

Analisis Komparatif Model MobilenetV1 Dan EfficientnetB0 Dalam Klasifikasi Citra Empat Musim Menggunakan Transfer Learning

Muhammad Iqbal Burhanuddin¹, Adam Syaifullah², Setiawan Adeka Putra Jaya³, Muhammad Gabriel Somoal⁴

^{1,2,3,4}Universitas Muhammadiyah Prof. Dr. HAMKA

e-mail: 1miqbalb26@gmail.com, 2adamsyaifullah49@gmail.com, 3setiawanadekapj@gmail.com,
4gabrielsomoal@gmail.com

Abstract – Seasons influence various aspects of the environment and human activities such as agriculture, tourism, and transportation. Automatic identification of seasons based on visual imagery offers a potential solution to support image-based monitoring systems. However, classifying images into four seasonal categories—autumn, spring, summer, and winter—has rarely been implemented in deep learning-based recognition systems. This study aims to compare two lightweight Convolutional Neural Network (CNN) architectures, namely MobileNetV1 and EfficientNetB0, to determine the most optimal model for seasonal image classification. Both models were implemented using a transfer learning approach and optimized with the Adam (Adaptive Moment Estimation) Optimizer. The dataset consists of 2,000 images, divided into 75% training data, 15% validation data, and 10% testing data. Evaluation results show that MobileNetV1 achieved the best performance with 95.5% accuracy, 95.5% precision, 95.5% recall, and 95.5% F1-Score, while EfficientNetB0 only reached 91.5% accuracy. This comparison indicates that MobileNetV1 is superior in terms of efficiency and accuracy for the task of seasonal visual classification. This study contributes to the development of season-based image classification systems that can be applied in website applications or automated visual environmental monitoring systems.

Keywords – Transfer Learning, MobileNetV1, EfficientNetB0, Season Image, Convolutional Neural Network, Adam Optimizer

Abstrak – Musim memengaruhi berbagai aspek lingkungan dan aktivitas manusia seperti pertanian, pariwisata, serta transportasi. Identifikasi otomatis terhadap musim berdasarkan citra visual menjadi solusi potensial untuk mendukung sistem monitoring berbasis teknologi. Namun, klasifikasi citra ke dalam empat kategori musim, yaitu *autumn*, *spring*, *summer*, dan *winter*, masih jarang diimplementasikan dalam sistem pengenalan berbasis *deep learning*. Penelitian ini bertujuan untuk membandingkan dua arsitektur ringan *Convolutional Neural Network* (CNN), yaitu *MobileNetV1* dan *EfficientNetB0*, untuk menentukan model terbaik dalam klasifikasi citra musim. Keduanya diimplementasikan menggunakan pendekatan *transfer learning* dan dioptimasi dengan *Adam (Adaptive Moment Estimation) Optimizer*. Dataset terdiri atas 2000 citra yang dibagi menjadi 75% data *train*, 15% data *validation*, dan 10% data *testing*. Hasil evaluasi menunjukkan bahwa *MobileNetV1* memberikan performa terbaik dengan *accuracy* 95,5%, *precision* 95,5%, *recall* 95,5%, dan *F1-Score* 95,5%, sedangkan *EfficientNetB0* hanya mencapai *accuracy* sebesar 91,5%. Perbandingan ini mengindikasikan bahwa *MobileNetV1* lebih unggul dalam efisiensi dan akurasi pada tugas klasifikasi visual musim. Penelitian ini memberikan kontribusi terhadap pengembangan sistem klasifikasi musim berbasis citra yang dapat diimplementasikan dalam aplikasi *website* atau sistem monitoring lingkungan berbasis citra otomatis.

Kata Kunci – Transfer Learning, MobileNetV1, EfficientNetB0, Citra Musim, Convolutional Neural Network, Adam Optimizer

I. PENDAHULUAN

Perubahan musim tidak hanya berdampak pada fenomena alam, tetapi juga memengaruhi aktivitas manusia seperti pertanian, transportasi, dan pariwisata. Setiap musim memiliki karakteristik visual yang khas: daun yang menguning dan berguguran pada musim gugur, bunga-bunga yang bermekaran penuh warna di musim semi, pencahayaan intens dan langit cerah di musim panas, hingga hamparan salju yang mendominasi pemandangan saat musim dingin. Identifikasi otomatis terhadap musim melalui citra visual sangat bermanfaat dalam berbagai aplikasi, mulai dari sistem monitoring cuaca hingga sistem informasi geografis berbasis citra [1].

Musim memiliki dampak signifikan terhadap perubahan lingkungan dan memengaruhi berbagai sektor kehidupan manusia, seperti pertanian, transportasi, pariwisata, hingga aktivitas harian masyarakat. Setiap musim memiliki karakteristik visual yang khas: dedaunan yang berguguran dan menguning pada musim gugur (*autumn*), bunga-bunga bermekaran penuh warna di musim semi (*spring*), pencahayaan intens dan langit cerah saat musim panas (*summer*), serta dominasi salju dan warna pucat pada musim dingin. Identifikasi otomatis terhadap musim melalui citra visual sangat bermanfaat dalam berbagai aplikasi, mulai dari sistem monitoring cuaca hingga sistem informasi geografis berbasis citra [1].

Penelitian mengenai klasifikasi musim menggunakan pendekatan berbasis pengenalan citra masih relatif terbatas, khususnya dalam konteks empat musim utama: *autumn*, *spring*, *summer*, dan *winter*. Sebagian besar studi yang ada lebih berfokus pada klasifikasi kondisi cuaca seperti cerah, berawan, hujan, atau berkabut [2][3]. Hingga saat ini, belum banyak penelitian yang secara spesifik mengkaji klasifikasi citra visual empat musim dalam satu sistem pengenalan otomatis berbasis CNN. Oleh karena itu, eksplorasi metode *deep learning* seperti *Convolutional Neural Network* (CNN) sangat relevan untuk dilakukan [4]. Penelitian terkini juga menunjukkan peningkatan pemanfaatan *deep learning* dalam klasifikasi fenomena visual berbasis lingkungan, termasuk upaya dalam memanfaatkan model ringan seperti *MobileNetV1* dalam konteks pengenalan objek visual alamiah [5]. Selain itu, studi oleh [6] memperkuat bahwa pemanfaatan *transfer learning* dan *unfreezing layer* mampu meningkatkan performa klasifikasi pada dataset dengan kompleksitas visual yang tinggi. Dalam penelitian terbaru, pendekatan berbasis CNN juga diterapkan untuk klasifikasi musim di wilayah subtropis dengan hasil akurasi yang kompetitif, meskipun penggunaan arsitektur ringan masih jarang dibahas secara eksplisit [7]. Temuan ini menegaskan pentingnya eksplorasi terhadap model ringan yang efisien untuk klasifikasi visual berskala besar, termasuk dalam konteks empat musim.

Permasalahan utama dalam penelitian ini terletak pada kurangnya studi yang secara langsung membandingkan performa dua arsitektur ringan CNN, yaitu *MobileNetV1* dan *EfficientNetB0*, dalam konteks klasifikasi citra empat musim. Selain itu, tantangan teknis juga muncul dari kompleksitas visual antar musim yang sering kali tumpang tindih, seperti kemiripan warna langit atau vegetasi yang berubah secara gradual. Variasi ini dapat menyebabkan kesalahan klasifikasi jika model tidak cukup sensitif dalam membedakan pola spasial dan tekstural yang khas. Oleh karena itu, diperlukan pendekatan arsitektural yang tepat serta strategi pelatihan yang mampu mengatasi perbedaan visual halus antar kelas dan memastikan kemampuan generalisasi yang baik terhadap data baru.

CNN telah terbukti efektif dalam mengekstraksi fitur visual dan melakukan klasifikasi citra secara otomatis melalui serangkaian *convolutional layer*, *pooling*, dan *activation function* [8]. CNN juga memungkinkan sistem untuk belajar langsung dari data mentah tanpa memerlukan proses ekstraksi fitur [9]. Arsitektur CNN yang dirancang untuk efisiensi seperti *MobileNetV1* dan *EfficientNetB0* banyak digunakan karena dapat bekerja optimal meskipun pada perangkat dengan sumber daya terbatas. *MobileNetV1* mengandalkan struktur *depthwise separable convolution* untuk mengurangi kompleksitas komputasi [8], sedangkan *EfficientNetB0* menggunakan pendekatan *compound scaling* yang secara proporsional mengatur kedalaman, lebar, dan resolusi model [9][10].

Untuk meningkatkan kemampuan model dalam memahami citra berdasarkan musim, teknik *transfer learning* diterapkan. Pendekatan ini memungkinkan model menggunakan bobot yang telah dipelajari dari dataset besar seperti ImageNet, kemudian disesuaikan ulang agar mengenali pola visual yang lebih spesifik dalam konteks pergantian musim [11]. Salah satu strategi penting adalah membuka kembali lapisan-lapisan tertentu dari model pra-latih (*unfreezing*), agar model dapat menyempurnakan pemahaman terhadap pola visual khas dari masing-masing musim. Proses ini berperan dalam memperkaya representasi fitur yang awalnya bersifat umum menjadi lebih kontekstual terhadap data berbasis musim [4].

Lapisan *GlobalAveragePooling2D* digunakan untuk mereduksi fitur spasial menjadi satu vektor ringkas yang tetap merepresentasikan keseluruhan informasi citra [12]. Jaringan kemudian dilengkapi dengan lapisan *Dense*,

Dropout, dan *BatchNormalization* untuk meminimalkan *overfitting* dan menjaga kestabilan pelatihan [13]. *Softmax activation function* digunakan pada lapisan output untuk mengklasifikasikan citra ke dalam empat kelas musim, dengan menghasilkan distribusi probabilitas terhadap setiap kelas [14].

Model dioptimasi menggunakan *Adam Optimizer*, dikombinasikan dengan teknik *callbacks* seperti *early stopping* dan pengurangan laju pembelajaran (*learning rate scheduling*) guna mempercepat konvergensi dan menghindari pelatihan berlebih [15]. Selain itu, teknik augmentasi citra seperti *image rotation*, *horizontal flip*, *vertical flip*, *image zoom*, serta *brightness level adjustment* digunakan untuk meningkatkan keragaman data *train* dan memperkuat generalisasi model [1].

Penelitian ini bertujuan untuk membandingkan performa dua arsitektur CNN ringan, yaitu *MobileNetV1* dan *EfficientNetB0*, dalam mengklasifikasikan citra visual berdasarkan musim. Evaluasi dilakukan menggunakan metrik *accuracy*, *precision*, *recall*, *F1-score*, *confusion matrix*, serta analisis grafik *accuracy* dan *loss* selama pelatihan. Studi ini diharapkan dapat memberikan kontribusi terhadap pengembangan sistem klasifikasi citra berbasis musim yang efisien dan akurat.

II. PENELITIAN YANG TERKAIT

Penelitian mengenai klasifikasi citra berbasis musim menggunakan pendekatan *deep learning* masih tergolong terbatas, terutama yang membandingkan arsitektur ringan seperti *MobileNetV1* dan *EfficientNetB0*. Namun, sejumlah studi sebelumnya dapat dijadikan acuan dalam memahami penerapan *Convolutional Neural Network (CNN)* dan teknik *transfer learning* untuk klasifikasi citra dengan karakteristik visual yang kompleks.

Penelitian oleh Yuliana dan Widodo [4] menunjukkan bahwa CNN efektif digunakan dalam klasifikasi citra berdasarkan elemen visual tertentu seperti tekstur dan warna, dengan memanfaatkan teknik *transfer learning* untuk mempercepat konvergensi dan mengurangi kebutuhan data pelatihan dalam jumlah besar. Studi tersebut menggunakan model *ResNet* untuk klasifikasi citra lanskap dan menghasilkan akurasi di atas 90%.

Tuna dan Kristianto [1] meneliti klasifikasi cuaca dengan menggunakan CNN berbasis VGG16 pada citra cuaca seperti cerah, berawan, dan hujan. Penelitian ini menunjukkan bahwa model pra-latih (*pre-trained*) dapat secara signifikan meningkatkan akurasi meskipun data yang digunakan tidak terlalu besar. Meskipun fokusnya bukan pada klasifikasi musim, pendekatan mereka relevan dengan konteks pengenalan kondisi visual berbasis citra.

Sementara itu, penelitian oleh Prisma dan Pamungkas [12] menggunakan arsitektur *EfficientNet* untuk klasifikasi jenis pemandangan alam. Dengan memanfaatkan *compound scaling* dan *dropout*, mereka mampu mengurangi *overfitting* dan meningkatkan generalisasi model terhadap berbagai variasi data. Penelitian ini mendukung pemilihan *EfficientNetB0* sebagai salah satu arsitektur dalam penelitian ini.

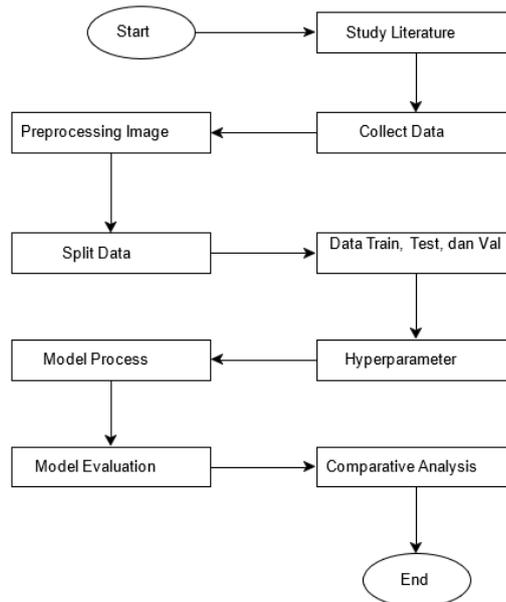
Dalam penelitian lain, Ananda et al. [16] menerapkan *MobileNetV2* untuk klasifikasi gambar tanaman tropis dengan akurasi mencapai 93,5%. Model ini dikombinasikan dengan augmentasi data dan *batch normalization*, menunjukkan performa optimal meskipun dengan sumber daya komputasi yang terbatas. Hal ini sejalan dengan penggunaan *MobileNetV1* dalam penelitian ini sebagai arsitektur ringan yang efisien.

Studi oleh Hidayat dan Utami [17] juga menunjukkan bahwa distribusi dataset yang seimbang sangat mempengaruhi performa klasifikasi. Mereka melakukan klasifikasi visual pada kondisi lingkungan dengan memanfaatkan pembagian data *train-validasi-test* yang proporsional, serta teknik augmentasi citra untuk mengatasi *overfitting*.

Penelitian-penelitian tersebut menjadi dasar dalam pemilihan metode dan arsitektur pada penelitian ini, khususnya penggunaan *transfer learning*, *data augmentation*, dan arsitektur ringan. Perbedaan utama pada penelitian ini adalah fokus pada klasifikasi empat musim utama, yaitu *autumn*, *spring*, *summer*, dan *winter*, yang belum banyak dikaji secara komprehensif dalam satu sistem klasifikasi berbasis CNN.

III. METODE PENELITIAN

Metodologi penelitian yang dilakukan terdiri dari *study literature*, *collect data*, *preprocessing*, *split data*, *custom hyperparameter*, *model process*, *model evaluation*, dan *comparative analysis*. Gambar 1 menunjukkan alur metodologi penelitian.



Gambar 1. Metode Penelitian.

A. Study Literature

Penulis menggunakan salah satu teknik bertujuan untuk melakukan penelusuran literatur untuk memahami berbagai pendekatan dan penerapan teknologi *deep learning*, khususnya pada model *convolutional neural network* (CNN) dalam klasifikasi citra [18]. Proses penelitian ini disusun berdasarkan tahapan sistematis, mulai dari *collect data*, *preprocessing*, *split data*, *custom hyperparameter*, *model process*, *model evaluation*, hingga *comparative analysis*, yang telah banyak digunakan dalam penelitian klasifikasi citra visual [19][20]. Pendekatan ini diadaptasi untuk memastikan proses klasifikasi citra empat musim dapat berjalan secara optimal, terukur, serta memberikan hasil evaluasi performa model yang objektif dan dapat dibandingkan secara adil antara dua arsitektur yang digunakan, yaitu *MobileNetV1* dan *EfficientNetB0*.

B. Collect Data

Pada penelitian ini, penulis melakukan proses pengumpulan data citra secara manual dari berbagai sumber daring seperti Google Images, Pinterest, iStock, dan situs penyedia gambar lainnya. Proses ini difokuskan untuk memperoleh gambar-gambar bertema musim, yaitu *autumn*, *spring*, *summer*, dan *winter*. Total data yang berhasil dikumpulkan sebanyak 2000 gambar, yang kemudian dikelompokkan secara seimbang ke dalam empat kelas musim, masing-masing terdiri dari 500 gambar.

Pengumpulan data secara manual dipilih untuk memastikan keberagaman visual yang lebih luas dan realistis, mencakup variasi sudut pengambilan gambar, resolusi, kondisi cuaca, pencahayaan, hingga karakteristik lingkungan yang khas pada setiap musim. Variasi tersebut meliputi elemen visual seperti dominasi warna, vegetasi, fenomena atmosfer, serta karakter pencahayaan yang berbeda antara *autumn* dengan dedaunan berguguran, *spring* dengan bunga yang bermekaran, *summer* dengan pencahayaan yang lebih intens, dan *winter* dengan dominasi elemen salju [21].

Keberagaman ini sangat penting dalam proses pelatihan model, agar arsitektur CNN mampu mengenali pola visual dari masing-masing musim secara lebih akurat dan tidak terjebak pada bias dataset yang homogen [22]. Dengan pendekatan ini, diharapkan model memiliki generalisasi yang lebih baik ketika menghadapi data baru yang merepresentasikan kondisi musim serupa di dunia nyata.



Gambar 2. Autumn, Spring, Summer, dan Winter.

C. Preprocessing Image

Pada Pada tahap ini, penulis melakukan serangkaian proses *preprocessing* citra untuk memastikan bahwa data input yang digunakan dalam pelatihan model berada dalam format dan skala yang sesuai dengan kebutuhan arsitektur *Convolutional Neural Network* (CNN). Proses *preprocessing* dimulai dengan mengubah ukuran seluruh gambar menjadi dimensi seragam 224x224 piksel, sesuai dengan standar input pada arsitektur *MobileNetV1* dan *EfficientNetB0* [23]. Penyesuaian dimensi ini bertujuan agar seluruh data memiliki resolusi yang konsisten, sehingga memudahkan proses *feature extraction* oleh *convolutional layer*.

Selanjutnya, dilakukan proses normalisasi piksel menggunakan teknik *rescaling*, yaitu mengubah nilai piksel dari rentang 0–255 menjadi 0–1. Proses ini penting untuk mempercepat konvergensi selama pelatihan dan menjaga kestabilan numerik agar nilai gradien tidak terlalu besar atau kecil saat propagasi mundur (*backpropagation*) [24]. *Rescaling* juga memastikan bahwa *input* data memiliki rentang yang seragam, sehingga membantu model dalam mengenali pola visual secara lebih efisien [25].

Untuk meningkatkan keragaman data dan mencegah model mengalami *overfitting*, dilakukan augmentasi data secara sintetik. *Overfitting* terjadi ketika model terlalu menyesuaikan diri dengan data *train* dan kehilangan kemampuan untuk mengenali data baru [26]. Augmentasi dilakukan menggunakan *class ImageDataGenerator* dari library *Keras* yang diterapkan secara dinamis saat proses pelatihan berlangsung [13]. Beberapa transformasi augmentasi yang diterapkan meliputi *image rotation* sebesar 10 derajat, *zoom_range* antara 0.95 hingga 1.05, *width_shift_range* dan *height_shift_range* masing-masing sebesar 0.03, *horizontal flip*, *vertical flip*, serta *brightness_range* antara 0.8 hingga 1.2. Seluruh strategi augmentasi ini bertujuan untuk memperkaya variasi bentuk, posisi, pencahayaan, dan orientasi visual citra, sehingga model mampu mengenali pola musim secara lebih adaptif [27].

Selain proses augmentasi, dilakukan juga proses *image hashing*, yaitu menghasilkan nilai unik dari masing-masing gambar menggunakan algoritma *average hashing*. Tujuannya adalah untuk mendeteksi dan menghindari kemunculan gambar duplikat dalam dataset. Dengan cara ini, dataset yang digunakan tetap bersih dan bebas dari data yang berulang secara visual, meskipun memiliki nama *file* berbeda. Proses ini sangat penting dalam menjaga integritas dataset, agar model tidak belajar dari data yang redundan yang dapat mengakibatkan bias dalam pelatihan.

Proses *rename* pada citra juga diterapkan untuk memberikan struktur penamaan *file* yang seragam berdasarkan kelas musim dan indeks gambar, sehingga mempermudah organisasi data dan pelabelan secara otomatis dalam proses pelatihan.

Dengan tahapan *preprocessing* ini, data *train* tidak hanya memiliki dimensi dan skala yang seragam, tetapi juga variasi visual yang beragam serta kualitas data yang terkontrol. Hal ini diharapkan mampu meningkatkan performa generalisasi model dan meminimalkan kecenderungan *overfitting* [16].

Contoh visual dari citra musim autumn yang belum di *preprocessing* dan sesudah dilakukan *preprocessing* dapat dilihat pada Gambar 3 dan Gambar 4. Gambar 3 menunjukkan kondisi citra mentah sebelum *preprocessing*, sedangkan Gambar 4 memperlihatkan hasil citra setelah dilakukan proses *preprocessing*.



Gambar 3. Before Preprocessing.



Gambar 4. After Preprocessing.

D. Split Data

Distribusi dataset yang seimbang sangat krusial dalam proses pelatihan model *deep learning* untuk klasifikasi citra musim. Hal ini dikarenakan variasi elemen visual seperti pencahayaan, bentuk awan, warna langit, serta tekstur lingkungan dapat memengaruhi akurasi model dalam mengenali kondisi visual tertentu [28][29]. Ketidakseimbangan distribusi antar kelas, misalnya dominannya citra cuaca cerah dibandingkan kondisi hujan atau berawan, dapat menyebabkan model menjadi bias terhadap kelas mayoritas dan menurunkan performa generalisasi [17].

Oleh karena itu, dalam penelitian ini, data citra dibagi secara proporsional untuk masing-masing kelas dengan tujuan menciptakan distribusi yang adil dan seimbang. Pembagian data dilakukan menjadi tiga bagian, yaitu data pelatihan, validasi, dan pengujian. Skema pembagian yang digunakan adalah 75% untuk data pelatihan (*training*), 15% untuk data validasi (*validation*), dan 10% untuk data pengujian (*testing*). Komposisi ini dipilih untuk memberikan proporsi data pelatihan yang cukup besar, sehingga model memiliki ruang belajar yang optimal, sekaligus menjaga kualitas pengujian pada data yang belum pernah dilihat sebelumnya.

Total dataset yang digunakan dalam penelitian ini terdiri atas 2000 citra yang dibagi rata ke dalam empat kelas musim, yaitu *autumn*, *spring*, *summer*, dan *winter*, dengan masing-masing kelas berisi 500 citra. Setiap kelas kemudian dibagi menjadi 375 citra untuk data pelatihan, 75 citra untuk data validasi, dan 50 citra untuk data pengujian. Proses pembagian dataset ini dilakukan secara otomatis menggunakan skrip *Python* dalam file terpisah, guna memastikan distribusi data antar kelas berlangsung acak dan proporsional.

Pendekatan ini tidak hanya membantu mencegah bias dalam proses klasifikasi, tetapi juga memastikan bahwa analisis perbandingan performa kedua arsitektur, *MobileNetV1* dan *EfficientNetB0*, dapat dilakukan secara objektif dan adil dalam kondisi distribusi data yang seimbang.

TABEL I
RINCIAN SPLI DATA

Jenis	Train	Val	Test
Autumn	375	75	50
Spring	375	75	50
Summer	375	75	50
Winter	375	75	50

E. Hyperparameter

Pengaturan *hyperparameter* merupakan komponen penting dalam proses pelatihan model *deep learning* karena berpengaruh langsung pada efisiensi konvergensi, kestabilan pelatihan, dan hasil evaluasi model. *Hyperparameter* adalah parameter eksternal yang diatur sebelum proses pelatihan dimulai, seperti *batch size*, jumlah *epoch*, *learning rate*, dan konfigurasi strategi *callbacks*, yang tidak dioptimasi melalui proses *training* melainkan ditentukan oleh perancang model [13].

Dalam penelitian ini, proses pelatihan dilakukan dalam dua model pengujian yang menggunakan arsitektur *MobileNetV1* dan *EfficientNetB0*. Masing-masing model dirancang dengan konfigurasi *hyperparameter* yang

disesuaikan berdasarkan kompleksitas arsitektur untuk memastikan proses pelatihan berjalan optimal, serta mengurangi risiko *overfitting* dan *underfitting*.

MobileNetV1 dikonfigurasi dengan *batch size* sebesar 16, *adam optimizer*, *learning rate* 0.0007, dan jumlah *epoch* sebanyak 25. Di sisi lain, *EfficientNetB0* menggunakan *batch size* yang sama namun *learning rate* lebih rendah, yaitu 0.00005, dengan jumlah *epoch* sebanyak 20. Pemilihan nilai *batch size* 16 dilakukan berdasarkan pertimbangan keseimbangan antara efisiensi memori dan kestabilan pembelajaran. Ukuran ini tergolong kecil hingga sedang, sehingga memungkinkan pembaruan bobot lebih sering tanpa mengorbankan kapasitas memori, serta membantu model belajar lebih variatif karena gradien yang dihasilkan lebih beragam dibanding *batch size* besar. Perbedaan pengaturan *learning rate* ini bertujuan agar kedua model dapat menyesuaikan laju pembaruan bobot dengan struktur jaringan masing-masing, karena model yang lebih kompleks cenderung membutuhkan langkah pembaruan yang lebih kecil dan proses konvergensi yang lebih bertahap [30].

Untuk proses optimasi, algoritma *Adam (Adaptive Moment Estimation)* digunakan sebagai *optimizer*. *Adam* merupakan algoritma yang secara adaptif menyesuaikan *learning rate* setiap parameter selama proses training, dengan menggabungkan keunggulan dari metode momentum dan *adaptive learning rate*. Hal ini membuat proses pembaruan bobot lebih stabil, mencegah osilasi yang berlebihan, dan mempercepat model menuju solusi global optimum [31].

Selain pengaturan dasar, strategi *callbacks* diterapkan untuk mendukung stabilitas pelatihan dan menghindari pembelajaran berlebihan pada data pelatihan. Fungsi *EarlyStopping* digunakan untuk menghentikan proses pelatihan ketika performa model pada data validasi tidak mengalami peningkatan dalam sejumlah *epoch* yang telah ditentukan, sedangkan *ReduceLROnPlateau* digunakan untuk menyesuaikan nilai *learning rate* secara adaptif apabila model mengalami stagnasi dalam penurunan *loss*. Pada *MobileNetV1*, nilai *patience* untuk *EarlyStopping* ditetapkan sebesar 5, dan pada *ReduceLROnPlateau* juga sebesar 5. Pada *EfficientNetB0*, nilai *patience* untuk *EarlyStopping* ditetapkan sebesar 3 dan *ReduceLROnPlateau* sebesar 2. Pemilihan nilai *patience* ini didasarkan pada pengamatan empiris terhadap tren *validation loss*. *MobileNetV1* diberikan nilai *patience* yang lebih tinggi karena model cenderung membutuhkan lebih banyak *epoch* untuk mencapai konvergensi, sedangkan *EfficientNetB0* menunjukkan perubahan performa yang lebih cepat sehingga digunakan *patience* lebih rendah agar respons terhadap stagnasi dapat segera dilakukan dan mencegah *overfitting*. Pengaturan ini ditentukan berdasarkan pengamatan empiris terhadap tren validasi *loss*, sehingga proses pelatihan dapat berhenti atau menyesuaikan *learning rate* tepat waktu [32].

Penggunaan kombinasi pengaturan *hyperparameter* ini tidak hanya bertujuan untuk mendapatkan hasil prediksi dengan tingkat akurasi tinggi, tetapi juga untuk menjaga keseimbangan antara kemampuan model dalam mempelajari pola data pelatihan dan kemampuannya dalam melakukan generalisasi terhadap data baru. Pengaturan ini diharapkan dapat meminimalkan risiko *overfitting*, di mana model terlalu menghafal data pelatihan, dan *underfitting*, di mana model gagal menangkap pola yang ada dalam data [26]. Dengan konfigurasi ini, diharapkan kedua arsitektur dapat menunjukkan performa optimal dalam mengklasifikasikan citra empat musim.

F. Model Process

Pada penelitian ini, pengembangan model klasifikasi citra empat musim dilakukan dengan memanfaatkan arsitektur *Convolutional Neural Network (CNN)* berbasis *transfer learning*. Dua arsitektur ringan dipilih, yaitu *MobileNetV1* dan *EfficientNetB0*, yang masing-masing telah terbukti memiliki keunggulan dalam efisiensi komputasi dan performa akurasi [8][9].

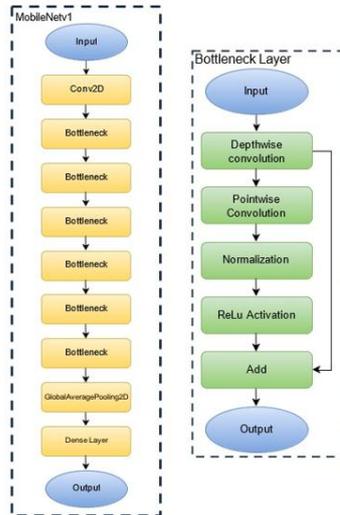
Pemilihan kedua arsitektur ini didasarkan pada kemampuannya dalam menangani tugas klasifikasi citra dengan efisiensi tinggi tanpa mengorbankan akurasi. *MobileNetV1* dikenal cocok digunakan pada perangkat dengan keterbatasan sumber daya karena jumlah parameternya yang kecil dan kecepatan inferensi yang tinggi [33]. Di sisi lain, *EfficientNetB0* banyak digunakan dalam berbagai penelitian karena menawarkan keseimbangan optimal antara kompleksitas model dan performa, sehingga relevan untuk digunakan dalam konteks klasifikasi citra visual empat musim [34].

MobileNetV1 memanfaatkan metode *depthwise separable convolution*, di mana proses ekstraksi fitur visual dari citra dipisahkan menjadi dua tahapan utama, yaitu *depthwise convolution* dan *pointwise convolution*. *Depthwise convolution* bertugas untuk mengekstraksi fitur dari masing-masing *channel* secara terpisah, sedangkan *pointwise convolution* menggunakan *filter* 1×1 untuk menggabungkan informasi antar *channel* menjadi representasi yang lebih komprehensif [8][19]. Pendekatan ini secara signifikan mengurangi jumlah *parameter* model dan beban komputasi,

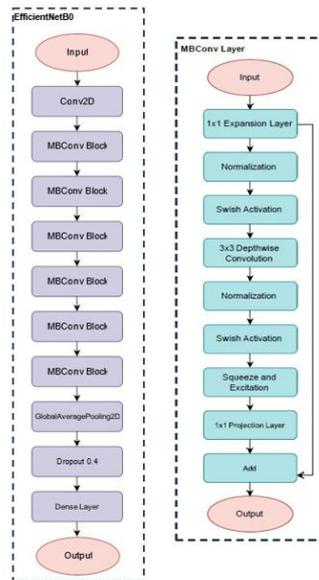
sehingga lebih hemat sumber daya namun tetap menjaga akurasi klasifikasi.

Sementara itu, *EfficientNetB0* menerapkan teknik *compound scaling* yang memungkinkan pengaturan skala arsitektur model secara proporsional pada kedalaman *layer*, lebar *layer*, dan resolusi *input*. Strategi ini dirancang untuk menghasilkan *trade-off* optimal antara efisiensi komputasi dan performa akurasi [9][10].

Gambar 5 menunjukkan model arsitektur *MobileNetV1* dan Gambar 6 menunjukkan model arsitektur *EfficientNetB0*.



Gambar 5. Model Arsitektur *MobileNetV1*.



Gambar 6. Model Arsitektur *EfficientNetB0*

Selanjutnya, proses *transfer learning* dilakukan dengan memanfaatkan *bobot* yang telah dipelajari oleh kedua arsitektur pada *dataset ImageNet*. Pada tahap ini, dilakukan pembaruan lapisan pada bagian *fully connected layer* untuk menyesuaikan model dengan kebutuhan klasifikasi citra empat musim. Struktur dari *fully connected layer* tersebut ditampilkan pada Tabel 2 untuk *MobileNetV1* dan Tabel 3 untuk *EfficientNetB0*.

Tabel III
CUSTOM FULLY CONNECTED LAYER MOBILENETV1

Layer	Output Shape
Global Avarage Pooling	(None, 1024)
Dropout Layer	(None, 1024)
Dense Layer	(None, 128)
Dropout Layer_1	(None, 128)
Dense Layer_1	(None, 4)

Tabel IIIII
CUSTOM FULLY CONNECTED LAYER EFFICIENTNETB0

Layer	Output Shape
Global Avarage Pooling	(None, 1280)
Dropout Layer	(None, 1280)
Dense Layer	(None, 128)
Dropout Layer_1	(None, 128)
Dense Layer_1	(None, 4)

Setelah proses *feature extraction* selesai, model dilengkapi dengan lapisan *GlobalAveragePooling2D* yang berfungsi untuk mereduksi dimensi spasial dari hasil ekstraksi fitur menjadi vektor satu dimensi. Proses ini bertujuan untuk mencegah *overfitting*, mempercepat konvergensi, dan menjaga informasi penting dari citra dengan cara menghitung rata-rata dari setiap *feature map*, sehingga menghasilkan representasi yang lebih ringkas tanpa kehilangan informasi esensial [12][35].

Model kemudian diperkaya dengan penambahan beberapa lapisan *Dense*, *Dropout*, dan *BatchNormalization*. Penggunaan *Dropout* dalam penelitian ini dilakukan sebanyak dua kali, dengan tingkat penghapusan neuron sebesar 0.4 pada *Dropout* pertama dan 0.3 pada *Dropout* kedua. Penerapan *Dropout* ini bertujuan untuk mengurangi risiko *overfitting* dengan cara menonaktifkan neuron secara acak selama proses pelatihan, sehingga model dapat mempelajari representasi fitur yang lebih *general* dan tidak bergantung pada pola data yang spesifik [36]. Selain itu, *BatchNormalization* digunakan untuk menstabilkan distribusi *output* antar lapisan dan mempercepat proses pelatihan [37].

Pada bagian akhir, lapisan *output* dari model menggunakan *softmax activation function*, yang mengubah hasil *output* menjadi distribusi probabilitas atas empat kelas musim. Fungsi ini memungkinkan proses klasifikasi menjadi lebih terstandarisasi dan memudahkan pengambilan keputusan berdasarkan probabilitas tertinggi dari masing-masing kelas [14][35].

G. Model Evaluation

Dalam proses evaluasi model, penulis menggunakan metode *confusion matrix* sebagai alat untuk mengukur performa model dalam mengklasifikasikan citra empat musim. Dari *confusion matrix* ini, diperoleh perhitungan metrik evaluasi seperti *accuracy*, *precision*, *recall*, dan *F1-score*, yang digunakan untuk menilai sejauh mana model mampu memprediksi kelas citra secara tepat. Proses evaluasi ini diterapkan pada masing-masing model pengujian yang telah dirancang, dan perhitungan nilai metrik evaluasi mengacu pada persamaan yang ditampilkan pada bagian berikut:

$$Accuracy = \frac{(TP+TN)}{(TP+FP+TN+FN)} \quad (1)$$

$$Precision = \frac{TP}{(TP+FP)} \quad (2)$$

$$Recall = \frac{TP}{(FP+FN)} \quad (3)$$

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

True Positive (TP) merupakan jumlah data yang diprediksi benar sebagai kelas yang sesuai, sedangkan *True Negative* (TN) adalah jumlah data yang diprediksi benar sebagai kelas yang bukan target. *False Positive* (FP) mengacu pada jumlah data yang seharusnya bukan kelas target, namun salah diklasifikasikan sebagai kelas tersebut oleh model. Sebaliknya, *False Negative* (FN) merupakan jumlah data yang seharusnya termasuk dalam kelas target, tetapi gagal diprediksi oleh model [19].

Keempat komponen ini menjadi dasar dalam perhitungan metrik evaluasi seperti *accuracy*, *precision*, *recall*, dan *F1-score*, sehingga model tidak hanya dinilai dari tingkat ketepatan prediksi secara keseluruhan, namun juga dari ketepatan dalam memisahkan kelas positif dan negatif secara spesifik [35].

H. Comparative Analysis

Pada penelitian ini, dilakukan analisis komparatif untuk mengevaluasi kinerja dua model pengujian yang menggunakan arsitektur *MobileNetV1* dan *EfficientNetB0* dalam klasifikasi citra empat musim, yaitu *autumn*, *spring*, *summer*, dan *winter*. Pendekatan ini bertujuan untuk mengidentifikasi model yang paling optimal dalam mengenali pola visual khas dari masing-masing musim, berdasarkan evaluasi metrik performa seperti *accuracy*, *precision*, *recall*, dan *F1-score*.

Penggunaan *transfer learning* dalam kedua model pengujian dilakukan untuk memanfaatkan bobot pra-latih dari dataset berskala besar, sehingga proses pelatihan dapat berjalan lebih efisien dan model mampu beradaptasi dengan data citra musim yang memiliki keragaman visual tinggi. Pendekatan ini telah banyak diterapkan dalam penelitian klasifikasi citra dan terbukti mampu mempercepat proses konvergensi dan meningkatkan akurasi [38]. Rincian model pengujian yang digunakan dalam penelitian ini disajikan pada tabel 4 berikut.

Tabel IVII
MODEL PENGUJIAN PENELITIAN

Model	Optimizer	Model Architecture
1	Adam	Transfer Learning <i>MobileNetV1</i>
2	Adam	Transfer Learning <i>EfficientNetB0</i>

IV. HASIL DAN PEMBAHASAN

Pada penelitian ini, dilakukan pengujian untuk mencari model klasifikasi citra empat musim dengan akurasi terbaik, menggunakan metode *confusion matrix* sebagai alat evaluasi. Perbandingan difokuskan pada dua arsitektur berbeda, yaitu *MobileNetV1* dan *EfficientNetB0*, dengan konfigurasi *transfer learning* dan *optimizer* yang serupa pada kedua model.

A. Model Pengujian *MobileNetV1*

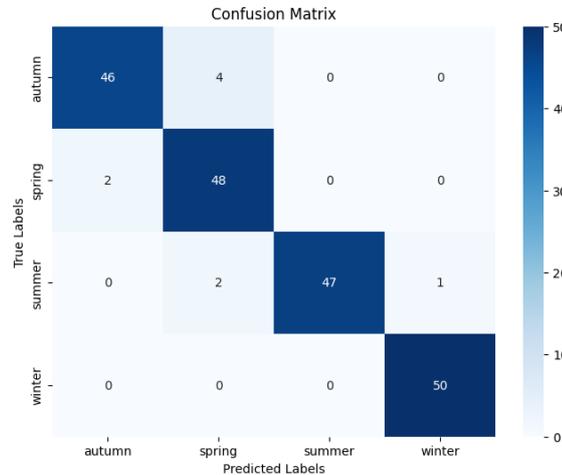
Berdasarkan hasil pengujian pada model pengujian pertama, yaitu arsitektur *MobileNetV1* yang menggunakan *adam optimizer* dan pendekatan *transfer learning*, didapatkan nilai *accuracy* sebesar 95,5%, *precision* sebesar 95,75%, *recall* sebesar 95,5%, dan *F1-score* sebesar 95,5%. Persentase hasil klasifikasi berdasarkan *confusion matrix* ditampilkan pada Tabel 5.

Tabel V
PERSENTASE *CONFUSION MATRIX MOBILENETV1*

Season	Precision	Recall	F1-Score
Autumn	96%	92%	94%
Spring	89%	96%	92%
Summer	100%	94%	97%
Winter	98%	100%	99%

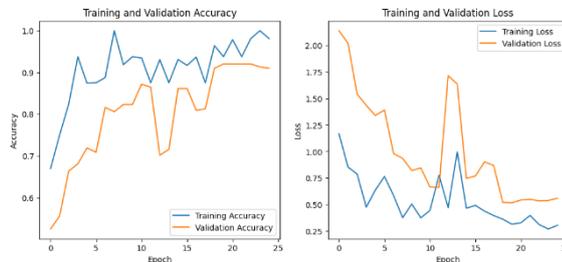
Model klasifikasi ini menunjukkan performa yang baik dalam mengidentifikasi keempat kelas musim, yang

ditunjukkan melalui nilai metrik evaluasi *recall*, *precision*, dan *F1-score* yang tinggi. Hal ini juga dibuktikan oleh dominasi prediksi benar pada setiap kelas dalam *confusion matrix*. Visualisasi dari distribusi hasil klasifikasi untuk model *MobileNetV1* dapat dilihat pada Gambar 7.



Gambar 7. Confusion Matrix *MobileNetV1*

Grafik proses pelatihan dan validasi pada model *MobileNetV1* menunjukkan pola peningkatan performa yang signifikan. Terlihat bahwa akurasi pelatihan mengalami kenaikan secara konsisten hingga mendekati 100%, sementara akurasi validasi juga menunjukkan tren peningkatan meskipun cenderung lebih fluktuatif dan berhenti di kisaran 90%. Di sisi lain, nilai *loss* pada data pelatihan mengalami penurunan yang stabil hingga mendekati nol, sedangkan *loss* validasi sempat mengalami fluktuasi sebelum akhirnya menurun dan cenderung stabil pada nilai yang rendah di akhir pelatihan. Grafik ini menunjukkan bahwa model mampu belajar dengan baik dan memiliki kemampuan generalisasi yang cukup kuat terhadap data validasi. Visualisasi detail proses pelatihan dan validasi ditampilkan pada Gambar 8.



Gambar 8. Grafik Model Pengujian *MobileNetV1*

B. Model Pengujian *EfficientNetB0*

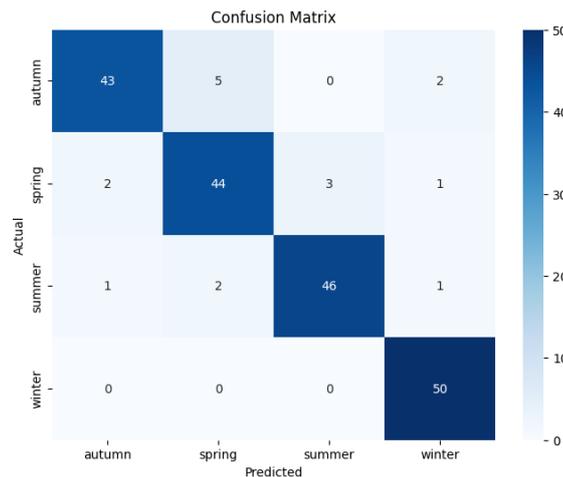
Berdasarkan hasil pengujian pada model pengujian kedua, yaitu arsitektur *EfficientNetB0* yang juga menggunakan *adam optimizer* dan pendekatan *transfer learning*, didapatkan nilai *accuracy* sebesar 91,5%, *precision* sebesar 91,5%, *recall* sebesar 91,5%, dan *F1-score* sebesar 91,5%. Persentase hasil klasifikasi berdasarkan *confusion matrix* ditampilkan pada Tabel 6.

Tabel VI
PERSENTASE *CONFUSION MATRIX EFFICIENTNETB0*

Season	Precision	Recall	F1-Score
Autumn	93%	86%	90%
Spring	86%	88%	87%

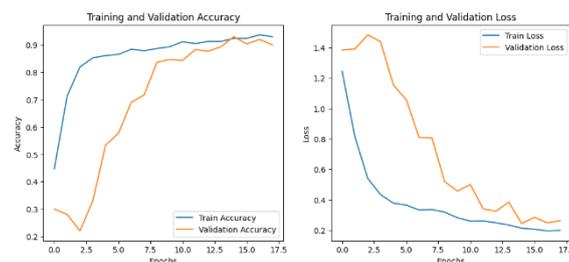
Summer	94%	92%	93%
Winter	93%	100%	96%

Model klasifikasi yang dibangun menggunakan arsitektur *EfficientNetB0* juga menunjukkan performa yang kompetitif dalam mengklasifikasikan citra empat musim. Nilai evaluasi seperti *accuracy*, *precision*, *recall*, dan *F1-score* yang tinggi mencerminkan bahwa model mampu mengenali pola visual dari masing-masing kelas secara efektif. Hasil prediksi benar yang mendominasi pada tiap kelas dalam *confusion matrix* semakin memperkuat bahwa model ini memiliki kemampuan generalisasi yang baik terhadap data validasi. Visualisasi distribusi hasil klasifikasi dari model *EfficientNetB0* disajikan pada Gambar 9



Gambar 9. Confusion Matrix *EfficientNetB0*

Grafik proses pelatihan dan validasi pada model *EfficientNetB0* menunjukkan pola pembelajaran yang stabil dan konvergen. Akurasi pelatihan mengalami peningkatan secara konsisten hingga mendekati angka 95%, sementara akurasi validasi juga menunjukkan tren kenaikan yang baik dan berhenti di kisaran lebih dari 90%, menandakan proses generalisasi yang optimal. Selain itu, nilai *loss* pada data pelatihan dan validasi sama-sama menunjukkan penurunan yang signifikan seiring bertambahnya *epoch*, dengan perbedaan yang semakin kecil di akhir proses pelatihan. Grafik ini mengindikasikan bahwa model *EfficientNetB0* mampu belajar secara efektif dan tidak menunjukkan indikasi *overfitting* maupun *underfitting*. Visualisasi lengkap proses pelatihan dan validasi dapat dilihat pada Gambar 10.



Gambar 10. Grafik Model Pengujian *EfficientNetB0*

V. KESIMPULAN

Penelitian ini bertujuan untuk mengklasifikasikan citra visual ke dalam empat kategori musim, yaitu *autumn*, *spring*, *summer*, dan *winter*, menggunakan dua arsitektur CNN ringan, yakni *MobileNetV1* dan *EfficientNetB0*, dengan pendekatan *transfer learning* dan optimasi menggunakan *Adam (Adaptive Moment Estimation) Optimizer*. Sebanyak 2000 citra digunakan dan dibagi secara proporsional menjadi data *training*, *validation*, dan *testing* dengan

rasio 75:15:10. Tujuan utama penelitian ini adalah untuk mengevaluasi dan membandingkan performa kedua arsitektur dalam melakukan klasifikasi citra musim secara efisien dan akurat.

Hasil pengujian menunjukkan bahwa kedua arsitektur mampu melakukan tugas klasifikasi dengan performa yang baik, namun *MobileNetV1* menunjukkan hasil yang lebih unggul dibandingkan *EfficientNetB0*. Model *MobileNetV1* berhasil mencapai nilai *accuracy* sebesar 95,5%, *precision* 95,75%, *recall* 95,5%, dan *F1-score* 95,5%, sedangkan *EfficientNetB0* hanya mencapai *accuracy* sebesar 91,5%. Keunggulan *MobileNetV1* ditunjukkan tidak hanya melalui metrik evaluasi kuantitatif, tetapi juga melalui visualisasi *confusion matrix* yang menggambarkan distribusi prediksi yang dominan tepat pada setiap kelas.

Secara arsitektural, *MobileNetV1* memiliki keunggulan dalam hal efisiensi komputasi melalui penerapan *depthwise separable convolution*, yang menjadikannya lebih ringan dan cepat dibandingkan *EfficientNetB0*. Meskipun *EfficientNetB0* mengusung pendekatan *compound scaling* yang bertujuan untuk menyeimbangkan kompleksitas dan performa, hasil penelitian menunjukkan bahwa arsitektur ini kurang optimal dalam mengenali perbedaan pola visual antar musim pada dataset yang digunakan.

Seluruh proses *training* dan *validation* menunjukkan tren yang stabil pada grafik *accuracy* dan *loss*, tanpa adanya indikasi *overfitting*. Hal ini menunjukkan bahwa kombinasi teknik *preprocessing*, augmentasi data, *transfer learning*, serta pengaturan *hyperparameter* yang tepat telah berhasil meningkatkan kemampuan generalisasi dari kedua model. Penerapan teknik *callback* seperti *EarlyStopping* dan *ReduceLROnPlateau* juga turut berkontribusi dalam mengontrol proses pelatihan agar tidak berlangsung secara berlebihan.

Secara keseluruhan, penelitian ini menyimpulkan bahwa *MobileNetV1* merupakan arsitektur yang paling optimal untuk tugas klasifikasi citra empat musim dalam konteks ini. Temuan ini menunjukkan bahwa model ringan seperti *MobileNetV1* tidak hanya cocok untuk perangkat dengan sumber daya terbatas, tetapi juga mampu menghasilkan akurasi tinggi dalam klasifikasi visual kompleks seperti musim. Penelitian ini diharapkan dapat memberikan kontribusi nyata terhadap pengembangan sistem klasifikasi visual berbasis citra untuk keperluan sistem monitoring lingkungan, sistem informasi geografis, maupun aplikasi berbasis web untuk prediksi cuaca otomatis..

DAFTAR PUSTAKA

- [1] M. D. Tuna dan H. Kristianto, "Klasifikasi Citra Cuaca Menggunakan Convolutional Neural Network Berbasis VGG16," *Jurnal Teknologi dan Sistem Komputer*, vol. 10, no. 1, pp. 55–61, 2021.
- [2] D. Mylonas, G. Tzagkarakis, dan P. Tsakalides, "Season Classification from Images Using Deep Learning Approaches," *IEEE Access*, vol. 11, pp. 25344–25353, 2023.
- [3] W. Xiao, C. Wang, dan L. Wang, "Weather Recognition Based on Deep Convolutional Neural Network," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 1, pp. 424–432, 2021.
- [4] N. Yuliana dan S. Widodo, "Penerapan Deep Learning untuk Klasifikasi Citra Lanskap Menggunakan Transfer Learning," *Jurnal Ilmu Komputer dan Informatika*, vol. 9, no. 2, pp. 88–96, 2022.
- [5] Dewantara, A., Suryanto, B., & Widodo, C. (2024). Penerapan Arsitektur Convolutional Neural Networks (CNN) dalam Klasifikasi Citra Medis. *Jurnal Teknologi Informasi dan Komputer*, 15(2), 23–34.
- [6] Prasetya, M., & Aisyah, R. (2025). Pengembangan Deep Learning untuk Pengenalan Pola pada Citra Digital Menggunakan CNN. *Jurnal Ilmu Komputer dan Teknologi*, 8(1), 45–56.
- [7] Herlambang, D., & Wulandari, T. (2025). Optimasi Arsitektur CNN dalam Sistem Pengenalan Wajah. *Jurnal Penelitian AI dan Deep Learning*, 10(3), 121–135.
- [8] A. G. Howard et al., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," arXiv preprint arXiv:1704.04861, 2017.
- [9] M. Tan dan Q. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in *International Conference on Machine Learning (ICML)*, pp. 6105–6114, 2019.
- [10] A. Setiawan, R. Firdaus, dan M. Nurfadillah, "Implementasi EfficientNet untuk Klasifikasi Citra Lingkungan," *Jurnal Pengolahan Citra dan AI*, vol. 6, no. 1, pp. 45–52, 2023.
- [11] R. Nandaputra, "Penerapan Transfer Learning pada Klasifikasi Citra Alam dengan Dataset Terbatas," *Jurnal Informatika Terapan*, vol. 5, no. 2, pp. 77–84, 2024.
- [12] R. Prisma dan A. D. Pamungkas, "Efektivitas EfficientNet pada Klasifikasi Citra Alam Menggunakan Teknik Compound Scaling," *Jurnal Pengolahan Citra dan AI*, vol. 6, no. 1, pp. 45–52, 2024.
- [13] R. Putra, A. Santoso, dan D. Pratama, "Optimasi Model CNN untuk Klasifikasi Citra Dengan Augmentasi Data," *Jurnal Informatika Terapan*, vol. 5, no. 3, pp. 110–117, 2020.
- [14] S. Wicaksono dan A. Halim, "Implementasi Softmax pada Jaringan Neural untuk Klasifikasi Multikelas," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 9, no. 2, pp. 112–120, 2023.
- [15] D. Rizky, T. Maulana, dan F. Hasan, "Penerapan Callbacks EarlyStopping dan ReduceLROnPlateau pada CNN untuk Klasifikasi Citra," *Jurnal Ilmu Komputer*, vol. 11, no. 1, pp. 44–51, 2022.
- [16] M. Ananda, A. Sutanto, dan A. Fauzan, "Klasifikasi Citra Tanaman Tropis Menggunakan MobileNetV2 dan Augmentasi Data," *Jurnal Informatika Terapan*, vol. 5, no. 3, pp. 110–117, 2023.
- [17] R. Hidayat dan D. Utami, "Pengaruh Distribusi Dataset Terhadap Akurasi Model Deep Learning untuk Klasifikasi Visual," *Jurnal Sains Data dan Teknologi Informasi*, vol. 7, no. 1, pp. 22–28, 2023.
- [18] Yanto, S. Rachmawati, dan T. Wibowo, "Penerapan CNN pada Klasifikasi Citra Dua Dimensi," *Jurnal Teknologi Informasi*, vol. 11, no. 1, pp. 33–40, 2021.

- [19] A. Hastomo dan Sudjiran, "Klasifikasi Citra Berbasis CNN untuk Deteksi Visual," *Jurnal Teknologi dan Sistem Komputer*, vol. 9, no. 2, pp. 77–83, 2021.
- [20] S. Miranda, R. Novamizanti, dan F. Rizal, "Implementasi CNN pada Klasifikasi Citra Objek Alam," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 6, no. 3, pp. 210–217, 2020.
- [21] Yanto et al., "Pengembangan Deep Learning untuk Identifikasi Visual," *Jurnal Teknologi dan Komputer*, vol. 13, no. 1, pp. 55–62, 2023.
- [22] A. Hafeez, M. S. Hossain, dan G. Muhammad, "Deep Learning Models for Environmental Scene Recognition," *IEEE Access*, vol. 11, pp. 34500–34512, 2023.
- [23] B. Sasongko, R. Irwansyah, dan A. Nugraha, "Pra-pemrosesan Citra Menggunakan Rescaling untuk Klasifikasi CNN," *Jurnal Informatika dan Sistem Informasi*, vol. 9, no. 1, pp. 12–19, 2023.
- [24] Husen, "Pengaruh Pra-pemrosesan Citra terhadap Akurasi Model CNN," *Jurnal Teknologi dan Sistem Komputer*, vol. 10, no. 2, pp. 64–70, 2024.
- [25] F. Ardianto, M. Farhan, dan A. Sari, "Normalisasi Citra untuk Peningkatan Performa CNN," *Jurnal Ilmu Komputer dan Sistem Informasi*, vol. 8, no. 2, pp. 89–95, 2023.
- [26] N. Nurfalah, F. Sari, dan T. Ramadhani, "Strategi Mengatasi Overfitting Pada CNN Menggunakan Augmentasi Data," *Jurnal Sains Komputer*, vol. 7, no. 1, pp. 33–40, 2022.
- [27] S. Rohmah, D. Saputra, dan I. Hidayati, "Implementasi Data Augmentation untuk Peningkatan Akurasi Deep Learning," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 6, no. 2, pp. 105–112, 2023.
- [28] Widodo dan Amalia, "Analisis Pengaruh Keseimbangan Dataset pada CNN," *Jurnal Sains Komputer Indonesia*, vol. 8, no. 2, pp. 77–83, 2023.
- [29] Hermanto dan Saputra, "Distribusi Data dan Akurasi Deep Learning," *Jurnal Teknologi Komputer*, vol. 5, no. 2, pp. 65–71, 2023.
- [30] Roy, M., M. Das, dan P. Das, "Comparative Study of Optimizers in Deep Learning Models," *Procedia Computer Science*, vol. 217, pp. 1667–1676, 2023.
- [31] D. P. Kingma dan J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv preprint arXiv:1412.6980*, 2015.
- [32] M. Achmad dan R. Haris, "Pengaruh Pengaturan Patience Pada EarlyStopping dan ReduceLRonPlateau dalam CNN," *Jurnal Teknologi Informasi*, vol. 11, no. 1, pp. 55–63, 2023.
- [33] Putri, A., Nugroho, R., & Sari, D. (2024). *Implementasi MobileNet pada klasifikasi citra cuaca ringan*. *Jurnal Teknologi Informasi dan Komputer*, 12(1), 55–62.
- [34] Hartono, A., & Wahyuni, S. (2025). *Efektivitas EfficientNet dalam klasifikasi citra berbasis musim menggunakan transfer learning*. *Jurnal Komputasi dan Sains Data*, 8(2), 88–97.
- [35] A. Arya, B. Huda, dan N. Kurniawan, "Optimasi GlobalAveragePooling pada Model CNN untuk Klasifikasi Citra," *Jurnal Sistem dan Informatika*, vol. 9, no. 1, pp. 30–38, 2023.
- [36] Y. Zhang, L. Yu, dan Q. Wang, "Dropout Strategies for Convolutional Neural Networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 271–283, 2021.
- [37] B. Almaslukh, "Batch Normalization in CNN: A Review," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 5, pp. 512–518, 2021.
- [38] A. Winarto, R. Mailasari, dan N. Purnamawati, "Implementasi Transfer Learning MobileNetV2 pada Klasifikasi Citra," *Jurnal Teknologi Informasi dan Sistem Komputer*, vol. 11, no. 1, pp. 45–51, 2023.