



## Classification of Environmental Images Using VGGNet

### Klasifikasi Citra Lingkungan Menggunakan VGGNet

Tatia Shafwa Sinatria<sup>a,1,\*</sup>, Deanita Nur Fauzizah<sup>a,2</sup>, Devina Putri Yuliani<sup>a,3</sup>, Divia Prisillia Prisca<sup>a,4</sup>, Annita Fadhilah Aprilia<sup>a,5</sup>, Amandasari Dinda Rabbani<sup>a,6</sup>, Alfanz Rizaldy Pratama<sup>a,7</sup>

<sup>a</sup> Program Studi Sains Data, Universitas Pembangunan Nasional Veteran Jawa Timur, Surabaya, Indonesia

<sup>1</sup> 22083010001@studentupnjatim.ac.id \*; <sup>2</sup> 22083010005@studentupnjatim.ac.id; <sup>3</sup> 22083010028@studentupnjatim.ac.id;

<sup>4</sup> 22083010031@studentupnjatim.ac.id; <sup>5</sup> 22083010033@studentupnjatim.ac.id; <sup>6</sup> 22083010093@studentupnjatim.ac.id; <sup>7</sup> alfanz.fasilkom@upnjatim.ac.id

\* corresponding author

#### ARTICLE INFO

#### ABSTRAK/ABSTRACT

##### Article history

Received : November 1, 2025

Revised : December 25, 2025

Accepted : January 22, 2026

Published : February 5, 2026

Perkembangan teknologi pengolahan citra digital mendorong pemanfaatan metode *deep learning* untuk melakukan klasifikasi citra secara otomatis dan akurat. Penelitian ini bertujuan untuk membangun model klasifikasi citra lingkungan dengan memanfaatkan arsitektur VGG16 berbasis *transfer learning* dan *fine-tuning* menggunakan dataset *Intel Image Classification* yang terdiri dari enam kategori, yaitu *buildings*, *forest*, *glacier*, *mountain*, *sea*, dan *street*. Metode yang digunakan meliputi pengunduhan dataset melalui KaggleHub, tahap pra-pemrosesan citra berupa *resize*, normalisasi, dan augmentasi data, pembangunan model VGG16 dengan lapisan atas yang dimodifikasi, serta proses pelatihan menggunakan optimizer Adam dan fungsi *loss categorical cross-entropy*. Evaluasi kinerja model dilakukan menggunakan *confusion matrix* dan *classification report*. Hasil penelitian menunjukkan bahwa model mampu mencapai overall accuracy sebesar 90,16% pada data pengujian, dengan performa terbaik diperoleh pada kelas *forest*, sementara kinerja yang relatif lebih rendah terjadi pada kelas *glacier* dan *mountain* akibat adanya kemiripan karakteristik visual antar kelas. Visualisasi hasil prediksi menunjukkan bahwa model memiliki kemampuan generalisasi yang cukup baik terhadap citra yang tidak digunakan selama pelatihan. Temuan ini menegaskan bahwa penerapan *transfer learning* yang dikombinasikan dengan *fine-tuning* pada arsitektur VGG16 efektif digunakan untuk klasifikasi citra lingkungan dan layak dijadikan baseline dalam pengembangan sistem pemantauan berbasis citra.

*Advances in digital image processing technology have encouraged the use of deep learning methods to perform automatic and accurate image classification. This study aims to build an environmental image classification model using a VGG16 architecture based on transfer learning and fine-tuning using the Intel Image Classification dataset, which consists of six categories, namely buildings, forest, glacier, mountain, sea, and street. The methods used include downloading the dataset via KaggleHub, pre-processing images by resizing, normalizing, and augmenting data, building a VGG16 model with a modified top layer, and training the model using the Adam optimizer and categorical cross-entropy loss function. Model performance was evaluated using a confusion matrix and classification report. The results of the study show that the model was able to achieve an overall accuracy of 90.16% on the test data, with the best performance obtained in the forest class, while*

Kata Kunci: Penyakit Jantung; Imputasi Data; KNN Imputer; Random Forest; Machine Learning

Keywords: Heart disease; Data Imputation; KNN Imputer; Random Forest; Machine Learning



*relatively lower performance occurred in the glacier and mountain classes due to the similarity of visual characteristics between classes. Visualization of the prediction results shows that the model has fairly good generalization capabilities for images that were not used during training. These findings confirm that the application of transfer learning combined with fine-tuning on the VGG16 architecture is effective for environmental image classification and is suitable as a baseline in the development of image-based monitoring systems.*

## 1. Pendahuluan

Perubahan lingkungan yang dipengaruhi oleh urbanisasi, industrialisasi, dan perubahan iklim membuat kebutuhan pemantauan kondisi lingkungan menjadi semakin penting. Teknologi citra digital dan penginderaan jauh kini mampu menyediakan data visual dalam jumlah sangat besar dan detail. Namun, analisis citra secara manual tidak praktis karena membutuhkan waktu lama dan tenaga ahli, terutama ketika dataset mencapai ribuan hingga jutaan gambar. Kondisi inilah yang mendorong penggunaan metode otomatis berbasis deep learning, khususnya Convolutional Neural Network (CNN), yang mampu mengekstraksi pola visual secara lebih efektif. Selain VGG, terdapat arsitektur CNN lain yang sering digunakan sebagai pembandingan, seperti ResNet, Inception, MobileNet, dan EfficientNet. Berdasarkan studi Rozi dkk (2023) di Indonesia, ketiga arsitektur populer (VGG16, ResNet-50, Inception-V3) sama-sama mampu mengklasifikasikan citra dengan baik, tetapi memiliki karakteristik berbeda: ResNet unggul dalam stabilitas jaringan dalam tetapi butuh komputasi besar yaitu Inception menangkap fitur multi-skala namun strukturnya kompleks seperti MobileNet lebih ringan tetapi akurasi bisa turun pada detail kecil dan EfficientNet akurat namun membutuhkan proses tuning yang intensif. Oleh karena itu, pemilihan arsitektur harus mempertimbangkan ukuran dataset, jenis citra, serta kapasitas komputasi yang tersedia.

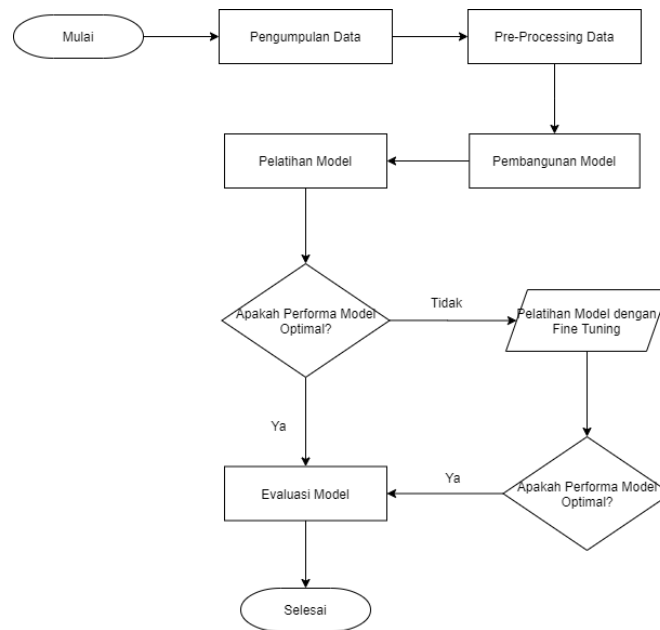
Di antara berbagai arsitektur tersebut, VGG-16 menjadi salah satu model yang paling banyak dipilih sebagai dasar (*baseline*) karena desainnya yang sederhana, terstruktur, dan mudah dimodifikasi. Blok konvolusi  $3 \times 3$  yang berulang membuat VGG-16 intuitif digunakan dan sangat cocok untuk transfer learning terutama ketika dataset terbatas. Studi Rozi dkk. (2023) menunjukkan bahwa dibanding ResNet-50 dan Inception-V3, VGG-16 justru menghasilkan akurasi tertinggi (97,78%) pada dataset citra daun di Indonesia, sehingga membuktikan bahwa meskipun arsitektur VGG lebih “lama”, kemampuannya mengekstraksi fitur dasar sampai detail masih sangat kuat. Hasil serupa ditunjukkan pada penelitian internasional oleh Naushad dkk. (2021), yang menegaskan bahwa transfer learning pada VGG-16 efektif untuk klasifikasi citra lingkungan (land use/land cover) ketika data terbatas, karena bobot pra-latih ImageNet dapat dipindahkan untuk memahami pola lingkungan seperti tekstur vegetasi, bangunan, dan struktur alam.

Namun, VGG-16 tetap memiliki kekurangan seperti jumlah parameter yang besar dan proses inferensi yang lebih lambat dibanding arsitektur ringan seperti MobileNet. Meski begitu, kekurangan ini dapat diatasi melalui strategi seperti augmentasi citra, regularisasi, dan fine-tuning pada beberapa layer akhir. Dengan pertimbangan bahwa model ini sederhana, mudah diimplementasikan, terbukti akurat pada studi lokal di Indonesia, dan mampu bekerja baik pada dataset lingkungan, VGG-16 menjadi pilihan tepat untuk penelitian ini.

Sebagai dasar penelitian ini, arsitektur VGG-16 akan diterapkan menggunakan pendekatan *transfer learning* untuk mengklasifikasikan citra lingkungan pada dataset *Intel Image Classification* (Kaggle), yang terdiri dari enam kategori yaitu *buildings, forest, glacier, mountain, sea, dan street*. Diharapkan model ini mampu memberikan hasil klasifikasi yang akurat dan dapat mendukung pengembangan sistem pemantauan lingkungan yang cepat, otomatis, dan berkelanjutan.

## 2. Metode Penelitian

Penelitian ini dilakukan melalui serangkaian tahapan yang sistematis untuk membangun model deep learning berbasis VGG16 dalam melakukan klasifikasi citra. Berikut adalah flowchart yang menjelaskan alur lengkap pada proses penelitian ini.



**Gambar 1.** Flowchart

### 2.1 Sumber Data

Dataset yang digunakan dalam penelitian ini adalah Intel Image Classification Dataset yang diperoleh melalui platform Kaggle menggunakan integrasi KaggleHub pada Google Colab. Dataset ini terdiri dari enam kelas citra lanskap, yaitu *buildings*, *forest*, *glacier*, *mountain*, *sea*, dan *street*. Pemilihan dataset ini dipertimbangkan karena struktur datanya yang baik serta relevan dengan kebutuhan analisis klasifikasi citra berbasis lingkungan.

### 2.2 Material

Material yang digunakan dalam penelitian ini meliputi perangkat lunak dan library pendukung, yaitu:

- 1) TensorFlow 2.15 (Google LLC) sebagai kerangka kerja utama untuk pembangunan model deep learning, Keras API (Google LLC) sebagai antarmuka pemodelan tinggi.
- 2) NumPy 1.26 (NumPy Developers) untuk pengolahan data numerik, Matplotlib 3.x (Matplotlib Community) untuk visualisasi hasil eksperimen,
- 3) KaggleHub API (Kaggle Inc.) yang digunakan dalam proses pengunduhan dataset. Dataset yang digunakan adalah *Intel Image Classification Dataset* (Kaggle Dataset Repository) yang terdiri dari enam kelas citra lanskap.

### 2.3 Instrumentasi

Instrumentasi yang digunakan dalam penelitian ini terdiri dari perangkat komputasi dan lingkungan pemrograman yang mendukung proses pelatihan model deep learning. Penelitian dijalankan menggunakan platform Google Colab yang dilengkapi dengan GPU NVIDIA Tesla T4 (Google Research) sebagai akselerator komputasi. Bahasa pemrograman yang digunakan adalah Python 3.10 (Python Software Foundation), dengan lingkungan kerja berbasis Jupyter Notebook pada Google Colab (Google LLC). Selain itu, model pelatihan VGG16 dari *Keras Applications* digunakan sebagai arsitektur dasar dalam proses *Transfer Learning*. Kombinasi instrumen tersebut

memastikan proses komputasi berjalan efisien serta mendukung eksperimen yang membutuhkan kemampuan pengolahan data berukuran besar.

#### 2.4 Prosedur

Prosedur penelitian ini disusun secara sistematis untuk membangun model klasifikasi citra berbasis transfer learning menggunakan arsitektur VGG16. Tahapan yang dilakukan adalah sebagai berikut:

##### 1) Persiapan Dataset

Dataset Intel Image Classification diunduh melalui integrasi KaggleHub pada Google Colab. Setelah diunduh, seluruh citra disusun kembali sesuai struktur direktori yang diperlukan oleh modul pemrosesan citra. Penataan folder ini dilakukan agar dataset dapat terbaca dengan baik pada proses pelatihan model.

##### 2) Pra-pemrosesan Citra

Tahap pra-pemrosesan citra yang meliputi pengubahan ukuran seluruh gambar menjadi  $224 \times 224$  piksel agar kompatibel dengan arsitektur VGG16. Nilai piksel kemudian dinormalisasi ke rentang 0–1 untuk menstabilkan proses pembelajaran. Selain itu, diterapkan augmentasi citra berupa rotasi, flipping, zooming, dan shifting guna meningkatkan keragaman data dan meminimalkan risiko overfitting. Setelah itu, dataset dibagi menjadi data pelatihan, data validasi, dan data pengujian.

##### 3) Inisialisasi Model Transfer Learning

Setelah proses pra-pemrosesan selesai, dilakukan pembangunan model menggunakan pendekatan transfer learning. Arsitektur VGG16 dimuat dengan bobot pra-latih dari ImageNet dan lapisan fully-connected pada bagian atas dihapus untuk digantikan dengan lapisan klasifikasi baru sesuai jumlah kelas pada dataset. Seluruh lapisan konvolusi kemudian dibekukan agar tidak mengalami pembaruan bobot pada tahap awal pelatihan.

##### 4) Penyusunan Lapisan Klasifikasi Baru

Lapisan Global Average Pooling ditambahkan untuk mereduksi fitur yang dihasilkan oleh basis VGG16, kemudian diikuti dengan lapisan dense pada bagian output yang berfungsi menghasilkan prediksi enam kategori citra. Penyusunan lapisan baru ini bertujuan menyesuaikan kemampuan ekstraksi fitur VGG16 dengan karakteristik data penelitian.

##### 5) Pelatihan Model

Model dilatih menggunakan data pelatihan dengan memanfaatkan *optimizer* Adam dan fungsi *loss categorical crossentropy*. Proses pelatihan dipantau menggunakan metrik akurasi dan nilai *loss* pada data validasi. Mekanisme *early stopping* diterapkan untuk mencegah *overfitting* ketika model tidak menunjukkan peningkatan performa. Setelah tahap pelatihan awal, dilakukan fine-tuning dengan membuka sebagian lapisan pada model pralatih (*pre-trained model*) agar jaringan dapat menyesuaikan fitur tingkat tinggi secara lebih spesifik terhadap data klasifikasi yang digunakan, sehingga diharapkan meningkatkan kinerja model secara keseluruhan.

##### 6) Evaluasi dan Analisis Hasil

Tahapan terakhir dilakukan dengan mengevaluasi performa model menggunakan data pengujian melalui akurasi, *confusion matrix*, dan tabel classification report. Hasil evaluasi kemudian dianalisis untuk mengidentifikasi pola kesalahan, terutama pada kelas dengan kemiripan visual, sehingga memberikan gambaran mengenai performa model serta potensi pengembangan lebih lanjut.

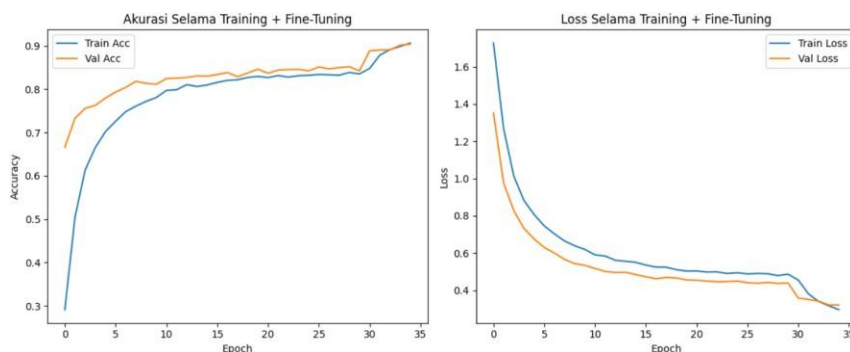
### 3. Hasil dan Pembahasan

Pada penelitian ini menghasilkan model klasifikasi citra yang menggunakan arsitektur VGG16 dengan berbasis transfer learning. Pada dataset Intel Image Classification yang terdiri dari enam kelas citra, yaitu *bangunan*, *hutan*, *pemandangan*, *gunung*, *laut*, dan *jalan*. Seluruh citra telah berhasil diproses menjadi ukuran  $224 \times 224$  piksel yang kemudian akan dibagi ke dalam data latih dan validasi data



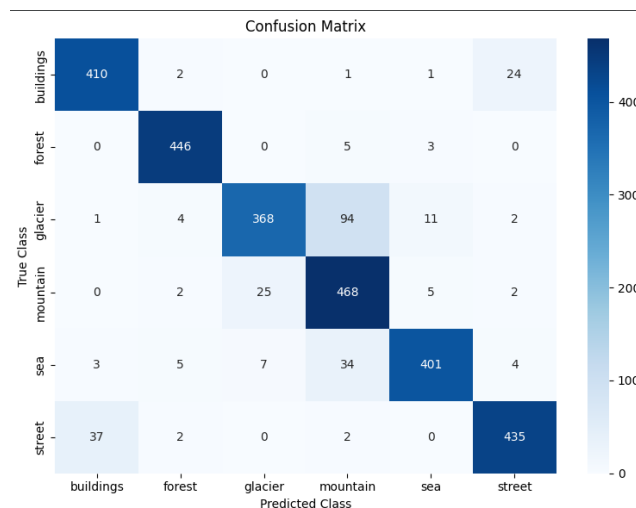
**Gambar 2.** Dataset *Intel Image Classification*

Proses pelatihan model akan dilakukan menggunakan optimizer Adam, fungsi *loss categorical\_crossentropy*, dan *batch size* 32. Mekanisme *Early Stopping* akan aktif selama pelatihan dan menghentikan proses ketika tidak terjadi peningkatan *validation loss*. Selama pelatihan, nilai *training loss* dan *validation loss* menunjukkan penurunan stabil, sedangkan nilai akurasi pelatihan dan akurasi validasi mengalami peningkatan pada setiap epoch. Hal ini menunjukkan bahwa model mampu mempelajari fitur secara efektif.



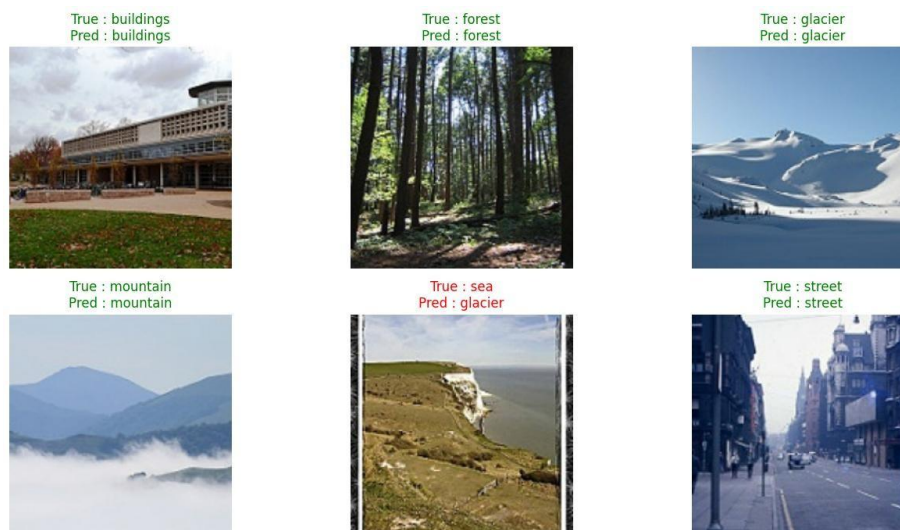
**Gambar 3.** Validasi Data

Model kemudian akan diuji menggunakan validasi data. Prediksi probabilitas diubah menjadi kelas akhir menggunakan *argmax*, kemudian dibandingkan dengan label sebenarnya. Evaluasi kinerja dilakukan menggunakan *Confusion Matrix*, yang menggambarkan jumlah prediksi yang benar dan salah untuk masing-masing kelas. Hasil evaluasi menunjukkan bahwa sebagian besar kelas dapat dikenali dengan cukup baik, terutama kelas yang punya ciri visual yang jelas. Namun, masih ditemukan beberapa kesalahan klasifikasi pada kelas yang memiliki pola visual yang mirip, seperti antara *sea* dengan *glacier*.



**Gambar 4.** Confusion Matrix

Dari visualisasi satu batch data validasi, nampaknya model berhasil memprediksi kesesuaian dengan sebagian besar label aslinya. Ini menunjukkan bahwa model memiliki kemampuan generalisasi yang cukup baik, bahkan untuk fitur-fitur yang tidak dipakai langsung saat proses pelatihan.



**Tabel 1.** Classification Report

	Precision	Recall	F1-score	Support
Buildings	0.90	0.93	0.92	438
Forest	0.96	0.98	0.97	454
Glacier	0.92	0.76	0.83	480
Mountain	0.77	0.93	0.84	502
Sea	0.95	0.88	0.91	454
Street	0.93	0.91	0.92	476
Accuracy			0.90	2804
Macro avg	0.90	0.90	0.90	2804
Weighted avg	0.90	0.90	0.90	2804
Overall Accuracy : 0.9016				

Berdasarkan hasil evaluasi menggunakan *classification report* dan visualisasi prediksi, model VGG16 berbasis *transfer learning* menunjukkan performa klasifikasi yang cukup baik. Nilai overall accuracy sebesar 90,16% menandakan bahwa sebagian besar citra pada data validasi berhasil diklasifikasikan dengan benar. Hal ini mengindikasikan bahwa bobot dalam pelatihan VGG16 mampu mengekstraksi fitur visual penting secara efektif pada dataset yang digunakan. Jika ditinjau per kelas, kelas hutan (forest) memiliki performa paling tinggi dengan nilai *precision* 0,96, *recall* 0,98, dan *F1-score* 0,97. Hasil ini sejalan dengan visualisasi prediksi, di mana citra hutan dapat dikenali dengan sangat baik oleh model. Karakteristik visual yang kuat dan konsisten, seperti dominasi warna hijau serta pola vegetasi yang jelas, memudahkan model dalam membedakan kelas ini dari kelas lainnya. Kelas bangunan (buildings) dan jalan (street) juga menunjukkan performa yang stabil dengan nilai *F1-score* di atas 0,92. Pada visualisasi, kedua kelas ini umumnya dapat diprediksi dengan benar karena memiliki struktur visual yang khas, seperti bentuk geometris bangunan dan pola jalan perkotaan yang relatif mudah dikenali oleh model.

Sebaliknya, performa pada kelas gletser (glacier) dan gunung (mountain) cenderung lebih rendah dibandingkan kelas lainnya. Kelas gletser memiliki *recall* sebesar 0,76, sedangkan kelas gunung memiliki *precision* yang relatif rendah, yaitu 0,77. Kondisi ini menunjukkan adanya kesalahan klasifikasi antara kedua kelas tersebut. Hal ini juga terlihat pada visualisasi prediksi, di mana salah satu citra dengan label asli *sea* diprediksi sebagai *glacier*. Kesalahan ini terjadi karena kemiripan visual antar kelas, khususnya pada citra dengan warna dominan biru atau putih, seperti laut, gletser, dan area pegunungan bersalju. Kemiripan tekstur dan warna tersebut membuat model sulit membedakan batas antar kelas secara tegas. Meskipun demikian, tingkat kesalahan yang terjadi masih tergolong wajar mengingat kompleksitas visual antar kelas yang cukup tinggi. Selain itu, nilai *macro average* dan *weighted average F1-score* yang sama-sama berada di angka 0,90 menunjukkan bahwa performa model relatif seimbang di seluruh kelas dan tidak terlalu bias terhadap kelas tertentu. Penerapan *fine-tuning* juga berkontribusi dalam meningkatkan kinerja model. Dengan membuka beberapa layer terakhir dari VGG16 dan melatihnya kembali menggunakan *learning rate* yang lebih kecil, model dapat menyesuaikan fitur tingkat tinggi agar lebih relevan dengan karakteristik dataset lingkungan. Pendekatan ini membuat representasi fitur menjadi lebih optimal dibandingkan hanya menggunakan bobot dalam pelatihan tanpa penyesuaian. Hasil visualisasi prediksi memperkuat temuan ini, di mana sebagian besar citra dapat dikenali dengan benar meskipun memiliki variasi pencahayaan dan sudut pengambilan gambar. Dengan demikian, dapat disimpulkan bahwa model VGG16 yang dibangun dalam penelitian ini berhasil mencapai performa klasifikasi yang baik dan layak digunakan sebagai baseline untuk tugas klasifikasi citra pada dataset lingkungan yang serupa. Meski begitu, performanya masih bisa ditingkatkan lagi, misalnya dengan penggunaan rumus ataupun persamaan sebagai berikut :

#### a. Konvolusi

Operasi konvolusi merupakan inti dari *Convolutional Neural Network* (CNN), di mana sebuah kernel (filter) digeser melintasi citra input untuk mengekstraksi fitur-fitur lokal seperti tepi, sudut, atau tekstur. Secara matematis, nilai piksel pada peta fitur keluaran pada posisi  $(i, j)$  dihitung dengan menjumlahkan hasil perkalian elemen per elemen antara kernel  $K$  dan *patch* citra input  $X$  yang terletak di sekitar posisi tersebut. Rumus di mana ukuran kernel  $k$  dan posisi input memengaruhi nilai keluaran pada peta fitur. Operasi ini memungkinkan jaringan untuk belajar representasi fitur visual yang penting saat proses pelatihan jaringan CNN[7]. Berikut rumus dari konvolusi:

$$Y(i, j) = \sum_{m=0}^{k-l} \sum_{n=0}^{k-l} X(i+m, j+n) K(m, n) \quad (1)$$

Keterangan:

- X : Citra input (patch gambar)
- K : Kernel / filter konvolusi

Y : Feature map hasil konvolusi  
 I,j : Posisi pixel pada output  
 M,n : Indeks elemen kernel  
 K : Ukuran kernel (misal 3×3)

#### b. ReLU Activation

Dalam penelitian klasifikasi citra yang menggunakan arsitektur *Convolutional Neural Network* seperti VGG-16, banyak peneliti menerapkan fungsi aktivasi ReLU (Rectified Linear Unit) pada lapisan tersembunyi untuk mengenalkan non-linearitas pada model. ReLU mengubah semua nilai negatif menjadi nol dan mempertahankan nilai positif, sehingga membantu mempercepat proses pelatihan dan mengurangi masalah gradien yang menghilang (*vanishing gradient*) dalam jaringan deep learning. Penerapan fungsi aktivasi ini dijelaskan eksplisit pada studi klasifikasi citra daun dengan VGG-16, di mana fungsi ReLU digunakan untuk setiap lapisan konvolusi dan fully-connected sebelum lapisan keluaran[10].

$$ReLU(x) = \max(0, x) \quad (2)$$

Keterangan:

X : Nilai input dari layer sebelumnya  
 ReLU(x) : Nilai output setelah aktivasi ReLU

ReLU digunakan pada lapisan Dense tambahan sebagai fungsi aktivasi untuk memberikan non-linearitas sehingga jaringan dapat mempelajari pola yang kompleks pada citra. Fungsi ini mempertahankan nilai positif dan menekan nilai negatif menjadi 0, sehingga mempercepat proses training pada CNN.

#### c. Regularisasi L2

Untuk mencegah *overfitting* pada model neural network, teknik regularisasi seperti L2 regularization sering diterapkan pada bobot jaringan. Regularisasi L2 menambahkan penalti terhadap besaran bobot yang besar sehingga model menjadi lebih general terhadap data baru. Dalam beberapa penelitian implementasi CNN, meskipun tidak selalu dijelaskan rumusnya secara eksplisit, penggunaan regularisasi L2 disebut sebagai bagian dari konfigurasi lapisan dense untuk meningkatkan performa model klasifikasi citra[8].

$$R(w) = \frac{\lambda}{2} \sum w^2 \quad (3)$$

Keterangan :

$\lambda$  : Koefisien regularisasi yang mengontrol seberapa kuat penalti diberikan  
 Nilai  $\lambda$  kecil  $\rightarrow$  penalti ringan; nilai besar  $\rightarrow$  penalti kuat.

$\sum w^2$  : Jumlah kuadrat seluruh bobot  $w$  dalam jaringan

Faktor  $\frac{1}{2}$  digunakan untuk mempermudah proses turunan pada optimisasi. Regularisasi L2 bekerja dengan menekan bobot menuju nilai kecil tetapi tidak nol, menjaga model lebih sederhana dan menghindari belajar pola-pola yang terlalu spesifik pada data pelatihan. Teknik ini secara luas digunakan pada CNN modern, termasuk VGGNet.

#### d. Akurasi

Sebagai metrik evaluasi kinerja model klasifikasi citra, akurasi dihitung sebagai perbandingan antara jumlah prediksi yang benar terhadap total jumlah sampel yang dievaluasi. Metrik ini sering dilaporkan dalam hasil uji model CNN/VGG16 untuk menunjukkan seberapa baik model dapat mengkategorikan citra sesuai label yang benar. Evaluasi akurasi dijadikan indikator utama dalam banyak penelitian klasifikasi citra berbasis deep learning yang dipublikasikan antara tahun 2021 hingga 2025[9].

$$\text{akurasi} = \frac{\text{prediksi benar}}{\text{total sampel}} \quad (4)$$

Keterangan :

Jumlah prediksi benar : Jumlah sampel yang diprediksi dengan label yang tepat oleh model

Total sampel : Seluruh jumlah data yang dievaluasi (baik data validasi maupun data uji)

Akurasi menunjukkan proporsi prediksi yang benar dari total prediksi yang dibuat. Nilainya berada antara 0 dan 1 (atau 0%-100%). Semakin tinggi nilai akurasi, semakin baik model dalam mengklasifikasikan data dengan benar.

#### e. Softmax (Lapisan Output)

Untuk menghasilkan distribusi probabilitas pada output klasifikasi multi-kelas, model CNN menggunakan fungsi aktivasi Softmax pada lapisan terakhir. Fungsi Softmax mengubah skor logit menjadi probabilitas setiap kelas sehingga jumlah total semua kelas bernilai 1. Implementasi Softmax paling sering ditemukan dalam penelitian klasifikasi citra berbasis CNN dengan transfer learning, yang menempatkan fungsi ini pada tahap akhir sebelum klasifikasi akhir. Penerapan Softmax pada *fully-connected layer* akhir dalam studi-studi tersebut menunjukkan peran pentingnya dalam menghasilkan prediksi kelas yang terinterpretasi sebagai probabilitas[8].

$$\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad (5)$$

Keterangan :

$Z_i$  : Nilai logit kelas ke-i

E : Bilangan Euler (2.718...)

K : Jumlah kelas

$\text{softmax}(z_i)$  : Probabilitas kelas ke-i

Softmax digunakan pada lapisan output untuk klasifikasi multi-kelas. Fungsi ini mengonversi skor logit menjadi probabilitas sehingga setiap nilai output menunjukkan peluang model bahwa citra termasuk ke kelas tertentu.

#### f. Categorical Cross-Entropy Loss

Dalam pelatihan model *Convolutional Neural Network* untuk klasifikasi citra, fungsi categorical cross-entropy digunakan sebagai fungsi loss utama untuk mengukur seberapa jauh hasil prediksi model menyimpang dari label yang benar. Fungsi ini secara matematis menghitung kesalahan prediksi probabilitas dibandingkan dengan target one-hot vector, sehingga memandu proses optimasi model melalui *backpropagation*. Banyak artikel penelitian deep learning pada dataset citra melaporkan penggunaan categorical cross-entropy karena kemampuannya menghasilkan gradien yang informatif untuk jaringan multi-kelas[9].

Keterangan:

L : Nilai loss

K : Jumlah kelas

$Y_i$  : Label benar (one-hot)

$\hat{y}_i$  : Probabilitas prediksi softmax

$\log$  : Logaritma natural

Loss ini menghitung selisih antara label sebenarnya (one-hot) dan probabilitas prediksi. Model belajar dengan meminimalkan nilai loss sehingga prediksi mendekati label yang benar.

## 4. Kesimpulan

Penelitian ini berhasil membangun model klasifikasi citra lingkungan berbasis arsitektur VGG16 dengan menerapkan metode *transfer learning* dan *fine-tuning* pada dataset *Intel Image Classification*, yang menghasilkan *overall accuracy* sebesar 90,16%. Hasil ini menunjukkan bahwa pengetahuan awal yang dimiliki VGG16 dapat dimanfaatkan secara efektif untuk mengekstraksi fitur visual penting tanpa perlu melatih model dari awal. Evaluasi per kelas memperlihatkan performa

terbaik pada kelas *forest* karena karakteristik visualnya yang kuat dan konsisten, sementara kesalahan klasifikasi lebih sering terjadi pada kelas *glacier* dan *mountain* akibat kemiripan pola visual, terutama pada citra bersalju. Penerapan *fine-tuning* dengan *learning rate* kecil membantu meningkatkan kesesuaian fitur tingkat tinggi terhadap karakteristik dataset, meskipun perbedaan kinerja antara data pelatihan dan data evaluasi mengindikasikan adanya kecenderungan *overfitting* ringan sehingga kemampuan generalisasi model masih dapat ditingkatkan. Selain itu, hasil visualisasi menunjukkan bahwa model belum sepenuhnya optimal dalam menangani variasi warna, intensitas cahaya, dan kemiripan spektrum antar kelas. Secara keseluruhan, model ini dapat dinyatakan berhasil dan layak digunakan sebagai *baseline* untuk klasifikasi citra lingkungan, dengan potensi peningkatan performa melalui *data augmentation* berbasis warna, regularisasi tambahan, dan eksplorasi arsitektur CNN yang lebih efisien.

## 5. Ucapan Terima Kasih

Penulis berterima kasih kepada semua pihak yang telah membantu, baik secara langsung maupun tidak langsung, dalam penyusunan jurnal ini. Dukungan, masukan, dan kerja sama yang diberikan sangat berarti bagi kelancaran penelitian ini.

## 6. Referensi

- [1] A. S. Tanenbaum dan M. Van Steen, *Distributed Systems: Principles and Paradigms*, 2nd ed. Upper Saddle River, NJ: Prentice Hall, 2022.
- [2] A. Verma, P. Singh, dan R. K. Sharma, "Efficient power management in wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 11, no. 5, pp. 1234–1243, May 2023.
- [3] D. L. Johnson, "Smart grid technology and applications," dalam *Advances in Smart Grid Technology*, T. Smith, Ed., 2nd ed. New York: McGraw-Hill, 2019, pp. 203–215.
- [4] M. Patel, "Emerging technologies in mobile computing," dalam *Proc. 15th Int. Conf. on Mobile Computing and Networking*, Chicago, IL, USA, 2019, pp. 45–50.
- [5] S. R. Jones, "Study of machine learning algorithms," Tesis, Universitas Oxford, Oxford, Inggris, 2017.
- [6] National Aeronautics and Space Administration (NASA), "Artemis program: Returning humans to the moon," NASA. [Online]. Available: <https://www.nasa.gov/specials/artemis/>. [Accessed: 3 May 2024].
- [7] I. Goodfellow, Y. Bengio, dan A. Courville, *Deep Learning*. Cambridge, MA: MIT Press, 2016.
- [8] M. D. Darajat, Y. A. Sari, dan R. C. Wihandika, "Convolutional Neural Network untuk klasifikasi citra makanan khas Indonesia," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 5, no. 11, pp. 4764–4769, 2021.
- [9] I. R. Ramadhani, A. Nilogiri, dan Q. A'yun, "Klasifikasi jenis tumbuhan berdasarkan citra daun menggunakan metode Convolutional Neural Network," *Jurnal Smart Teknologi*, vol. 3, no. 3, pp. 249–260, 2022.
- [10] M. Akbar, P. T. Prasetyaningrum, P. W. Setyaningsih, dan M. Ahsan, "Deteksi leukemia limfoblastik akut menggunakan Convolutional Neural Network," *J. Teknol. Sist. Inf. dan Apl.*, vol. 7, no. 1, pp. 292–297, 2024.
- [11] G. Statnikov, S. Karthik, Y. Boumhaout, dan A. Elfikky, "A comprehensive survey of convolutions in CNNs: Evolution, architectures, and applications," *Preprints.org*, pp. 0–31, 2025.