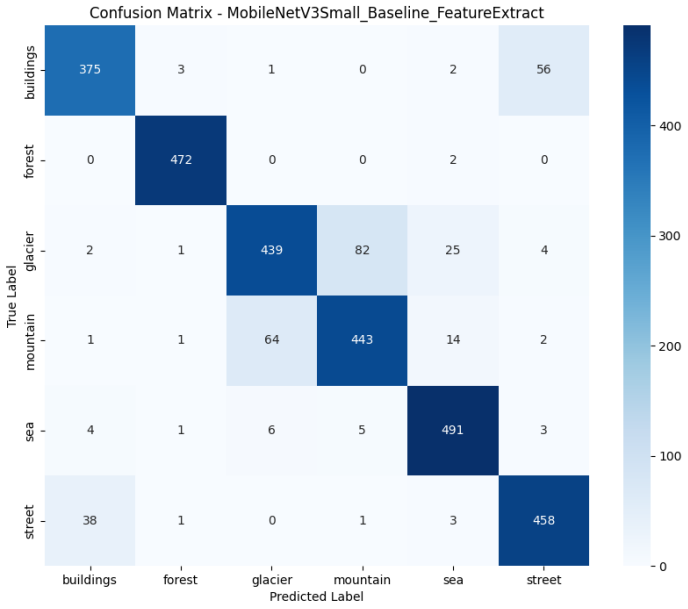
5.2 Eksperimen Baseline

Pada tahap ini, *base model* MobileNetV3 (Small dan Large) dibekukan, dan hanya *head classifier* baru yang dilatih. *Learning rate* yang digunakan untuk optimizer Adam adalah 1e-4.

- **MobileNetV3 Small (Baseline):**
  - Akurasi Tes: **89.27%**
  - Loss Tes: **0.2700**
  - Epoch terbaik pada validasi: Epoch ke-27, val_accuracy: 0.8894, val_loss: 0.3391.



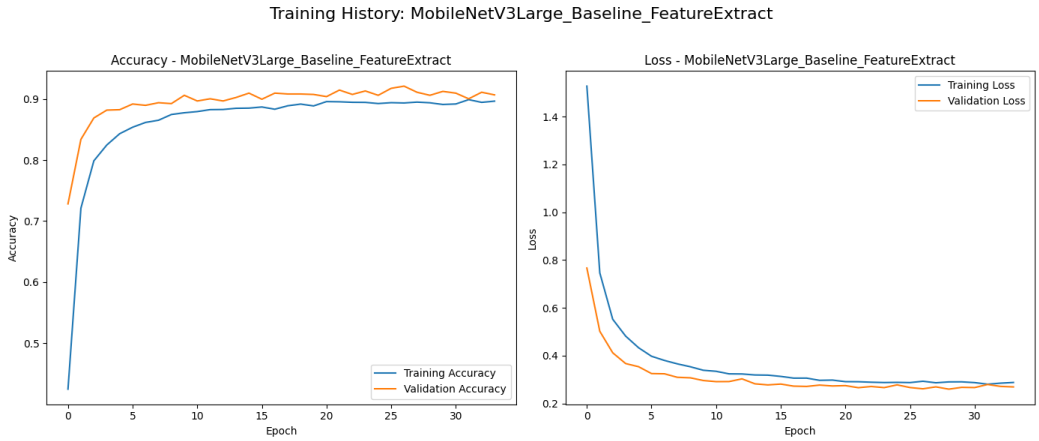
Gambar 4. Histori Pelatihan MobileNetV3(Baseline) Small



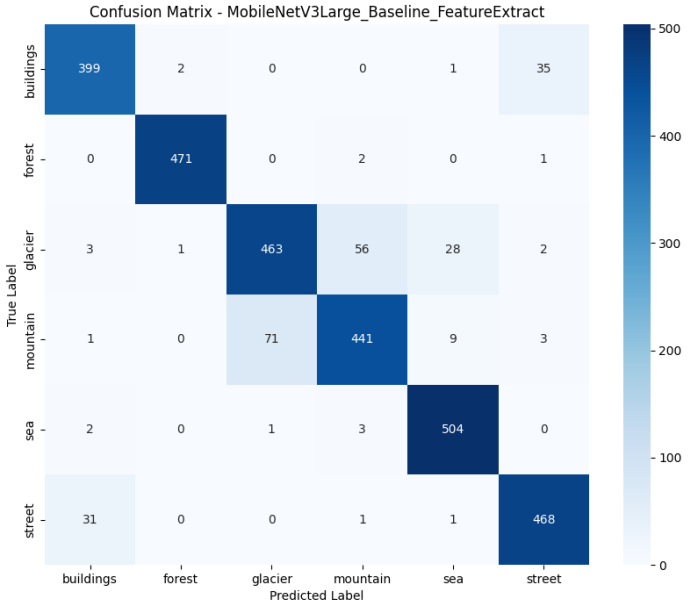
Gambar 5. Confusion Matrix MobileNetV3(Baseline) Small

- **MobileNetV3 Large (Baseline):**
  - Akurasi Tes: **91.53%**
  - Loss Tes: **0.2310**

- Epoch terbaik pada validasi (sebelumnya dilaporkan): Epoch ke-28, val_accuracy: 0.9108, val_loss: 0.2698.



Gambar 6. Histori Pelatihan MobileNetV3(Baseline) Large



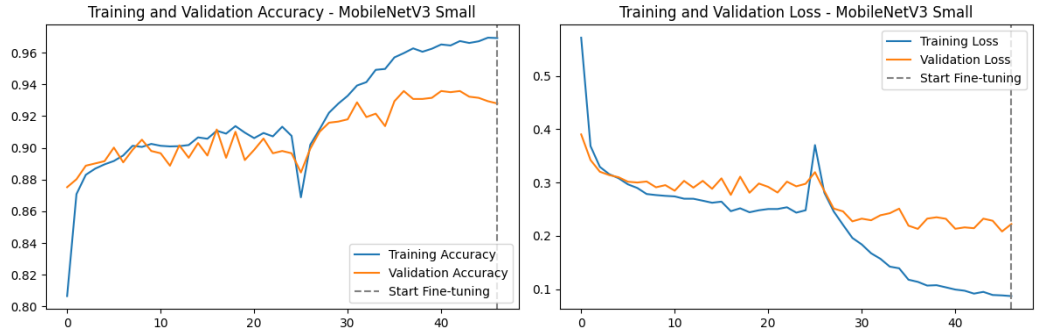
Gambar 7. Confusion Matrix MobileNetV3(Baseline) Large

Hasil baseline menunjukkan bahwa bahkan dengan hanya melatih *head classifier*, model *pre-trained* MobileNetV3 mampu mengekstrak fitur yang cukup representatif dari dataset pemandangan alam, dengan MobileNetV3 Large menunjukkan performa awal yang sedikit lebih baik.

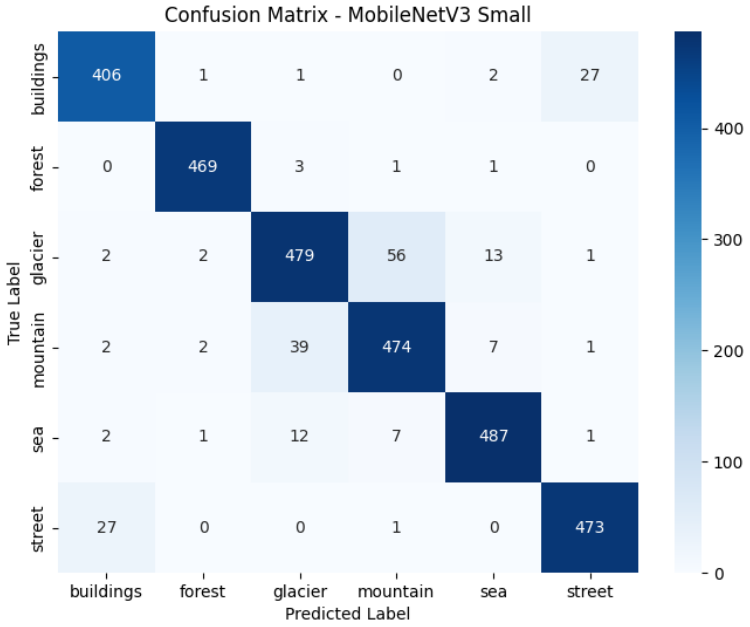
5.3 Eksperimen Fine-tuning

Setelah tahap ekstraksi fitur, seluruh lapisan *base model* di-*unfreeze*, dan model dilatih kembali dengan *learning rate* yang lebih kecil untuk *fine-tuning*. Bobot terbaik dari tahap baseline dimuat sebelum memulai *fine-tuning*.

- **MobileNetV3 Small (Fine-tuned):**
    - Akurasi Tes: **92.93%**
    - Loss Tes: **0.2106**
    - Epoch terakhir dilaporkan (training): Epoch ke-37, accuracy: 0.9605, loss: 0.1196.

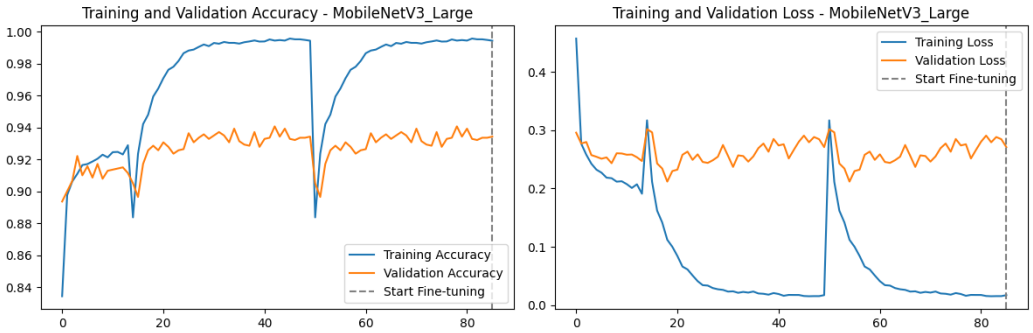


Gambar 8. Histori Pelatihan MobileNetV3(Finetune) Small

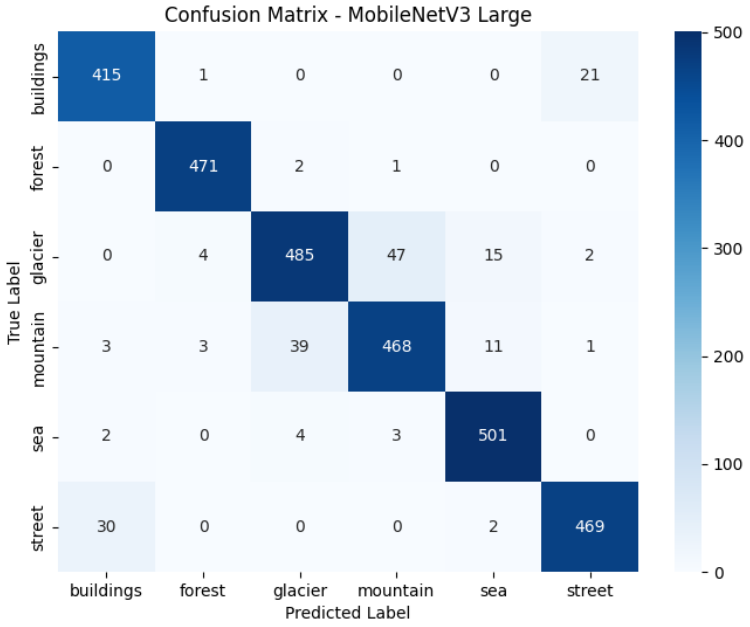


Gambar 9. Confusion Matrix MobileNetV3(Finetune) Small

- **MobileNetV3 Large (Fine-tuned):**
    - Akurasi Tes: **93.63%**
    - Loss Tes: **0.2376**
    - Epoch terakhir dilaporkan (training): Epoch ke-43, accuracy: 0.9944, loss: 0.0178.



Gambar 10. Histori Pelatihan MobileNetV3(Finetune) Large



Gambar 11. Confusion Matrix MobileNetV3(Finetune) Large

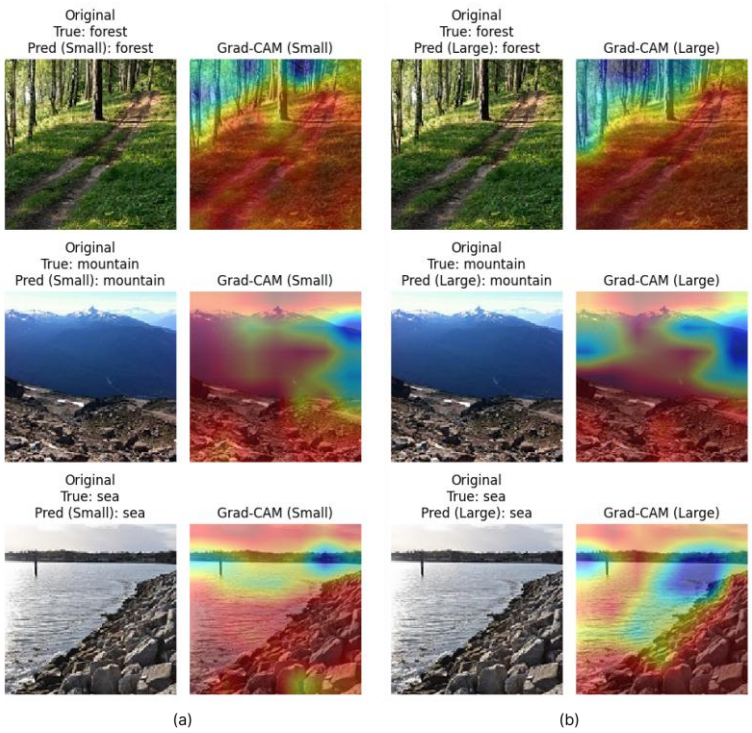
5.4 Analisis Hasil

Tabel berikut merangkum performa utama model pada *test set*:

Tabel 3. Rangkuman Performa Model Baseline dan Finetune

Model	Jenis	Test Accuracy	Test Loss
MobileNetV3 Small	Baseline	0.8927	0.2700
MobileNetV3 Large		0.9153	0.2310
MobileNetV3 Small	Finetune	0.9293	**0.2106**
MobileNetV3 Large		**0.9363**	0.2376

- **Dampak Fine-tuning:** Jelas terlihat bahwa *fine-tuning* secara signifikan meningkatkan akurasi untuk kedua arsitektur (peningkatan sekitar 3.66% untuk Small dan 2.1% untuk Large). Ini menunjukkan bahwa adaptasi bobot *pre-trained* pada lapisan-lapisan yang lebih dalam agar lebih sesuai dengan karakteristik dataset target sangat bermanfaat.
- **Perbandingan Small vs. Large:** MobileNetV3 Large secara konsisten mengungguli MobileNetV3 Small dalam hal akurasi, baik pada tahap baseline maupun setelah *fine-tuning*. Ini sesuai harapan karena MobileNetV3 Large memiliki kapasitas model yang lebih besar. Perbedaan akurasi setelah *fine-tuning* adalah sekitar 0.7%.
- **Loss vs. Akurasi:** Menariknya, meskipun MobileNetV3 Large (Fine-tuned) memiliki akurasi tertinggi, *test loss*-nya (0.2376) sedikit lebih tinggi daripada MobileNetV3 Small (Fine-tuned) (0.2106) dan bahkan sedikit lebih tinggi dari MobileNetV3 Large (Baseline) (0.2310). Ini bisa mengindikasikan bahwa model Large menjadi sangat percaya diri pada prediksinya yang benar, tetapi mungkin membuat kesalahan dengan margin yang lebih besar pada beberapa kasus, atau sedikit tanda *overfitting* meskipun akurasi tesnya tinggi.
- **Analisis Confusion Matrix:** Dari confusion matrix model terbaik yaitu MobileNetV3 Large (finetune), terlihat bahwa terdapat beberapa kelas yang salah klasifikasi seperti kelas glacier yang terprediksi mountain dan sebaliknya atau kelas buildings yang terprediksi street dan sebaliknya. Hal tersebut mungkin disebabkan oleh kemiripan visual dalam beberapa kondisi pencahayaan.
- **Perbandingan dengan metode lainnya:** Metode yang diusulkan yaitu MobileNetV3 Large Finetune dengan nilai akurasi test sebesar 0.9363 berhasil mengalahkan metode lainnya yaitu MobileNetV2 dengan nilai akurasi 0.900 [9], ResNet modifikasi dengan nilai akurasi 0.9020 [10], dan Vision Eagle Attention + ResNet-18 dengan nilai akurasi 0.9243 [11].
- **Visualisasi Grad-CAM:** Seperti yang terlihat pada Gambar 12 visualisasi menggunakan GradCam dapat memberikan pemahaman bagaimana model dapat melakukan klasifikasi terhadap gambar. Sebagai contoh, analisis gradcam menunjukkan bahwa untuk gambar ”forest”, model cenderung fokus pada jalanan yang memiliki rumput sedangkan untuk ”mountain” memerlukan untuk mengecek seluruh area gambar. MobileNetV3 Small dan Large menunjukan hasil yang mirip namun terkadang untuk versi large hanya membutuhkan sedikit area saja untuk menentukan gambar tersebut masuk ke kelas mana, dilihat dengan cara membandingkan jumlah intensitas warna merah antara small dan large.

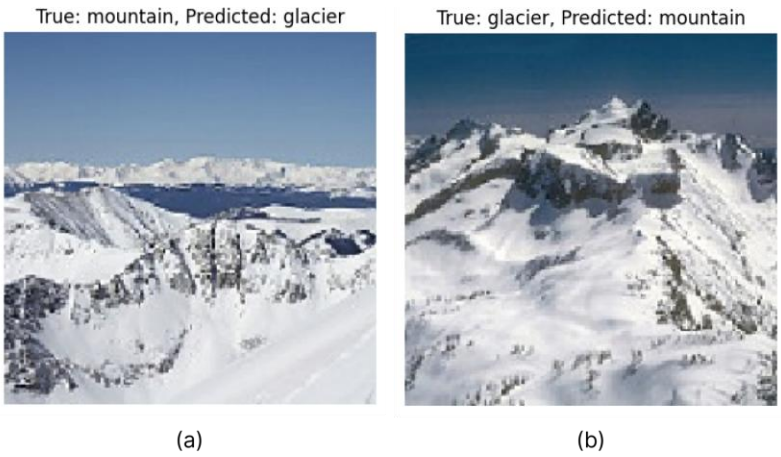


Gambar 12. Visualisasi GradCam: (a) MobileNetV3 Small dan (b) MobileNetV3 Large.

5.5 Potensi Kegagalan

Berdasarkan analisis kualitatif beberapa kegagalan yang mungkin terjadi meliputi:

- Gambar dengan pencahayaan yang sangat tidak biasa (terlalu gelap atau terlalu terang).
- Pemandangan yang ambigu yang dapat masuk ke lebih dari satu kategori misalnya, glacier bisa mirip dengan mountain seperti yang terlihat pada Gambar 13.
- Oklusi signifikan oleh objek yang tidak relevan.
- Sudut pandang atau komposisi yang tidak umum.



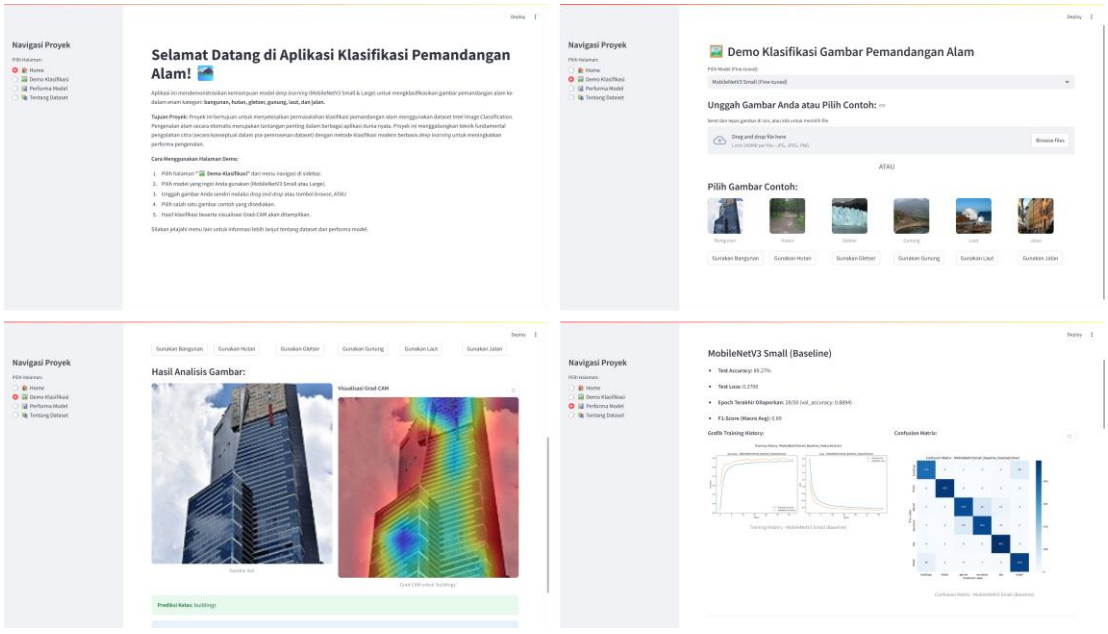
Gambar 13. Contoh Kesalahan Model Dalam Memprediksi Kelas.(a) memiliki nilai true mountain namun model memprediksi sebagai glacier dan (b) memiliki nilai true glacier namun model memprediksi sebagai mountain.

6. Kesimpulan

Proyek ini berhasil mengembangkan dan mengevaluasi sistem klasifikasi pemandangan alam menggunakan arsitektur MobileNetV3 dengan pendekatan *transfer learning*. Hasil eksperimen menunjukkan bahwa:

1. Teknik *fine-tuning* secara signifikan meningkatkan performa dibandingkan dengan hanya ekstraksi fitur, dengan peningkatan akurasi yang substansial untuk kedua varian MobileNetV3 (Small dan Large).
2. MobileNetV3 Large yang di-*fine-tune* mencapai akurasi tes tertinggi sebesar **93.63%**, menunjukkan kemampuannya untuk menangani tugas klasifikasi enam kelas pemandangan alam dengan baik.
3. MobileNetV3 Small yang di-*fine-tune* juga memberikan performa yang sangat kompetitif (akurasi tes 92.93%) dengan ukuran model dan kebutuhan komputasi yang lebih rendah, menjadikannya pilihan yang menarik untuk aplikasi dengan batasan sumber daya.
4. Visualisasi menggunakan Grad-CAM memberikan wawasan berharga mengenai daerah fokus model dalam membuat prediksi, yang berguna untuk analisis kualitatif dan membangun kepercayaan pada sistem.

Pembelajaran utama dari proyek ini meliputi pemahaman praktis tentang implementasi *transfer learning*, pentingnya strategi *fine-tuning* (terutama penggunaan *learning rate* kecil), evaluasi model yang komprehensif, dan teknik untuk menginterpretasikan keputusan model *deep learning*. Pengembangan aplikasi web interaktif menggunakan Streamlit juga memberikan pengalaman dalam mendemokan kemampuan model secara nyata. *User Interface* dari aplikasi web yang sudah dibuat dapat dilihat pada Gambar 14.



Gambar 14. User Interface Web Interaktif.

7. Saran Pengembangan Selanjutnya

Meskipun hasil yang dicapai sudah baik, ada beberapa area yang dapat dieksplorasi lebih lanjut untuk potensi peningkatan:

- **Hyperparameter Tuning yang Lebih Ekstensif:** Melakukan pencarian hyperparameter yang lebih sistematis (misalnya, menggunakan Grid Search, Random Search, atau teknik optimasi Bayesian) untuk *learning rate*, *batch size*, parameter optimizer, dan arsitektur *head classifier*.
- **Arsitektur Model Lain:** Mengeksplorasi arsitektur CNN SOTA lainnya seperti EfficientNetV2, ConvNeXt, atau bahkan Vision Transformers (ViT) untuk melihat apakah performa yang lebih tinggi dapat dicapai.
- **Teknik Augmentasi Data Lanjutan:** Menggunakan teknik augmentasi yang lebih canggih seperti CutMix, Mixup, atau augmentasi berbasis GAN.
- **Eksperimen Pra-pemrosesan Klasik:** Melakukan studi ablasi yang lebih mendalam mengenai dampak penerapan filter noise (Gaussian, Median) dan *histogram equalization* sebelum dimasukkan ke CNN, untuk melihat apakah ada kondisi tertentu di mana teknik ini memberikan manfaat.
- **Optimalisasi Model untuk Deployment:** Mengkonversi model terlatih ke format yang lebih ringan dan cepat seperti TensorFlow Lite atau ONNX untuk deployment yang lebih efisien pada perangkat *mobile* atau *edge*, serta melakukan kuantisasi model.
- **Perluasan Dataset:** Menambahkan lebih banyak data atau kategori pemandangan untuk meningkatkan generalisasi dan cakupan model.
- **Analisis Ensemble:** Menggabungkan prediksi dari beberapa model (misalnya, MobileNetV3 Small dan Large, atau dengan arsitektur lain) untuk potensi peningkatan akurasi.

Referensi

[1] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2015, pp. 3431–3440. doi: 10.1109/CVPR.2015.7298965.

[2] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2016, pp. 770–778. doi: 10.1109/CVPR.2016.90.

[3] C. Szegedy *et al.*, “Going deeper with convolutions,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2015, pp. 1–9. doi: 10.1109/CVPR.2015.7298594.

[4] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2016, pp. 770–778. doi: 10.1109/CVPR.2016.90.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, and Li Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Jun. 2009, pp. 248–255. doi: 10.1109/CVPR.2009.5206848.

[6] A. G. Howard *et al.*, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” Apr. 2017.

[7] A. Howard *et al.*, “Searching for MobileNetV3,” May 2019.

[8] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization,” Oct. 2016, doi: 10.1007/s11263-019-01228-7.

[9] S. Vats, J. P. Bhati, A. Singla, V. Kukreja, and R. Sharma, “Advanced Image Classification on Intel Datasets Using Optimized EfficientNet and MobileNetV2,” in *2024 IEEE 9th International Conference for Convergence in Technology (I2CT)*, IEEE, Apr. 2024, pp. 1–4. doi: 10.1109/I2CT61223.2024.10543649.

[10] A. A. Yahya, K. Liu, A. Hawbani, Y. Wang, and A. N. Hadi, “A Novel Image Classification Method Based on Residual Network, Inception, and Proposed Activation Function,” *Sensors*, vol. 23, no. 6, p. 2976, Mar. 2023, doi: 10.3390/s23062976.

[11] M. Hasan, “Vision Eagle Attention: a new lens for advancing image classification,” Nov. 2024.