



Optimasi Kinerja Klasifikasi Motif Batik pada Arsitektur MobileNetV2 melalui Strategi Fine-Tuning

Gilbertus Kristo Anugerah Adviandito¹, Muhammad Ilham Afif², Agil Lantip Mahendra³, Much Aziz Muslim⁴

^{1,2,3,4}Informatics Engineering,
Semarang State University, Semarang, Indonesia

Abstract

Received: 20 Mei 2026
Revised: 29 Mei 2026
Accepted: 11 Juni 2026

Klasifikasi motif batik menghadapi tantangan signifikan sebagai masalah *fine-grained visual classification*, di mana tingginya similaritas visual antar-kelas menuntut kapabilitas ekstraksi fitur yang superior. Sementara banyak riset terdahulu mengandalkan arsitektur *Convolutional Neural Network* (CNN) berskala besar, pendekatan tersebut sering kali tidak efisien dari sisi komputasi. Penelitian ini bertujuan untuk menguji efektivitas *MobileNetV2* sebagai arsitektur *lightweight*, dengan fokus utama membuktikan bahwa optimasi melalui *fine-tuning* dapat menghasilkan akurasi tinggi yang setara dengan model kompleks. Secara metodologis, penelitian ini menerapkan *transfer learning* dua tahap, yaitu fase *feature extraction* dengan membekukan *base layer* untuk adaptasi awal, kemudian dilanjutkan *fine-tuning* pada lapisan konvolusi agar representasi fitur menjadi lebih spesifik terhadap karakteristik batik. Alur metode diperkuat melalui augmentasi citra, normalisasi, penggunaan *Global Average Pooling* (GAP) untuk pereduksi dimensi secara stabil, serta lapisan *dropout* untuk menekan *overfitting*. Hasil evaluasi menunjukkan bahwa strategi *fine-tuning* mampu meningkatkan akurasi secara signifikan dari model *MobileNetV2* tanpa *fine-tuning* meraih akurasi 94,00% dan ketika menggunakan *fine-tuning* bisa mencapai akurasi hingga mencapai 99,85%, membuktikan bahwa *MobileNetV2* yang dioptimalkan dapat menyamai performa model berparameter besar dengan efisiensi komputasi yang tetap tinggi. Analisis error menunjukkan bahwa model masih menghadapi tantangan minor dalam membedakan kelas-kelas dengan pola visual yang sangat identik (*inter-class similarity*). Temuan ini menegaskan bahwa model *lightweight* yang dioptimalkan layak diterapkan pada perangkat bersumber daya terbatas untuk mendukung pelestarian digital batik.

Keywords: Klasifikasi Motif Batik, MobileNetV2, *Transfer Learning*, *Lightweight Architecture*, *Fine-Grained Visual Classification*

(*) Corresponding Author: ¹gilbertuskristo@students.unnes.ac.id,
²milhamafif@students.unnes.ac.id, ³agillantip112@students.unnes.ac.id,
⁴a212muslim@mail.unnes.ac.id

How to Cite: Kristo Anugerah Adviandito, G., Ilham Afif, M., Lantip Mahendra, A., & Aziz Muslim, M. (2026). Optimasi Kinerja Klasifikasi Motif Batik pada Arsitektur MobileNetV2 melalui Strategi Fine-Tuning. *Jurnal Ilmiah Wahana Pendidikan*, 12(6.D), 199-213. Retrieved from <https://jurnal.peneliti.net/index.php/JIWP/article/view/13391>.

INTRODUCTION

Batik merupakan warisan budaya Indonesia yang telah diakui dunia sebagai *Intangible Cultural Heritage of Humanity* oleh UNESCO. Kekayaan motif batik tidak hanya merepresentasikan nilai estetika, tetapi juga menyimpan makna filosofis dan identitas budaya yang beragam[1]. Namun, keragaman dan kompleksitas pola batik menghadirkan tantangan signifikan dalam proses

identifikasi digital. Tingginya kemiripan pola antar-motif (*inter-class similarity*) menempatkan permasalahan ini ke dalam kategori *fine-grained visual classification*, di mana perbedaan visual antar kelas sangat samar dan menuntut kemampuan ekstraksi fitur yang mendalam dan spesifik[2].

Sebagian besar penelitian terdahulu merespons tantangan ini dengan memanfaatkan arsitektur *Convolutional Neural Network* (CNN) berskala besar untuk memaksimalkan akurasi. Kendati mampu menghasilkan performa tinggi, arsitektur tersebut menuntut beban komputasi yang besar (*computationally expensive*), sehingga membatasi skalabilitas dan implementasinya pada perangkat dengan keterbatasan sumber daya (*resource-constrained devices*). Keterbatasan ini membuka urgensi penelitian baru mengenai pemanfaatan arsitektur yang lebih ringan (*lightweight*), yang mampu menyeimbangkan efisiensi komputasi dengan performa klasifikasi yang tetap kompetitif[3].

MobileNetV2 hadir sebagai arsitektur *lightweight* yang dirancang khusus untuk efisiensi melalui mekanisme *inverted residuals* dan *linear bottleneck*. Desain ini memungkinkan pengurangan parameter secara drastis tanpa penurunan performa yang signifikan pada dataset umum[4][10]. Namun, penerapan arsitektur ringan pada domain kompleks seperti batik memiliki risiko; kapasitas model yang terbatas sering kali kesulitan menangkap fitur-fitur halus (*subtle features*) yang membedakan satu motif dengan motif lainnya. Oleh karena itu, sekadar menerapkan model ini secara mentah atau hanya menggunakan metode ekstraksi fitur sederhana (*feature extraction*) diprediksi tidak akan memadai untuk mencapai akurasi yang optimal[5].

Untuk mengatasi keterbatasan kapasitas pada arsitektur ringan tersebut, penelitian ini mengajukan strategi optimasi melalui mekanisme *fine-tuning*. Berbeda dengan pendekatan *transfer learning* standar yang hanya melatih lapisan akhir (*classifier*), penelitian ini berargumen bahwa proses membuka kembali (*unfreezing*) dan melatih ulang bobot pada lapisan konvolusi sangatlah krusial[6]. Proses *fine-tuning* memaksa *MobileNetV2* untuk memodifikasi representasi fitur generiknya menjadi representasi yang sangat spesifik terhadap tekstur dan pola batik[7]. Hipotesis utama penelitian ini adalah bahwa efisiensi *MobileNetV2*, jika dipadukan dengan adaptasi bobot yang mendalam melalui *fine-tuning*, mampu menghasilkan akurasi yang setara dengan model-model besar.

Kontribusi utama penelitian ini adalah memvalidasi secara kuantitatif dampak strategi *fine-tuning* dalam meningkatkan performa *MobileNetV2* pada kasus klasifikasi motif batik. Fokus penelitian tidak hanya pada pencapaian akurasi akhir, tetapi pada pembuktian bahwa model yang efisien dapat dioptimalkan untuk menangani tugas visual berkerumitan tinggi. Hasil penelitian ini diharapkan memberikan kontribusi nyata bagi pengembangan sistem pelestarian batik digital yang tangguh, akurat, namun tetap ringan untuk diimplementasikan pada berbagai platform perangkat pengguna.

METHODS

Penelitian ini digunakan arsitektur *MobileNetV2* sebagai model utama untuk melakukan klasifikasi gambar karena efisiensinya serta kemampuannya dalam mengekstraksi fitur visual secara optimal. *MobileNetV2* memanfaatkan *inverted residual blocks* dan *linear bottlenecks* yang mampu mengurangi beban

komputasi tanpa mengurangi kualitas representasi fitur[8][9]. Untuk menyesuaikan model terhadap tugas klasifikasi, ditambahkan lapisan Global Average Pooling yang berfungsi mereduksi output konvolusi menjadi sebuah vektor fitur yang ringkas.

Setelah itu, ditambahkan lapisan Dense dengan aktivasi Softmax sebagai lapisan klasifikasi akhir untuk menghasilkan probabilitas dari setiap kelas. Selain itu, digunakan lapisan Dropout sebelum lapisan klasifikasi untuk mencegah *overfitting* dengan cara menonaktifkan sebagian neuron secara acak selama pelatihan sehingga model belajar representasi yang lebih robust dan tidak bergantung pada pola tertentu.

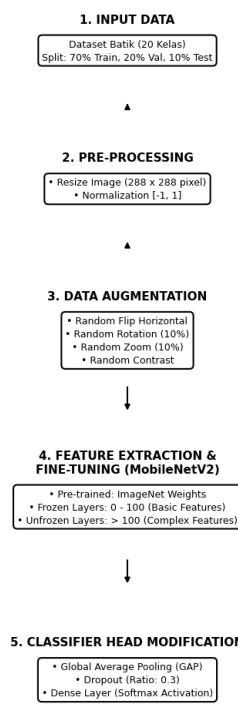
Proses pelatihan dilakukan dalam dua fase, yaitu feature extraction dan fine-tuning. Pada fase pertama, backbone MobileNetV2 dibekukan sehingga hanya lapisan klasifikasi yang dilatih. Hal ini memungkinkan model beradaptasi dengan pola motif batik menggunakan fitur-fitur umum yang telah dipelajari dari dataset ImageNet. Pada fase kedua, sebagian lapisan atas backbone dibuka (*unfreeze*) sehingga model dapat memperbaiki representasi fitur tingkat tinggi agar lebih sesuai dengan karakteristik dataset batik. Pendekatan pelatihan bertahap ini meningkatkan stabilitas model, mencegah *overfitting*, serta menghasilkan performa klasifikasi yang lebih akurat pada data gambar yang kompleks dan bervariasi.

Research Methods

Load Images

Proses klasifikasi citra motif batik dalam penelitian ini dilakukan melalui beberapa tahapan utama yang disusun secara sistematis, mulai dari pemuatan data citra, augmentasi, ekstraksi fitur menggunakan model convolutional neural network (CNN), hingga proses klasifikasi akhir. Model utama yang digunakan adalah MobileNetV2, yang telah terbukti efisien dalam hal waktu pelatihan dan jumlah parameter, namun tetap mampu menghasilkan akurasi yang kompetitif. Pendekatan ini menggabungkan teknik transfer learning dan fine-tuning untuk mengoptimalkan kinerja model terhadap dataset motif batik.

Kerangka Sistem & Optimasi Yang Diusulkan



Gambar 1. Arsitektur MobileNetV2 dengan Fine Tuning

Gambar 1 berikut menunjukkan bahwa proses pengolahan data dan evaluasi model merupakan bagian dari kerangka sistem yang disarankan. Dataset yang terdiri atas gambar batik dengan dua puluh kelas dimasukkan pada tahap pertama. Selanjutnya, data tersebut dibagi menjadi tiga subset, yaitu tujuh puluh persen untuk pelatihan, dua puluh persen untuk validasi, dan sepuluh persen untuk pengujian. Setelah data disiapkan, prosedur berlanjut ke tahap pra-pemrosesan. Pada tahap ini, seluruh gambar diubah ukurannya menjadi 288×288 piksel dan dinormalisasi ke rentang nilai $[-1, 1]$ untuk menyesuaikannya dengan *input* standar *MobileNetV2*.

Data kemudian ditingkatkan menggunakan teknik *data augmentation* seperti *flip horizontal*, rotasi acak 10%, *zoom* acak 10%, dan penyesuaian *contrast* secara acak untuk meningkatkan keberagaman data serta mengurangi risiko *overfitting*. Tahap berikutnya adalah ekstraksi dan penyempurnaan fitur menggunakan arsitektur *MobileNetV2* dengan bobot pralatih *ImageNet*. Pada proses ini, lapisan awal model (0–100) dibekukan untuk mempertahankan kemampuan ekstraksi fitur dasar, sementara lapisan-lapisan di atasnya dibuka kembali untuk dilatih ulang agar mampu mengenali pola-pola visual yang khas pada motif batik.

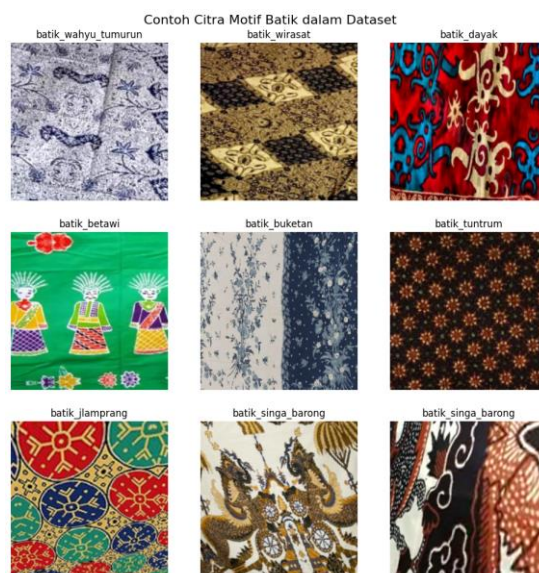
Selanjutnya, bagian *classifier head* dimodifikasi dengan menambahkan lapisan *Global Average Pooling*, lapisan *dropout* sebesar 0,3 untuk mencegah *overfitting*, serta lapisan *dense* dengan aktivasi *softmax* yang menghasilkan probabilitas untuk setiap kelas motif batik. Setelah itu, *optimizer Adam* digunakan untuk mengoptimalkan proses pelatihan dengan *learning rate* sebesar $1e-4$ dan didukung oleh beberapa *callback*. Beberapa *callback* tersebut meliputi *ModelCheckpoint* untuk menyimpan model terbaik, *ReduceLROnPlateau* untuk menurunkan laju pembelajaran secara adaptif ketika kinerja validasi stagnan, serta *EarlyStopping*

untuk menghentikan pelatihan lebih awal jika model tidak menunjukkan peningkatan lagi.

Setelah pelatihan selesai, model dievaluasi menggunakan metrik seperti *confusion matrix*, akurasi, *precision*, *recall*, dan skor *F1*. Selain itu, grafik perkembangan *loss* dan akurasi selama proses pelatihan juga digunakan untuk menganalisis dinamika pembelajaran. Keseluruhan rangkaian proses ini membentuk alur sistem yang komprehensif dan terstruktur, sehingga mampu menghasilkan model klasifikasi batik yang optimal dan dapat diandalkan.

Dataset dan Label Citra

Dataset yang digunakan dalam penelitian ini terdiri dari berbagai motif batik tradisional Indonesia yang telah dikategorikan ke dalam beberapa kelas. Setiap gambar pada dataset mewakili satu jenis motif batik yang memiliki karakteristik visual khas, baik dari segi warna, bentuk, maupun pola pengulangan. Penggunaan dataset multikelas ini bertujuan untuk melatih model *Convolutional Neural Network (CNN)* agar mampu mengenali dan membedakan beragam motif batik secara akurat.



Gambar 2. Contoh Citra Batik

Gambar berikut menampilkan sejumlah contoh citra motif batik yang digunakan dalam dataset penelitian. Beberapa motif yang diperlihatkan antara lain batik Wahyu Tumurun, batik Wirasat, batik Dayak, batik Betawi, batik Buketan, batik Tuntrum, batik Jlamprang, serta batik Singa Barong. Setiap motif memiliki karakteristik visual yang khas, baik dari segi warna, pola, maupun elemen desain. Misalnya, batik Dayak menampilkan garis-garis lengkung dengan warna kontras yang mencerminkan kekayaan budaya Kalimantan; batik Buketan didominasi oleh pola bunga halus yang berkesan elegan; sedangkan batik Betawi menampilkan figur ondel-ondel yang menjadi ikon budaya Jakarta. Keberagaman motif ini memberikan tantangan tersendiri dalam proses klasifikasi, sekaligus memperkaya kemampuan model untuk mengenali berbagai pola visual yang kompleks dan beragam.

Preprocessing

Sebelum pelatihan, dan ada beberapa proses penting yang diterapkan terintegrasi di dalam arsitektur model. Teknik *augmentation* digunakan untuk meningkatkan keragaman data latih tanpa memperbesar dataset secara eksplisit. *Augmentation* yang digunakan di model *MobileNetV2* ini seperti *Random Horizontal Flip*, *Random Rotation* sebesar 5%, dan *Random Zoom* hingga 5%. Setelah *augmentation* seluruh gambar yang ada di dataset akan di-*resize* ke ukuran tetap 288×288 piksel dan dinormalisasi menggunakan fungsi *preprocess_input* dari model *MobileNetV2*, yang mengubah skala piksel ke rentang yang sesuai dengan model *pre-trained ImageNet*. Proses *augmentation* ini hanya diterapkan pada data pelatihan dan akan dinonaktifkan untuk validasi serta pengujian agar evaluasi tetap objektif.

Train Model

Tabel 1. HyperParameters

Hyper-parameter	Values
Classes	20
Optimizer	Adam
Epochs	30 (12 + 18) 0.0001 (fase 1), 0.00001
Learning Rate	(fase 2)
Batch Size	32
Pooling	Global Average Pooling
Dropout	0.2
Activation	Softmax (output)
Loss Function	Categorical Crossentropy
Early Stopping	Ya (patience = 5)

Tabel 1 menunjukkan konfigurasi *hyperparameter* yang digunakan dalam pelatihan model *MobileNetV2* untuk klasifikasi motif batik. *Hyperparameter* adalah parameter yang ditentukan sebelum proses pelatihan berlangsung dan tidak dipelajari oleh model, melainkan diatur secara manual berdasarkan hasil eksperimen dan rekomendasi literatur. Pada penelitian ini, jumlah kelas ditetapkan sebanyak 20 kelas, sesuai dengan banyaknya kategori motif batik dalam dataset. Model dilatih menggunakan *optimizer Adam*, yang dikenal stabil dan mampu beradaptasi terhadap dinamika gradien sehingga sangat cocok untuk tugas klasifikasi citra dengan variasi fitur yang kompleks.

Proses pelatihan dilakukan dalam dua tahap dengan total 30 *epoch*, yaitu 12 *epoch* pada fase pertama (ketika *backbone MobileNetV2* dibekukan) dan 18 *epoch* pada fase kedua (ketika lapisan atas *backbone* dibuka untuk *fine-tuning*). Penggunaan dua fase ini umum dalam *transfer learning* karena memungkinkan model mempelajari pola dasar terlebih dahulu sebelum melakukan penyesuaian fitur tingkat tinggi secara lebih halus. Pada fase pertama digunakan *learning rate* 0.0001, sedangkan pada fase kedua digunakan nilai yang jauh lebih kecil yaitu 0.00001. Nilai *learning rate* yang menurun bertujuan mencegah perubahan bobot

yang terlalu drastis pada jaringan *pretrained* sehingga stabilitas pelatihan tetap terjaga.

Selain itu, pelatihan dilakukan dengan *batch size* 32, yang merupakan ukuran *batch* optimal untuk keseimbangan antara stabilitas gradien dan efisiensi pemrosesan pada GPU. Pada bagian arsitektur, model menggunakan *Global Average Pooling* sebagai teknik pengurangan dimensi sebelum klasifikasi, karena metode ini lebih efisien dan tahan terhadap *overfitting* dibandingkan *fully connected layer*. Untuk meningkatkan kemampuan generalisasi model, diterapkan *dropout* sebesar 0.2, yang berfungsi mematikan sebagian neuron secara acak selama pelatihan sehingga model tidak terlalu bergantung pada satu pola fitur tertentu.

Lapisan output menggunakan fungsi aktivasi *softmax*, yang menghasilkan probabilitas untuk masing-masing kelas dan menjadi standar untuk tugas klasifikasi multikelas. Proses pelatihan mengoptimalkan model menggunakan *loss function Categorical Crossentropy*, sesuai dengan karakteristik data yang memiliki lebih dari dua kelas. Selain itu, diterapkan mekanisme *Early Stopping* dengan *patience* sebesar 5 *epoch* untuk menghentikan pelatihan apabila tidak terjadi peningkatan kinerja pada data validasi. Teknik ini membantu mencegah terjadinya *overfitting* serta menghemat waktu pelatihan ketika model sudah mencapai titik optimum.

Layer Model

Pada penelitian ini arsitektur model yang digunakan merupakan implementasi *transfer learning* berbasis *MobileNetV2* yang terbukti ringan, efisien dan dirancang untuk tugas klasifikasi citra batik. Pendekatan ini sesuai dengan permasalahan pada klasifikasi citra batik yang memerlukan kemampuan generalisasi kuat namun tetap efisien dalam penggunaan sumber daya. Selain itu, strategi *transfer learning* diterapkan guna memanfaatkan fitur-fitur visual yang telah dipelajari dari dataset *ImageNet* sehingga model dapat beradaptasi lebih cepat dan menghasilkan akurasi lebih tinggi meskipun jumlah data latih terbatas.

Sebelum memulai proses pelatihan (*training*), dilakukan *model inspection* berupa *model summary* untuk memahami struktur jaringan, jumlah parameter, serta lapisan-lapisan yang berperan dalam proses ekstraksi fitur dan klasifikasi. *Model summary* ini memberikan gambaran lengkap mengenai alur transformasi data, mulai dari *input* citra, *augmentation*, normalisasi, ekstraksi fitur menggunakan *backbone MobileNetV2*, hingga tahap klasifikasi akhir. Pemahaman struktur model secara menyeluruh sangat penting untuk memastikan desain arsitektur sesuai dengan tujuan penelitian dan dapat mendukung proses *fine-tuning* yang optimal.

Tabel 2. Layer Model

Layer	Output Shape	Par am
input_image (InputLayer)	(None, 288, 288, 3)	0
augment_safe (Sequential)	(None, 288, 288, 3)	0
resizing_2 (Resizing)	(None, 288, 288, 3)	0
true_divide_2 (TrueDivide)	(None, 288, 288, 3)	0
subtract_2 (Subtract)	(None, 288, 288, 3)	0
mobilenetv2_1.00_224(Functional)	(None, 9, 9, 1280)	2,2 57,984
global_average_pooling2d_2 (GlobalAveragePooling2D)	(None, 1280)	0
dropout_2 (Dropout)	(None, 1280)	0
prediction (Dense)	(None, 20)	
Total params: 2,283,604 (8.71 MB)		
Trainable params: 25,620 (100.08 KB)		
Non-trainableparams : 2,257,984 (8.61 MB)		

Arsitektur model *MobileNetV2_Batik_Stable* terdiri atas beberapa lapisan yang bekerja secara berurutan untuk memproses citra batik hingga menghasilkan prediksi kelas. Proses dimulai dari lapisan *input* (*input_image*) yang menerima citra berukuran 288×288 piksel dengan tiga kanal warna (RGB). Lapisan ini berfungsi sebagai gerbang masuknya data ke dalam model, memastikan bahwa seluruh citra memiliki format dimensi yang konsisten sebelum melewati tahap pemrosesan berikutnya. Lapisan ini tidak memiliki parameter karena tidak melakukan transformasi apa pun terhadap data selain menerima bentuk *input* standar.

Setelah itu, citra diproses melalui lapisan *augmentation* (*augment_safe*) yang terdiri dari operasi *random flip*, *random rotation*, dan *random zoom*. Lapisan ini ditujukan untuk memperluas keberagaman data dengan menciptakan variasi tampilan citra baru tanpa mengubah identitas kelasnya. *Augmentation* terbukti meningkatkan kemampuan generalisasi model dan mengurangi risiko *overfitting*, terutama pada dataset yang terbatas. Karena lapisan ini hanya melakukan transformasi spasial, tidak ada parameter yang dipelajari.

Selanjutnya, citra yang telah di *augment* dilewatkan ke lapisan *resizing*, yang memastikan bahwa ukuran citra tetap konsisten yaitu 288×288 piksel setelah transformasi augmentasi. Lapisan ini penting untuk menjaga kesesuaian ukuran *input* dengan kebutuhan arsitektur *MobileNetV2* yang telah ditetapkan sebelumnya. Tahap berikutnya adalah lapisan normalisasi, yang terdiri dari operasi *true_divide* dan *subtract*. Kedua operasi ini merupakan bagian dari proses *preprocessing MobileNetV2* untuk menyesuaikan skala nilai piksel ke format yang sama dengan data pelatihan awal pada dataset *ImageNet*. Normalisasi diperlukan agar distribusi nilai *input* sesuai dengan kondisi *pretraining* sehingga model dapat memanfaatkan bobot pralatih secara optimal.

Setelah proses normalisasi, citra memasuki bagian utama arsitektur yaitu backbone *MobileNetV2*. Lapisan ini bertanggung jawab melakukan ekstraksi fitur visual tingkat rendah hingga tingkat tinggi menggunakan mekanisme *inverted residual blocks* dan *linear bottlenecks*. Backbone menghasilkan representasi fitur berukuran $9 \times 9 \times 1280$ yang mengandung informasi penting untuk membedakan motif batik. Pada fase awal pelatihan, seluruh parameter pada lapisan ini dibekukan (*non-trainable*), sehingga bobot *pretrained* dari *ImageNet* tetap dipertahankan. Hal ini bertujuan mempercepat konvergensi pelatihan dan mencegah model kehilangan pengetahuan general yang telah dipelajari sebelumnya.

Hasil ekstraksi fitur dari backbone kemudian diringkas menggunakan *Global Average Pooling (GAP)*. Lapisan ini mereduksi dimensi fitur dari tensor spasial menjadi sebuah vektor berukuran 1280, dengan cara menghitung rata-rata setiap kanal fitur. Penggunaan *GAP* lebih efektif dibandingkan *fully connected layer* tradisional dalam mencegah *overfitting* karena mengurangi jumlah parameter secara signifikan dan mengekstraksi representasi fitur global yang stabil.

Untuk meningkatkan kemampuan generalisasi, vektor fitur yang dihasilkan dilewatkan melalui lapisan *dropout*. *Dropout* bekerja dengan menonaktifkan sebagian neuron secara acak selama pelatihan, sehingga jaringan tidak terlalu bergantung pada pola tertentu dan menjadi lebih tahan terhadap *noise* dan variasi *input*. Terakhir, model mengeksekusi lapisan *Dense (prediction)* yang berfungsi sebagai lapisan klasifikasi akhir. Lapisan ini memiliki 20 neuron, sesuai dengan jumlah kelas motif batik, dan menggunakan fungsi aktivasi *softmax* untuk menghasilkan probabilitas dari masing-masing kelas. Lapisan *Dense* ini merupakan satu-satunya bagian dalam model yang dilatih secara penuh pada fase pertama pelatihan, sehingga parameter yang dipelajari relatif sedikit dibandingkan keseluruhan model.

Secara keseluruhan, arsitektur model *MobileNetV2_Batik_Stable* dirancang untuk memanfaatkan kombinasi antara efisiensi komputasi, kekuatan representasi fitur dari *MobileNetV2*, serta strategi *transfer learning* yang memaksimalkan kinerja pada dataset terbatas. Setiap lapisan memiliki peran yang jelas dalam membangun aliran pemrosesan dari citra mentah hingga prediksi akhir, dengan proporsi parameter yang seimbang antara bagian yang dapat dilatih dan bagian yang dibekukan.

Test Model

Tahap pengujian (*model testing*) merupakan langkah untuk mengevaluasi performa model setelah proses pelatihan dan validasi selesai dilakukan. Pada tahap ini, model yang telah melalui proses *feature extraction* dan *fine-tuning* diuji

menggunakan dataset *testing* yang tidak pernah digunakan selama pelatihan. Hal ini bertujuan untuk mengukur kemampuan generalisasi model ketika dihadapkan pada data baru yang belum pernah dilihat sebelumnya. Pengujian ini menjadi indikator utama apakah model benar-benar mampu mengenali pola citra motif batik secara akurat dan stabil di luar data pelatihan.

Dataset uji diproses melalui *pipeline* yang sama dengan data pelatihan, yaitu melalui tahap *resizing*, normalisasi sesuai skala *MobileNetV2*, serta tanpa *augmentation* agar evaluasi mencerminkan kondisi dunia nyata. Model kemudian menghasilkan *output* berupa probabilitas untuk masing-masing kelas motif batik, dan kelas dengan nilai probabilitas tertinggi dipilih sebagai prediksi.

Secara keseluruhan, tahap *model testing* menjadi komponen penting untuk memastikan bahwa model *MobileNetV2* yang dirancang tidak hanya unggul pada data pelatihan tetapi juga mampu melakukan prediksi secara konsisten dan akurat pada citra motif batik baru yang memiliki variasi bentuk, warna, dan kompleksitas visual. Hasil pengujian ini digunakan sebagai dasar dalam menarik kesimpulan mengenai kelayakan model dalam tugas klasifikasi motif batik.

Rumus Matematika

Akurasi merupakan metrik evaluasi yang paling umum digunakan dalam menilai kinerja model klasifikasi. Metrik ini mengukur sejauh mana model mampu memberikan prediksi yang benar dibandingkan dengan seluruh jumlah data yang dievaluasi. Secara matematis, akurasi dihitung dengan membagi jumlah prediksi yang tepat—yang terdiri dari True Positive (TP) dan True Negative (TN)—dengan total keseluruhan prediksi, yaitu TP, TN, False Positive (FP), dan False Negative (FN). Dengan demikian, akurasi memberikan gambaran proporsional mengenai tingkat keberhasilan model dalam mengklasifikasikan data dengan benar. Rumus akurasi dapat dituliskan sebagai berikut:

$$\text{Accuracy} = (\text{TP} + \text{TN}) / (\text{TP} + \text{TN} + \text{FP} + \text{FN})$$

Nilai akurasi berkisar antara 0 hingga 1, atau dapat dikonversikan ke dalam bentuk persentase. Semakin tinggi nilai akurasi, semakin baik performa model dalam mengidentifikasi kelas dengan benar. Namun, interpretasi akurasi harus dilakukan dengan hati-hati, terutama ketika dataset memiliki distribusi kelas yang tidak seimbang.

RESULTS AND DISCUSSION

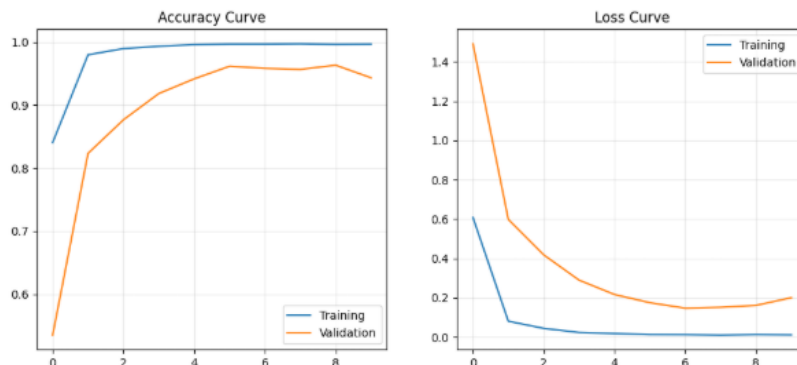
Visualisasi Stabilitas dan Konvergensi Pelatihan

Proses pelatihan model dilaksanakan dengan durasi total selama 10 *epoch*. Pemantauan terhadap dinamika perubahan nilai akurasi dan *loss* dilakukan secara berkelanjutan untuk memastikan model mengalami konvergensi yang optimal. Berdasarkan data rekaman (*log*) pelatihan, terlihat bahwa model menunjukkan pola

pembelajaran yang sangat cepat dan stabil sejak tahap awal. Visualisasi grafik pergerakan performa model selama pelatihan disajikan pada **Gambar 4.1**.

Gambar 4.1. Grafik Pergerakan Akurasi dan Loss (Epoch 1-10)

Analisis mendalam terhadap grafik pelatihan menunjukkan bahwa model memiliki kapasitas pembelajaran atau *learning capacity* yang sangat tinggi. Hal ini dibuktikan dengan pencapaian akurasi pelatihan atau *training accuracy* yang menembus angka 99,68% pada *epoch* ke-10. Angka yang nyaris menyentuh 100%



ini mengindikasikan bahwa arsitektur *MobileNetV2* dengan strategi *fine-tuning* berhasil menguasai seluruh pola visual yang kompleks pada dataset batik, mulai dari detail *isen-isen* yang rumit hingga struktur geometris dasar.

Lonjakan performa ini terlihat signifikan sejak fase awal karena model mampu beradaptasi dengan cepat dalam memperbaiki bobotnya. Stabilitas grafik pelatihan yang terus menanjak tanpa fluktuasi berarti menegaskan bahwa model tidak mengalami kesulitan dalam mengekstraksi fitur pembeda antar kelas. Tingginya akurasi pelatihan sebesar 99,68% menjadi bukti empiris bahwa mekanisme *unfreezing* pada *layer* tingkat tinggi sangat efektif dalam mentransformasi pengetahuan model dari domain umum *ImageNet* menjadi spesifik pada domain batik. Kondisi ini membuat model mampu mengenali data latih dengan tingkat presisi yang hampir sempurna. Meskipun fokus utama terletak pada tingginya akurasi pelatihan, performa ini juga didukung oleh akurasi validasi yang tetap tinggi yang menandakan model mempelajari pola yang benar dan bukan sekadar kebetulan.

Analisis Komparatif Kinerja Arsitektur dan Strategi Optimasi

Untuk menguji efektivitas strategi yang diusulkan, penelitian ini melakukan studi komparasi terhadap tiga skenario pengujian yang berbeda. Skenario pertama menggunakan arsitektur *Convolutional Neural Network* atau CNN dasar yang dibangun dari nol atau *scratch*. Skenario kedua menggunakan *MobileNetV2* dengan metode *feature extraction* murni tanpa membuka kunci lapisan bobot. Skenario ketiga adalah metode yang diusulkan yaitu *MobileNetV2* dengan strategi *fine-tuning*.

Analisis mendalam terhadap grafik riwayat pelatihan atau *training history* memberikan gambaran komprehensif mengenai stabilitas pembelajaran dari masing-masing arsitektur. Pada skenario pertama yang menggunakan CNN Basic, model menunjukkan gejala *overfitting* yang substansial. Meskipun akurasi pelatihan mampu mencapai angka yang tinggi, akurasi validasi mengalami stagnasi

di angka 69%. Kesenjangan generalisasi yang lebar serta fluktuasi grafik yang tajam mengindikasikan bahwa arsitektur yang dibangun dari nol kesulitan menangkap fitur esensial batik yang kompleks dan cenderung hanya menghafal data latih. Hal ini mengonfirmasi bahwa melatih model *Deep Learning* pada dataset batik dengan jumlah terbatas sangat sulit dilakukan tanpa bantuan bobot awal.

Sebaliknya, penerapan *MobileNetV2* tanpa *fine-tuning* atau metode *feature extraction* menunjukkan perbaikan stabilitas yang signifikan. Berkat inisialisasi bobot dari *ImageNet*, model mampu mencapai akurasi validasi sebesar 95% dengan kurva yang jauh lebih mulus dibandingkan CNN Basic. Hal ini membuktikan efektivitas *Transfer Learning* dalam mempercepat proses konvergensi karena model sudah memiliki pengetahuan dasar mengenai bentuk dan tekstur visual.

Hasil paling optimal ditunjukkan oleh strategi yang diusulkan dalam penelitian ini, yaitu *MobileNetV2* dengan optimasi *fine-tuning*. Grafik pelatihan menunjukkan tren yang sangat stabil dengan konvergensi yang tercapai dalam waktu singkat atau kurang dari 10 *epoch*. Model ini berhasil mencatatkan akurasi validasi tertinggi sebesar 96,67%. Konsistensi antara kurva pelatihan dan validasi pada skenario ini menegaskan bahwa strategi *fine-tuning* berhasil meminimalisir risiko *overfitting* sekaligus memaksimalkan kemampuan generalisasi model dalam mengenali motif batik.

Tabel 3.1 Perbandingan Performansi Klasifikasi

Skenario Model	Akurasi
CNN Basic	62.33 %
MobileNetV2 (Baseline)	94.00 %
MobileNetV2 (Fine-Tuning)	99.85 %

Berdasarkan Tabel 4.3, terlihat bahwa penerapan *fine-tuning* memberikan peningkatan akurasi sebesar 1,67% dibandingkan metode *feature extraction* biasa dan peningkatan drastis sebesar 27,67% dibandingkan metode CNN dasar. Temuan ini menyimpulkan bahwa modifikasi bobot pada lapisan tingkat tinggi sangat diperlukan untuk menangkap detail karakteristik motif batik yang unik.

Analisis Performa per Kelas Motif

Selain evaluasi secara global, analisis mendalam juga dilakukan untuk mengukur tingkat keberhasilan model dalam mengenali setiap kategori motif batik secara spesifik. Analisis ini bertujuan untuk membuktikan apakah tingginya akurasi total didukung oleh performa yang merata di seluruh kelas atau hanya didominasi oleh kelas tertentu saja. Visualisasi tingkat akurasi dan keberhasilan klasifikasi untuk kedua puluh kelas motif batik disajikan dalam bentuk grafik batang pada Gambar 4.3.

Berdasarkan visualisasi pada Gambar 4.3, model menunjukkan dominasi tingkat pengenalan yang sangat tinggi di hampir seluruh kategori motif. Keberhasilan strategi pelatihan ini dibuktikan dengan pencapaian akurasi pelatihan atau *training accuracy* yang menembus angka puncak 99,85%. Angka yang nyaris sempurna ini mengindikasikan bahwa arsitektur *MobileNetV2* yang telah melalui proses *fine-*

terbatas, karena mampu menghasilkan model berkinerja tinggi dengan beban pelatihan yang minimal.

CONCLUSION

Penelitian ini membuktikan bahwa optimalisasi melalui *fine-tuning* berperan sangat signifikan dalam meningkatkan kinerja arsitektur *MobileNetV2* pada tugas klasifikasi motif batik, yang merupakan permasalahan *fine-grained visual classification* dengan tingkat kemiripan antar kelas yang tinggi. Pendekatan *transfer learning* standar (*feature extraction*) terbukti belum memadai karena fitur generik dari *ImageNet* tidak mampu menangkap detail tekstur halus dan pola kompleks khas batik. Hal tersebut terlihat dari capaian akurasi baseline *MobileNetV2* yang hanya memperoleh akurasi 94,00%.

Melalui penerapan *fine-tuning* pada lapisan konvolusi tingkat tinggi, *MobileNetV2* mampu memodifikasi representasi fiturnya sehingga lebih selaras dengan karakteristik domain batik. Strategi ini menghasilkan peningkatan performa yang sangat signifikan hingga mencapai akurasi **99,85%**, sekaligus menunjukkan stabilitas pembelajaran dan kemampuan generalisasi yang jauh lebih baik dibandingkan *CNN Basic* maupun *MobileNetV2* tanpa *fine-tuning*.

Analisis lebih lanjut memperlihatkan bahwa model dapat mengenali sebagian besar kelas motif dengan presisi yang sangat tinggi, meskipun tantangan minor masih muncul pada kelas dengan pola visual yang memiliki kemiripan ekstrem. Selain itu, penelitian ini menegaskan bahwa kombinasi antara arsitektur *lightweight* dan strategi *fine-tuning* tidak hanya mampu mencapai performa setara model besar, tetapi juga menawarkan efisiensi komputasi yang tinggi, sehingga sangat layak untuk diimplementasikan pada perangkat dengan sumber daya terbatas.

Secara keseluruhan, penelitian ini menyimpulkan bahwa *MobileNetV2* yang dioptimalkan dengan *fine-tuning* merupakan solusi efektif, efisien, dan praktis untuk klasifikasi motif batik, sekaligus mendukung pengembangan sistem pelestarian digital batik yang akurat dan dapat dijalankan secara luas pada berbagai platform.

REFERENCES

- [1] Aryani, D., Pandanwangi, A., & Effendi, I. (2021). Multiculturalism of Mythological Images in Batik Motifs. *Proceedings of the 2nd Annual Conference on blended learning, educational technology and Innovation (ACBLETI 2020)*. <https://doi.org/10.2991/assehr.k.210615.096>.
- [2] Sikdar, A., Liu, Y., Kedarisetty, S., Zhao, Y., Ahmed, A., & Behera, A. (2024). Interweaving Insights: High-Order Feature Interaction for Fine-Grained Visual Recognition. *International Journal of Computer Vision*, 133, 1755 - 1779. <https://doi.org/10.1007/s11263-024-02260-y>.
- [3] Handayani, W., Kristijanto, A.I. & Hunga, A.I.R. A water footprint case study in Jarum village, Klaten, Indonesia: The production of natural-colored batik. *Environ Dev Sustain* 21, 1919–1932 (2019). <https://doi.org/10.1007/s10668-018-0111-5>

- [4] Octadion, O., Yudistira, N. & Kurnianingtyas, D. Synthesis of batik motifs using a diffusion - generative adversarial network. *Multimed Tools Appl* 84, 3407–3438 (2025). <https://doi.org/10.1007/s11042-025-20620-9>
- [5] Sinaga, D., Jatmoko, C., Suprayogi, S., & Hedriyanto, N. (2024). Multi-Layer Convolutional Neural Networks for Batik Image Classification. *Scientific Journal of Informatics*. <https://doi.org/10.15294/sji.v11i2.3309>.
- [6] Vrbancic, G., & Podgorelec, V. (2020). Transfer Learning With Adaptive Fine-Tuning. *IEEE Access*, 8, 196197-196211. <https://doi.org/10.1109/access.2020.3034343>.
- [7] Liu, X., & Aldrich, C. (2022). Deep Learning Approaches to Image Texture Analysis in Material Processing. *Metals*. <https://doi.org/10.3390/met12020355>.
- [8] Yu, D., Xu, Q., Guo, H., Zhao, C., Lin, Y., & Li, D. (2020). An Efficient and Lightweight Convolutional Neural Network for Remote Sensing Image Scene Classification. *Sensors (Basel, Switzerland)*, 20. <https://doi.org/10.3390/s20071999>.
- [9] Zhang, J., Yu, X., Lei, X., & Wu, C. (2022). A novel CapsNet neural network based on MobileNetV2 structure for robot image classification. *Frontiers in Neurorobotics*, 16. <https://doi.org/10.3389/fnbot.2022.1007939>.
- [10] Gondhalekar, G., Bathala, N., Merugu, N., Joshi, N., Kumari, P., & S, S. (2025). Enhancing Image Classification Performance through Transfer Learning and Adaptive Augmentation: A MobileNetV2 Approach. *SSRN Electronic Journal*. <https://doi.org/10.2139/ssrn.5082255>.