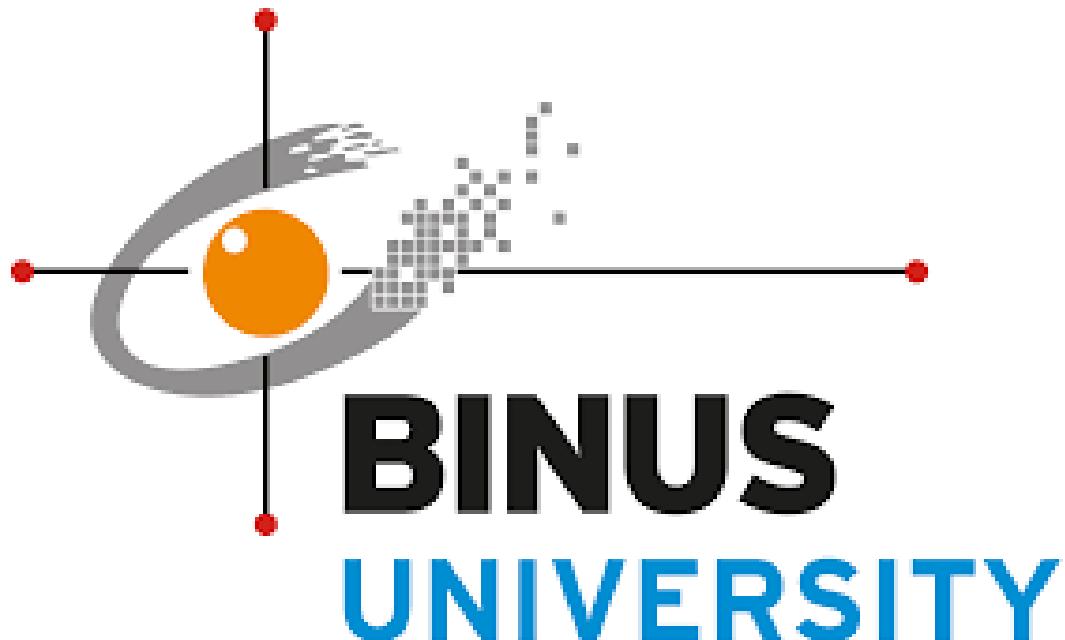


DEEP LEARNING-BASED MULTICLASS CLASSIFICATION FOR INDONESIAN BATIK PATTERNS



Oleh:

Davin Edbert Santoso Halim 2602067086

Felicia Andrea Tandoko 2602059342

Steve Marcello Liem 2602071410

Dosen:

Lili Ayu Wulandari, S. Si., M. Sc., Ph.D.

Mata Kuliah:

Deep Learning

Kelas:

LA-09

Tahun Ajaran:

Semester Ganjil 2024/2025

PENDAHULUAN

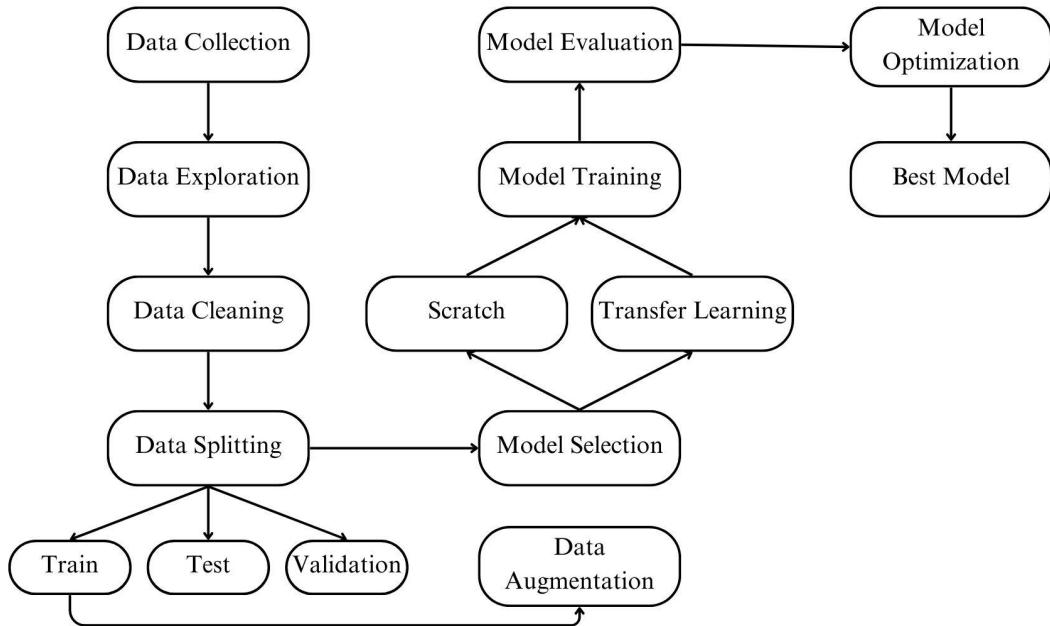
Batik merupakan salah satu warisan budaya Indonesia yang kaya akan nilai seni dan sejarah. Kekayaan budaya dan keragaman etnis di Indonesia tercermin dari berbagai motif batik yang memiliki keunikannya masing-masing. Namun, terdapat salah satu tantangan dalam mempromosikan dan melestarikan batik, yaitu kemampuan untuk mengenali dan mengklasifikasikan berbagai motif batik dengan akurat. Dalam era *digital* ini, penggunaan teknologi, khususnya *deep learning*, dapat memberikan solusi efektif untuk masalah tersebut, yaitu pengenalan motif batik secara otomatis.

Oleh karena itu, kami akan melakukan klasifikasi gambar dengan *deep learning* untuk tiga motif batik yang sangat dikenal oleh banyak orang, yaitu Parang yang sudah ada sejak zaman Keraton Mataram, Mega Mendung yang berasal dari Cirebon, dan Kawung yang berasal dari Yogyakarta. Kami akan melatih model *deep learning* untuk mengklasifikasikan gambar motif batik ke dalam salah satu dari tiga kelas tersebut. Dataset yang kami gunakan diambil dari kaggle (<https://www.kaggle.com/datasets/dionisiusdh/indonesian-batik-motifs>) yang menyediakan berbagai gambar motif batik.

Proses klasifikasi ini akan melibatkan beberapa model *deep learning*, baik yang dibangun dari awal (*scratch*) maupun yang menggunakan teknik *transfer learning*. *Transfer learning* memungkinkan kita untuk memanfaatkan model yang telah dilatih pada dataset besar dan mengaplikasikannya pada dataset yang lebih kecil, yang seringkali menghasilkan performa yang lebih baik. Untuk menemukan model terbaik, kami juga akan melakukan berbagai *tuning* pada *hyperparameter* model.

Analisa awal terhadap data menunjukkan bahwa ketiga motif batik ini memiliki pola visual yang cukup berbeda satu sama lain. Motif Parang biasanya memiliki pola yang memanjang atau garis miring yang berulang, motif Mega Mendung cenderung memiliki pola awan yang melengkung dan mengalir, sementara motif Kawung memiliki pola melingkar yang teratur dan simetris. Perbedaan-perbedaan ini memberikan dasar yang kuat bagi model *deep learning* untuk belajar dan mengklasifikasikan setiap motif dengan tepat. Dengan menggunakan teknik *deep learning* yang tepat, diharapkan model yang dihasilkan dapat mengenali dan mengklasifikasikan ketiga motif batik tersebut dengan akurasi yang tinggi.

METODOLOGI



Data Collection

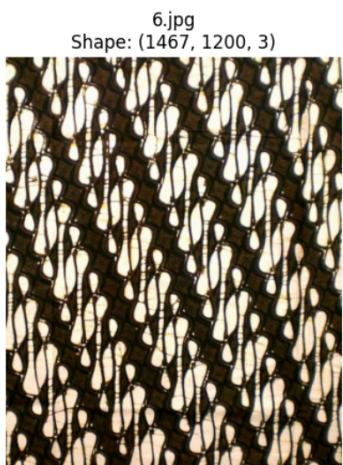
Langkah pertama dalam tugas klasifikasi gambar motif batik ini adalah mengumpulkan data yang akan digunakan. Kami mengambil data dari public dataset Kaggle (<https://www.kaggle.com/datasets/dionisiusdh/indonesian-batik-motifs>). Dataset ‘Indonesian Batik Motifs’ ini memiliki 983 gambar yang mencakup 20 motif batik Indonesia. Motif-motif batik yang terdapat pada dataset ini antara lain: Batik Bali, Batik Betawi, Batik Celup, Batik Cendrawasih, Batik Ceplok, Batik Ciamis, Batik Garutan, Batik Gentongan, Batik Kawung, Batik Keraton, Batik Lasem, Batik Mega Mendung, Batik Parang, Batik Pekalongan, Batik Priangan, Batik Sekar, Batik Sidoluhur, Batik Sidomukti, Batik Sogan, dan Batik Tambal.

Untuk keperluan tugas ini, kami hanya memilih tiga motif batik dari 20 motif yang tersedia, yaitu Batik Mega Mendung, Batik Parang, dan Batik Kawung. Pemilihan ketiga motif ini didasarkan pada popularitas dan pola visual yang berbeda-beda, yang diharapkan dapat memberikan tantangan yang menarik dan relevan untuk model klasifikasi yang akan kami kembangkan.

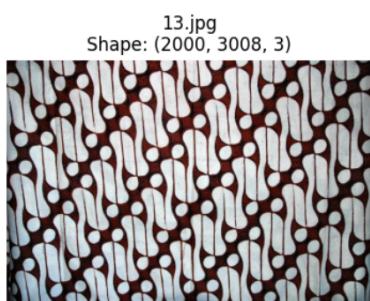
Data Exploration

Setelah mengumpulkan data, langkah berikutnya adalah mengeksplorasi dataset untuk memahami karakteristik dan distribusi data. Dalam dataset ini, kami memiliki jumlah gambar yang berbeda untuk setiap kelas motif batik. Terdapat 50 gambar untuk Batik Parang, 47 gambar untuk Batik Mega Mendung, dan 45 gambar untuk Batik Kawung.

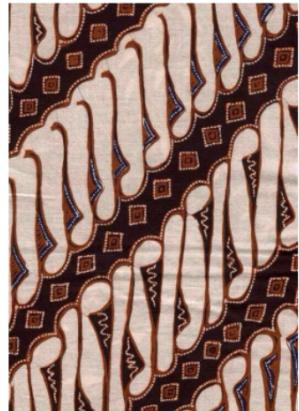
Berikut adalah beberapa contoh gambar dari data kami:



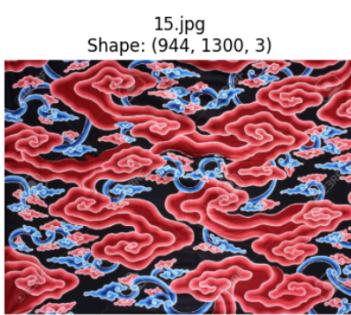
3 Random Images from batik-parang



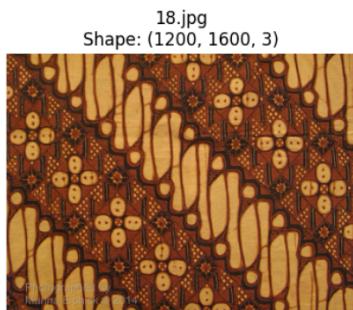
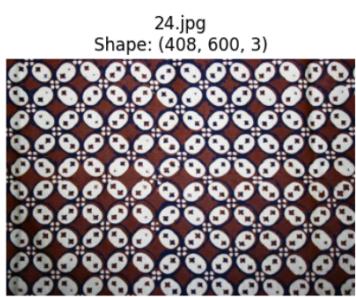
23.jpg
Shape: (748, 526, 3)



3 Random Images from batik-megamendung



3 Random Images from batik-kawung

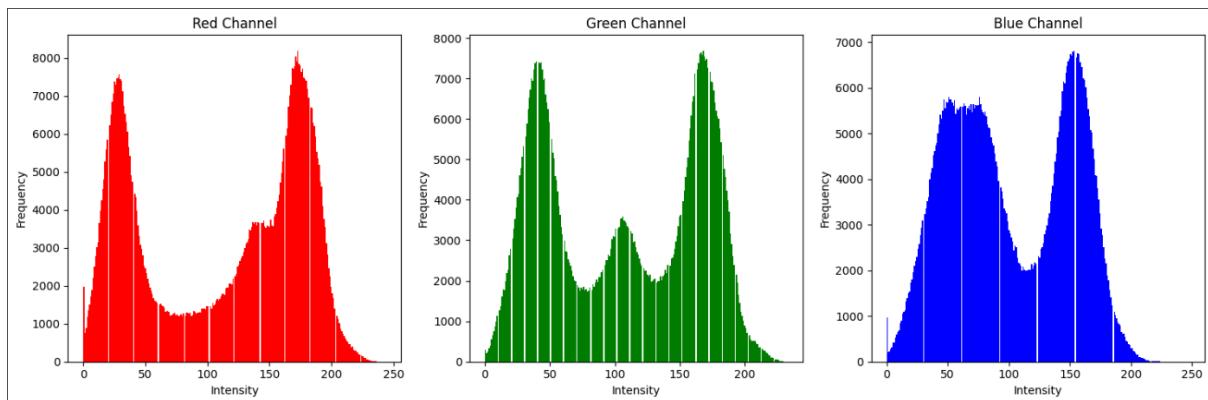


Kami juga menganalisis ukuran gambar dalam dataset. Ukuran minimum gambar adalah (229, 220) piksel, sementara ukuran maksimum adalah (3264, 2448) piksel. Ukuran rata-rata gambar dalam dataset ini adalah (992.48, 885.76) piksel. Analisis ini penting karena variasi ukuran gambar yang besar memerlukan penyesuaian ukuran (resizing) sebelum pelatihan model. Penyesuaian ukuran ini memastikan bahwa semua gambar memiliki dimensi yang konsisten, sehingga model dapat lebih mudah mengenali pola tanpa terganggu oleh perbedaan resolusi.

Selain itu, kami melakukan analisis tekstur pada gambar menggunakan Gray Level Co-occurrence Matrix (GLCM) untuk menganalisa contrast, dissimilarity, homogeneity, energy, dan correlation:

- Contrast: 140.271, menunjukkan tingkat variasi intensitas antara piksel yang berdekatan. Nilai yang tinggi menunjukkan adanya perbedaan tajam dalam intensitas, yang umum dalam pola batik.
- Dissimilarity: 7.593, mengukur perbedaan antara piksel yang berdekatan. Nilai yang lebih tinggi menunjukkan adanya variasi yang signifikan dalam pola.
- Homogeneity: 0.617, mengukur keseragaman pola dalam gambar. Nilai yang rendah menunjukkan bahwa gambar memiliki variasi pola yang kompleks.
- Energy: 0.016, mengukur kekonsistenan dan intensitas piksel dalam gambar. Nilai yang lebih rendah biasanya menunjukkan pola yang lebih rumit.
- Correlation: 0.980, menunjukkan sejauh mana piksel yang berdekatan memiliki hubungan linear. Nilai yang tinggi menunjukkan bahwa ada pola yang berulang dan teratur dalam gambar.

Kami juga melakukan analisis histogram warna untuk memahami distribusi warna dalam gambar. Histogram warna yang ditunjukkan dalam tiga channel (Red, Green, Blue) menunjukkan frekuensi intensitas warna di masing-masing channel. Berikut adalah hasil dari histogram warna:

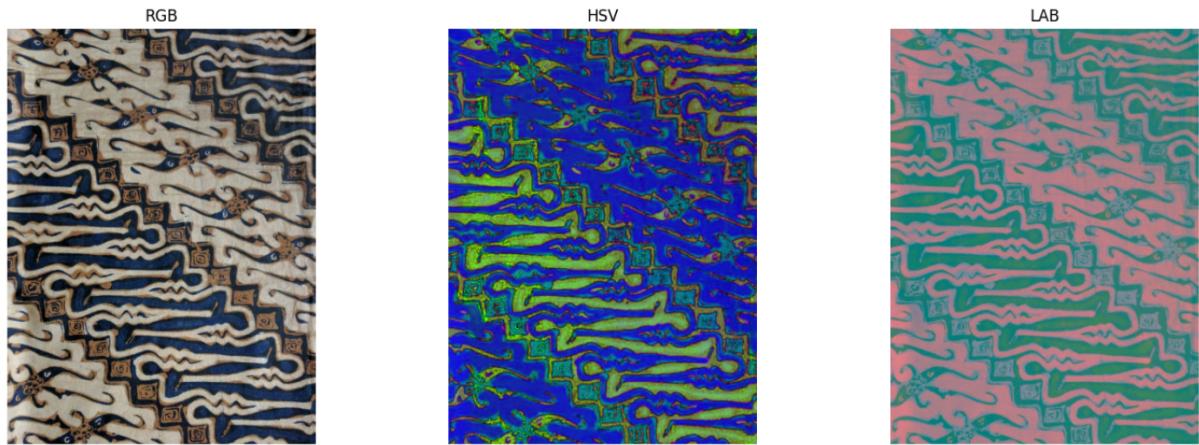


- Red Channel: Histogram menunjukkan dua puncak frekuensi tinggi, satu di intensitas rendah (sekitar 0-50) dan satu lagi di intensitas tinggi (sekitar 150-200). Ini mengindikasikan bahwa ada banyak piksel dengan warna merah gelap atau mendekati hitam serta warna merah yang lebih terang. Distribusi intensitasnya cukup merata di seluruh rentang, dengan beberapa variasi di antara puncak tersebut.
- Green Channel: Mirip dengan channel merah, histogram pada channel hijau juga menunjukkan dua puncak frekuensi tinggi, satu di intensitas rendah (sekitar 0-50) dan satu lagi di intensitas tinggi (sekitar 150-200). Ini menunjukkan banyaknya piksel hijau gelap serta warna hijau yang lebih terang. Distribusi intensitas pada channel hijau sedikit lebih rendah dibandingkan dengan channel merah, tetapi tetap menunjukkan pola yang serupa.
- Blue Channel: Channel biru menunjukkan pola yang serupa dengan dua puncak frekuensi tinggi pada intensitas rendah (sekitar 0-50) dan tinggi (sekitar 150-200). Ini mengindikasikan banyaknya piksel biru gelap serta warna biru yang lebih terang.

Distribusi intensitas pada channel biru juga cukup merata di antara puncak-puncak tersebut.

Distribusi warna ini memberikan informasi tentang intensitas dan frekuensi warna tertentu dalam motif batik, yang bisa menjadi fitur penting bagi model untuk mengenali pola.

Kami juga memvisualisasikan gambar dalam berbagai ruang warna, seperti RGB, HSV, dan LAB, untuk melihat bagaimana pola dan warna motif batik dapat bervariasi dalam representasi warna yang berbeda:



- RGB (Red, Green, Blue): Visualisasi ini menunjukkan gambar dalam ruang warna standar yang paling umum digunakan. Ini membantu dalam memahami representasi warna alami dari motif batik. Warna-warna yang dominan adalah biru, putih, dan coklat, dengan pola geometris yang khas dari motif batik.
- HSV (Hue, Saturation, Value): Dalam ruang warna HSV, gambar lebih menonjolkan perbedaan hue (warna), saturation (kejemuhan), dan value (kecerahan). Ini membantu dalam mengidentifikasi pola warna yang mungkin kurang terlihat dalam ruang RGB.
- LAB: Ruang warna LAB memisahkan informasi kecerahan (L) dari informasi warna (A dan B). Ini sering digunakan dalam pengolahan gambar karena mendekati cara manusia melihat warna dan dapat menunjukkan perbedaan warna yang lebih halus.

Data Cleaning

Pada tahap ini, kami melakukan proses cleaning atau pembersihan data untuk memastikan bahwa dataset yang digunakan dalam pelatihan model bebas dari kesalahan dan inkonsistensi. Dalam proses ini, kami menemukan bahwa terdapat satu file data yang corrupt pada kelas Batik Mega Mendung. Oleh karena itu, kami melakukan cleaning secara manual dengan menghapus file yang corrupt tersebut dari dataset.

Setelah cleaning, jumlah total gambar dalam dataset berubah. Sebelumnya, jumlah gambar untuk Batik Mega Mendung adalah 47, namun setelah penghapusan file corrupt, jumlahnya menjadi 46. Dengan demikian, distribusi data terbaru dalam dataset kami adalah sebagai berikut:

- Batik Parang: 50 gambar
- Batik Mega Mendung: 46 gambar
- Batik Kawung: 45 gambar

Data Splitting

Setelah proses pembersihan data selesai, langkah selanjutnya adalah membagi dataset menjadi tiga set: train set, validation set, dan test set. Proses splitting atau pembagian ini penting untuk memastikan model dapat dilatih dan dievaluasi dengan baik pada data yang terpisah, sehingga mengurangi risiko overfitting dan memberikan prediksi yang lebih akurat terhadap performa model pada data yang tidak terlihat sebelumnya (test set). Kami membagi data dengan rasio 80% untuk training, 10% untuk validation, dan 10% untuk testing.

Distribusi data setelah splitting adalah sebagai berikut:

- Train set:
 - Batik Parang: 48 gambar
 - Batik Mega Mendung: 36 gambar
 - Batik Kawung: 36 gambar
- Validation set:
 - Batik Parang: 5 gambar
 - Batik Mega Mendung: 4 gambar
 - Batik Kawung: 4 gambar
- Test set:
 - Batik Parang: 5 gambar
 - Batik Mega Mendung: 6 gambar
 - Batik Kawung: 5 gambar

Dengan splitting ini, total gambar dalam train set adalah 120, validation set adalah 13, dan test set adalah 16.

Data Augmentation

Setelah membagi dataset, langkah selanjutnya adalah melakukan augmentasi data pada train set. Data augmentasi adalah proses mengaplikasikan transformasi tertentu pada gambar, seperti rotasi, pemotongan, penskalaan, dan perubahan intensitas warna, untuk menghasilkan variasi baru dari data yang ada. Tujuan dari augmentasi ini adalah untuk meningkatkan jumlah dan keragaman training data, yang dapat membantu model untuk belajar lebih baik dan menggeneralisasi pola dari data yang lebih bervariasi.

Dalam tugas ini, kami hanya melakukan augmentasi pada data train set. Hal ini karena tujuan utama dari augmentasi adalah untuk meningkatkan kemampuan model dalam mengenali dan mengklasifikasikan pola dengan lebih baik. Validation set dan test set tidak di augmentasi agar tetap representatif terhadap data nyata yang tidak terlihat oleh model selama pelatihan. Dengan demikian, evaluasi performa model pada data validation dan test dapat memberikan gambaran yang lebih akurat tentang bagaimana model akan bekerja pada data yang sebenarnya.

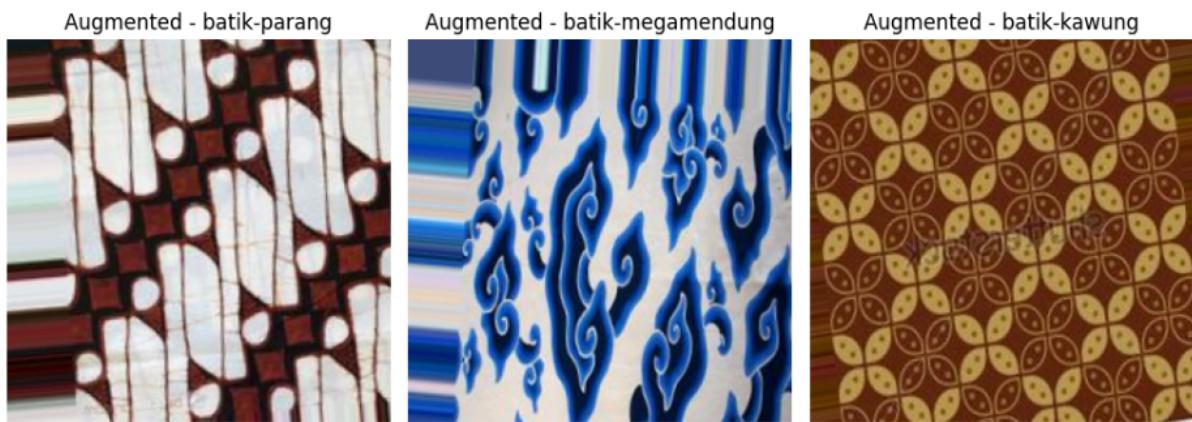
Untuk augmentasi data pada train set, kami menerapkan berbagai transformasi, termasuk:

- Rescaling: Mengubah skala pixel gambar menjadi antara 0 dan 1 dengan cara membagi nilai pixel dengan 255. Ini penting karena gambar digital biasanya memiliki

nilai pixel dalam rentang 0 hingga 255. Hal ini memastikan bahwa data yang dimasukkan ke dalam model memiliki rentang nilai yang konsisten dan standar sehingga dapat mempercepat dan menstabilkan proses pelatihan.

- Rotation: Memutar gambar secara acak hingga 20 derajat.
- Width Shift: Menggeser gambar secara horizontal hingga 20% dari lebar gambar.
- Height Shift: Menggeser gambar secara vertikal hingga 20% dari tinggi gambar.
- Shear: Menerapkan transformasi geser secara acak hingga 20%.
- Zoom: Memperbesar atau memperkecil gambar secara acak hingga 20%.
- Horizontal Flip: Membalik gambar secara horizontal secara acak.
- Fill Mode: Mengisi pixel kosong yang mungkin muncul setelah transformasi dengan metode 'nearest'.

Berikut adalah contoh hasil augmentasi pada data train set kami:



Setelah melakukan augmentasi, persebaran kelas data train set kami juga meningkat.

- Persebaran Kelas Sebelum Augmentasi:
 - Batik Parang: 40 gambar
 - Batik Mega Mendung: 36 gambar
 - Batik Kawung: 36 gambar
- Persebaran Kelas Setelah Augmentasi:
 - Batik Parang: 280 gambar
 - Batik Mega Mendung: 252 gambar
 - Batik Kawung: 252 gambar

Model Selection

Setelah melakukan augmentasi data, langkah selanjutnya adalah memilih model deep learning yang akan digunakan untuk tugas klasifikasi motif batik ini. Kami menggunakan dua pendekatan utama dalam pemilihan model: model dari awal (scratch) dan model transfer learning.

Model scratch berarti kami membangun arsitektur model deep learning dari awal. Hal ini memungkinkan kami untuk merancang model yang spesifik dan disesuaikan dengan karakteristik dataset motif batik kami. Kami dapat menentukan jumlah lapisan, jenis lapisan

(convolutional, pooling, fully connected), dan hyperparameter lainnya seperti learning rate, batch size, dan optimizer yang digunakan. Namun, training model scratch biasanya membutuhkan waktu dan sumber daya komputasi yang lebih besar, terutama jika dataset yang digunakan relatif kecil. Model juga mungkin memerlukan lebih banyak waktu untuk mencapai kinerja yang optimal.

Sedangkan transfer learning berarti kami menggunakan model yang telah dilatih sebelumnya pada dataset besar (seperti ImageNet) dan menyesuaikannya dengan dataset motif batik kami. Kami mengganti lapisan-lapisan terakhir dari model dengan lapisan-lapisan baru yang sesuai dengan jumlah kelas dalam tugas klasifikasi kami, yaitu tiga kelas motif batik. Transfer learning memungkinkan kami untuk memanfaatkan pengetahuan yang telah dipelajari oleh model sebelumnya, sehingga dapat meningkatkan kinerja model secara signifikan dengan lebih sedikit data dan waktu pelatihan. Transfer learning ini juga dapat memberikan hasil yang lebih baik karena model telah memiliki kemampuan untuk mengenali fitur-fitur umum dalam gambar.

Scratch:

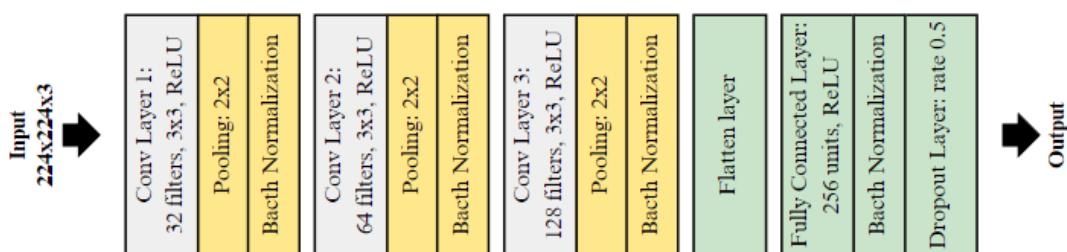
1. Model 1 (no augment)

Arsitektur Model scratch 1 ini dirancang dengan menggunakan beberapa lapisan konvolusi dan pooling yang bertujuan untuk mengekstraksi fitur dari gambar motif batik. Model ini terdiri dari:



- Dimulai dengan lapisan konvolusi pertama yang memiliki 32 filter dengan kernel berukuran 3x3 dan menggunakan fungsi aktivasi ReLU. Lapisan ini menerima input gambar berukuran 224x224 piksel dengan 3 saluran warna (RGB).
- Setelah itu, terdapat lapisan pooling dengan ukuran 2x2 untuk mengurangi dimensi fitur dan mengendalikan overfitting.
- Selanjutnya, lapisan konvolusi kedua memiliki 64 filter dengan kernel berukuran 3x3 dan juga menggunakan fungsi aktivasi ReLU, diikuti oleh lapisan pooling dengan ukuran 2x2.
- Kemudian, terdapat lapisan konvolusi ketiga yang memiliki 128 filter dengan kernel berukuran 3x3 dan fungsi aktivasi ReLU, diikuti oleh lapisan pooling dengan ukuran 2x2.
- Lapisan konvolusi keempat juga memiliki 128 filter dengan kernel berukuran 3x3 dan fungsi aktivasi ReLU, diikuti oleh lapisan pooling terakhir dengan ukuran 2x2.

- Setelah proses ekstraksi fitur selesai, output dari lapisan konvolusi dan pooling ini diubah menjadi vektor 1D oleh lapisan flatten.
 - Vektor ini kemudian dihubungkan ke lapisan fully connected dengan 512 unit dan fungsi aktivasi ReLU.
 - Untuk mengurangi risiko overfitting, terdapat lapisan dropout dengan rate 0.5.
 - Terakhir, model ini memiliki lapisan output yang sesuai dengan jumlah kelas yang ada dalam tugas klasifikasi dan menggunakan fungsi aktivasi softmax untuk menghasilkan probabilitas klasifikasi untuk setiap kelas.
2. Model 1 (with augment)
- Arsitektur Model Scratch 1 dengan augmentasi memiliki struktur yang sama dengan Model 1 tanpa augmentasi. Namun, dalam model ini, train data akan mengalami augmentasi untuk meningkatkan variasi dan keragaman data, yang dapat membantu model belajar lebih baik dan menggeneralisasi pola dari data yang lebih bervariasi.
3. Model 2 (no augment)
- Arsitektur Model scratch 2 dirancang untuk meningkatkan stabilitas dan kinerja model dengan menambahkan lapisan normalisasi batch di setiap lapisan konvolusi. Model ini terdiri dari:



- Dimulai dengan lapisan konvolusi pertama yang memiliki 32 filter dengan kernel berukuran 3x3 dan menggunakan fungsi aktivasi ReLU, diikuti oleh lapisan normalisasi batch untuk menstabilkan proses pelatihan dan mempercepat konvergensi.
- Setelah itu, terdapat lapisan pooling dengan ukuran 2x2 untuk mengurangi dimensi fitur.
- Lapisan konvolusi kedua memiliki 64 filter dengan kernel berukuran 3x3 dan fungsi aktivasi ReLU, diikuti oleh lapisan normalisasi batch dan lapisan pooling dengan ukuran 2x2.
- Selanjutnya, lapisan konvolusi ketiga memiliki 128 filter dengan kernel berukuran 3x3 dan fungsi aktivasi ReLU, diikuti oleh lapisan normalisasi batch dan lapisan pooling dengan ukuran 2x2.
- Setelah proses ekstraksi fitur, output dari lapisan konvolusi dan pooling ini diubah menjadi vektor 1D oleh lapisan flatten.
- Vektor ini kemudian dihubungkan ke lapisan fully connected dengan 256 unit dan fungsi aktivasi ReLU, diikuti oleh lapisan normalisasi batch untuk lebih lanjut menstabilkan pelatihan.
- Untuk mengurangi risiko overfitting, terdapat lapisan dropout dengan rate 0.5.

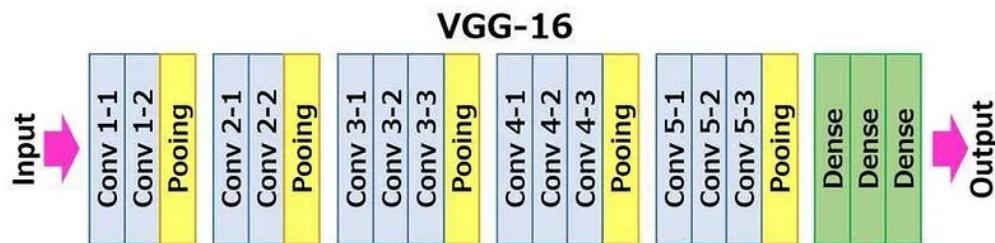
- Terakhir, model ini memiliki lapisan output yang sesuai dengan jumlah kelas yang ada dalam tugas klasifikasi dan menggunakan fungsi aktivasi softmax untuk menghasilkan probabilitas klasifikasi untuk setiap kelas.
4. Model 2 (with augment)

Arsitektur Model Scratch 2 dengan augmentasi data memiliki struktur yang sama dengan Model 2 tanpa augmentasi. Namun, dalam model ini, train data akan mengalami augmentasi untuk meningkatkan variasi dan keragaman data, yang dapat membantu model belajar lebih baik dan menggeneralisasi pola dari data yang lebih bervariasi.

Transfer Learning:

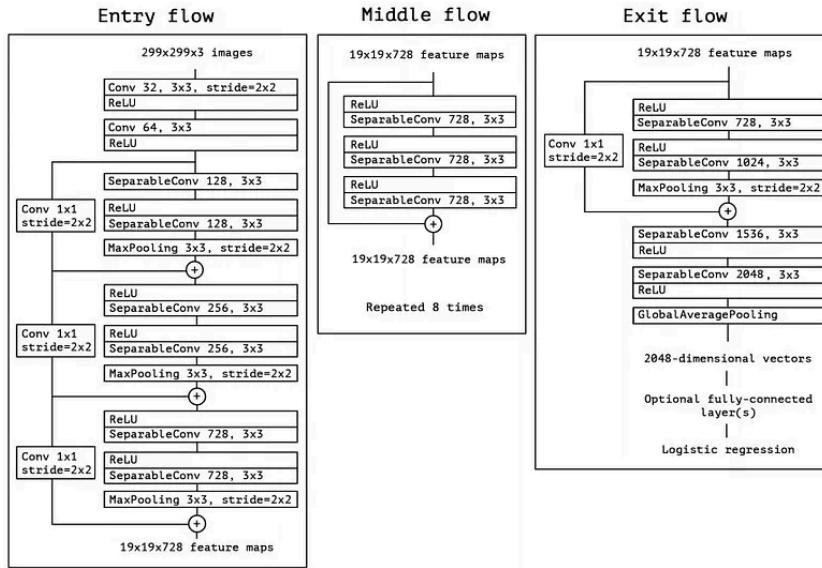
1. VGG-16

VGG-16 adalah model konvolusi yang terdiri dari 16 lapisan yang dapat dilatih, termasuk 13 lapisan konvolusi dan 3 lapisan fully connected. Model ini dikenal dengan arsitektur yang sederhana namun efektif dalam mengenali pola dalam gambar. Kami memilih model ini karena VGG-16 memiliki arsitektur sederhana yang mudah dipahami dan diimplementasikan, cocok untuk dataset dengan jumlah gambar yang terbatas seperti dataset motif batik kami.



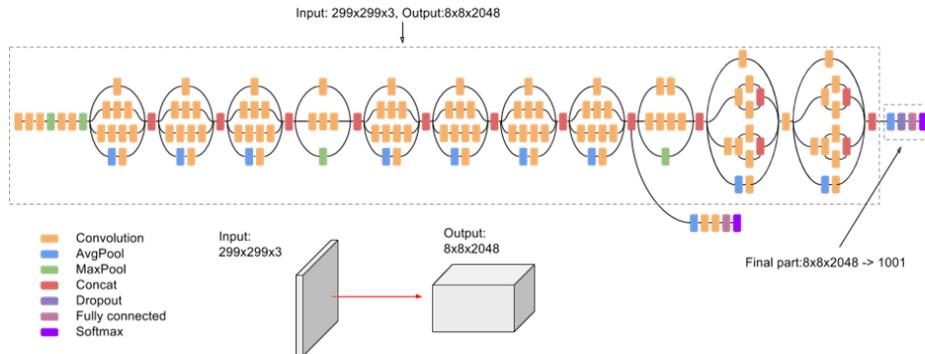
2. Xception

Xception (Extreme Inception) adalah model yang menggunakan depthwise separable convolutions untuk meningkatkan efisiensi dan kinerja. Model ini terdiri dari 71 lapisan dan merupakan perbaikan dari arsitektur Inception. Kami memilih model ini karena Xception mengurangi jumlah parameter dan meningkatkan kecepatan pelatihan tanpa mengorbankan akurasi, cocok untuk dataset yang lebih kecil namun beragam seperti dataset motif batik kami.



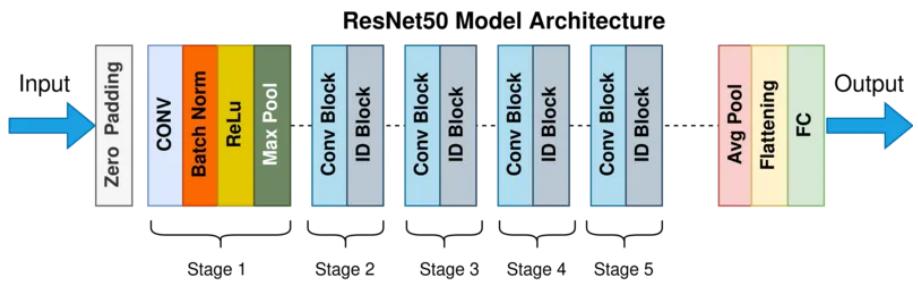
3. InceptionV3

InceptionV3 adalah salah satu versi dari arsitektur Inception yang lebih kompleks dan dalam, terdiri dari modul-modul Inception yang menggabungkan berbagai ukuran filter dalam satu lapisan. Kami memilih model ini karena InceptionV3 mampu menggabungkan informasi dari berbagai skala yang berbeda dalam satu lapisan, yang penting untuk menangkap detail-detail halus dalam motif batik.



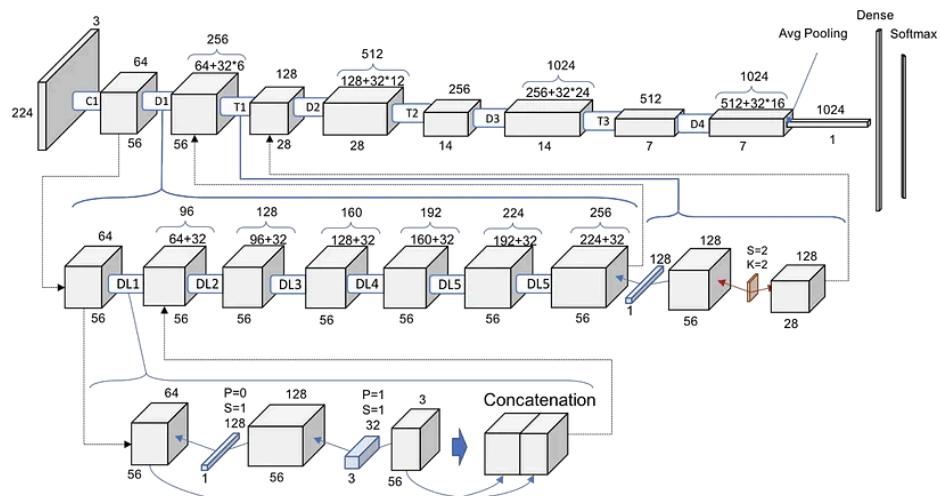
4. ResNet50

ResNet50 (Residual Networks) adalah model dengan 50 lapisan yang menggunakan konsep residual learning untuk mengatasi masalah degradasi di jaringan yang sangat dalam. Model ini memperkenalkan shortcut connections yang memungkinkan informasi mengalir lebih lancar. Kami memilih model ini karena ResNet50 sangat efektif dalam melatih jaringan yang sangat dalam dan mencapai performa yang sangat baik dalam klasifikasi gambar, cocok untuk mengenali pola kompleks dalam motif batik.



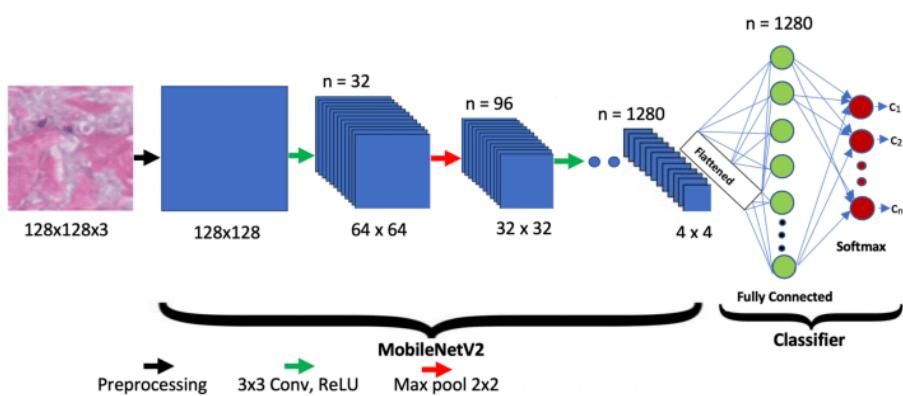
5. DenseNet121

DenseNet121 (Densely Connected Convolutional Networks) adalah model yang menghubungkan setiap lapisan ke semua lapisan sebelumnya dalam jaringan. Setiap lapisan menerima "feature maps" dari semua lapisan sebelumnya. Kami memilih model ini karena DenseNet121 memperbaiki aliran informasi dan gradien di dalam jaringan, yang dapat meningkatkan kinerja model pada dataset yang relatif kecil dan memastikan pemanfaatan fitur yang efisien.



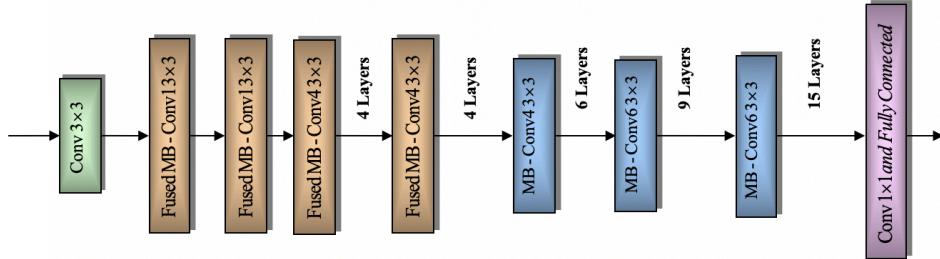
6. MobileNetV2

MobileNetV2 adalah model yang dirancang untuk perangkat mobile dan embedded dengan menggunakan depthwise separable convolutions dan inverted residuals. Model ini sangat efisien dalam hal ukuran dan kecepatan. Kami memilih model ini karena MobileNetV2 ringan dan cepat, sangat cocok untuk aplikasi real-time dan perangkat dengan keterbatasan sumber daya, yang relevan jika model akan diterapkan pada perangkat dengan daya komputasi terbatas.



7. EfficientNetV2

EfficientNetV2 adalah model yang dirancang untuk meningkatkan efisiensi dan akurasi dengan menggunakan teknik compound scaling. Model ini mengoptimalkan dimensi kedalaman, lebar, dan resolusi jaringan. Kami memilih model ini karena EfficientNetV2 menawarkan keseimbangan yang sangat baik antara efisiensi komputasi dan akurasi tinggi, menjadikannya pilihan ideal untuk dataset motif batik yang memerlukan model dengan kinerja optimal dan efisiensi tinggi.



Model Training

Setelah memilih model-model yang akan digunakan, baik scratch maupun transfer learning, langkah berikutnya adalah training atau melatih model tersebut dengan dataset motif batik yang telah displit dan di augmentasi. Proses training melibatkan beberapa tahapan penting untuk memastikan model dapat belajar dan mengenali pola dengan baik.

Model dilatih menggunakan train set, di mana gambar-gambar motif batik digunakan sebagai input. Pada setiap epoch pelatihan, model memperbarui bobot-bobotnya berdasarkan perhitungan kesalahan (error) antara prediksi model dan label sebenarnya. Kami menggunakan algoritma optimasi Adam untuk memperbarui bobot-bobot ini.

Untuk mengevaluasi efektivitas augmentasi data, kami melatih dua model pertama (scratch) tanpa dan dengan augmentasi data, dan membandingkan hasilnya. Berdasarkan performa kedua model tersebut, kami akan menentukan pendekatan terbaik (tanpa atau dengan augmentasi) untuk digunakan pada model-model berikutnya.

Selama training, kami juga memantau kinerja model menggunakan menggunakan data validation set untuk mengevaluasi model di setiap epoch tanpa mempengaruhi pembaruan bobot. Ini membantu dalam mendeteksi overfitting, di mana model mungkin memiliki kinerja yang sangat baik pada data training tetapi tidak pada data yang tidak terlihat sebelumnya (test set). Untuk mengatasi overfitting, kami menggunakan teknik seperti early stopping, di mana pelatihan dihentikan jika kinerja pada set validasi tidak membaik setelah beberapa epoch tertentu.

Hyperparameter tuning juga dilakukan selama proses pelatihan untuk menemukan kombinasi terbaik dari parameter seperti learning rate, batch size, dan jumlah epoch. Ini dilakukan untuk mengoptimalkan kinerja model.

Pada model-model transfer learning, kami melakukan fine-tuning dengan membekukan lapisan awal (freeze layer) dari model yang telah dilatih sebelumnya dan hanya melatih lapisan-lapisan terakhir. Proses ini memungkinkan model untuk menyesuaikan bobot pada lapisan akhir terhadap dataset motif batik kami tanpa mengubah bobot pada lapisan awal yang sudah memiliki pengetahuan umum dari dataset besar seperti ImageNet. Teknik ini membantu dalam mempercepat pelatihan dan meningkatkan performa model dengan memanfaatkan pengetahuan yang sudah ada.

Model Evaluation

Setelah model dilatih, langkah berikutnya adalah mengevaluasi kinerjanya menggunakan test set. Evaluasi ini penting untuk memahami seberapa baik model dapat menggeneralisasi pola dari data training ke data baru yang tidak terlihat sebelumnya.

Kami menggunakan dua metrik utama untuk mengevaluasi model: test loss dan test accuracy. Test loss mengukur kesalahan model dalam memprediksi kelas motif batik pada test set. Nilai loss yang lebih rendah menunjukkan model yang lebih baik dalam memprediksi dengan benar. Sedangkan test accuracy akan mengukur persentase prediksi yang benar dari total prediksi pada test set. Akurasi yang tinggi menunjukkan bahwa model dapat mengenali motif batik dengan benar dalam sebagian besar kasus.

Selain itu, kami juga menggunakan confusion matrix untuk masing-masing model. Confusion matrix adalah alat evaluasi yang memberikan gambaran rinci tentang kinerja model dengan menunjukkan jumlah prediksi yang benar dan salah untuk setiap kelas. Matrix ini terdiri dari empat komponen utama untuk setiap kelas: true positives (TP), true negatives (TN), false positives (FP), dan false negatives (FN).

- True Positive (TP): Prediksi model benar dan sesuai dengan label sebenarnya.
- True Negative (TN): Prediksi model salah dan sesuai dengan label yang bukan target.
- False Positive (FP): Prediksi model salah tetapi dianggap benar oleh model.
- False Negative (FN): Prediksi model benar tetapi dianggap salah oleh model.

Dengan melihat confusion matrix, kami dapat memahami di mana model membuat kesalahan dan apakah ada kelas tertentu yang lebih sulit untuk diklasifikasikan. Diagram di bawah ini menggambarkan struktur dasar dari confusion matrix multiclass:

		Estimate		
		$c_0 \dots c_{k-1}$	c_k	$c_{k+1} \dots c_n$
annotated ground truth	$c_{k+1} \dots c_n$	TN	FP	TN
	c_k	FN	TP	FN
	$c_0 \dots c_{k-1}$	TN	FP	TN

Legend:

- TN: true negative
- TP: true positive
- FN: false negative
- FP: false positive

Model Optimization

Setelah mengevaluasi kinerja awal dari sembilan model yang dilatih, langkah berikutnya adalah melakukan optimisasi lebih lanjut pada model yang memiliki akurasi tertinggi. Kami memutuskan untuk melakukan fine-tuning pada model terbaik ini dengan tujuan untuk meningkatkan akurasinya lebih lanjut.

Fine-tuning adalah proses melanjutkan training model dengan learning rate yang lebih rendah, biasanya pada beberapa lapisan terakhir dari model. Ini memungkinkan model untuk menyesuaikan lebih baik dengan karakteristik spesifik dari dataset motif batik tanpa mengubah secara drastis bobot-bobot yang telah dipelajari. Dalam proses fine-tuning ini, kami akan membuka lapisan awal model dan hanya melatih ulang lapisan-lapisan akhir.

Setelah proses fine-tuning, kami akan mengevaluasi kembali model menggunakan test set untuk memastikan bahwa perbaikan yang dilakukan memberikan peningkatan nyata dalam akurasi dan kemampuan model dalam mengenali motif batik. Dengan demikian, kami dapat memastikan bahwa model yang dihasilkan tidak hanya akurat tetapi juga dapat diandalkan dan efisien dalam aplikasi praktis.

Best Model

Setelah melalui serangkaian tahapan mulai dari data preprocessing, model selection, model training, hingga model optimization, akhirnya kami berhasil mendapatkan model terbaik untuk tugas klasifikasi motif batik ini. Model terbaik ini dipilih berdasarkan kinerja akurasi tertinggi yang dicapai setelah evaluasi dan fine-tuning.

Model terbaik yang kami peroleh menunjukkan performa yang unggul dalam mengenali dan mengklasifikasikan tiga motif batik yang kami gunakan dalam dataset: Parang, Mega Mendung, dan Kawung. Hasil akhir menunjukkan bahwa model ini mampu mencapai akurasi yang tinggi, yang berarti model dapat secara efektif dan akurat mengklasifikasikan gambar motif batik dengan tingkat kesalahan yang sangat rendah.

HASIL DAN ANALISA

Training dan evaluation

Semua model dilatih menggunakan hyperparameter yang sama untuk memastikan konsistensi dalam evaluasi. Kami menggunakan optimizer Adam dengan learning rate 0.001 dan loss function categorical cross entropy. Setiap model dilatih untuk jumlah epoch tertentu dengan menggunakan metrik akurasi untuk mengukur performanya. Untuk dua model pertama (Scratch 1 dan Scratch 2), kami membandingkan hasil training tanpa dan dengan augmentasi data untuk menentukan pendekatan terbaik. Berdasarkan hasil dari kedua model ini, kami melanjutkan pelatihan model-model selanjutnya (transfer learning) dengan pendekatan yang memberikan hasil lebih baik.

Tabel 1: Loss dan accuracy

Model	Augment	Epoch	First epoch	Last epoch	Evaluation
Scratch 1	Without	50 (early stop: 19)	Loss: 1.7627 Acc: 0.3917 Val loss: 1.1177 Val acc: 0.3077	Loss: 0.553 Acc: 0.7083 Val loss: 0.7230 Val acc: 0.6923	Test loss: 0.4736 Test acc: 0.7500
Scratch 1	With	50 (early stop: 19)	Loss: 1.5581 Acc: 0.2667 Val loss: 1.0957 Val acc: 0.3077	Loss: 0.6442 Acc: 0.6570 Val loss: 0.5919 Val acc: 0.6923	Test loss: 0.6237 Test acc: 0.6250
Scratch 2	Without	50 (early stop: 6)	Loss: 1.9101 Acc: 0.4500 Val loss: 0.9651 Val acc: 0.6923	Loss: 0.0788 Acc: 0.9570 Val loss: 1.2059 Val acc: 0.3077	Test loss: 0.9744 Test acc: 0.6875
Scratch 2	With	50 (early stop: 8)	Loss: 2.0583 Acc: 0.4083 Val loss: 1.1090 Val acc: 0.3077	Loss: 0.8019 Acc: 0.6667 Val loss: 2.7389 Val acc: 0.3077	Test loss: 0.9707 Test acc: 0.5000
VGG-16	Without	50	Loss: 0.1646 Acc: 0.9667 Val loss: 0.0911 Val acc: 1.0000	Loss: 0.0233 Acc: 0.9917 Val loss: 0.0054 Val acc: 1.0000	Test loss: 0.8002 Test acc: 0.8125
Xception	Without	50	Loss: 0.1182 Acc: 0.9917 Val loss: 0.1344 Val acc: 1.0000	Loss: 0.0230 Acc: 0.9917 Val loss: 0.0327 Val acc: 1.0000	Test loss: 0.5171 Test acc: 0.8125
Inception V3	Without	50 (early stop: 10)	Loss: 0.0661 Acc: 0.9833 Val loss: 0.2430 Val acc: 0.8462	Loss: 0.0391 Acc: 0.9917 Val loss: 0.1998 Val acc: 0.8462	Test loss: 0.3547 Test acc: 0.8750

ResNet50	Without	50 (early stop: 29)	Loss: 0.7237 Acc: 0.7500 Val loss: 0.9457 Val acc: 0.6154	Loss: 0.5465 Acc: 0.8083 Val loss: 0.8422 Val acc: 0.6154	Test loss: 0.5914 Test acc: 0.7500
DenseNet 121	Without	50	Loss: 0.1872 Acc: 0.9917 Val loss: 0.2139 Val acc: 1.0000	Loss: 0.0370 Acc: 0.9833 Val loss: 0.0558 Val acc: 1.0000	Test loss: 0.2462 Test acc: 0.8125
MobileNet V2	Without	50	Loss: 1.2624 Acc: 0.2750 Val loss: 1.0132 Val acc: 0.6154	Loss: 0.0274 Acc: 0.9917 Val loss: 0.0513 Val acc: 1.0000	Test loss: 0.2980 Test acc: 0.9375
Efficient NetV2	Without	50	Loss: 1.1231 Acc: 0.3833 Val loss: 1.0986 Val acc: 0.3846	Loss: 1.0768 Acc: 0.4333 Val loss: 1.0931 Val acc: 0.3077	Test loss: 1.3175 Test acc: 0.2500

Dari tabel di atas, dua model scratch kami dilatih dua kali untuk membandingkan hasil training tanpa dan dengan augmentasi data. Dari hasil perbandingan tersebut, kami mendapatkan bahwa model yang dilatih tanpa augmentasi data memberikan hasil accuracy yang lebih tinggi (Scratch 1: 0.7500, Scratch 2: 0.6875) dibandingkan dengan model yang dilatih dengan augmentasi data. Oleh karena itu, kami memutuskan untuk melanjutkan training model-model selanjutnya (transfer learning) tanpa augmentasi data dengan harapan akan memberikan hasil accuracy yang lebih tinggi.

Tabel 2: Confusion matrix

Model	Actual class	Predicted class		
		Batik Parang	Batik Mega Mendung	Batik Kawung
Scratch 1 (Without augment)	Batik Parang	1	0	4
	Batik Mega Mendung	0	6	0
	Batik Kawung	0	0	5
Scratch 1 (With augment)	Batik Parang	0	2	3
	Batik Mega Mendung	0	0	0
	Batik Kawung	0	1	4
Scratch 2 (Without augment)	Batik Parang	5	0	0
	Batik Mega Mendung	1	5	0

	Batik Kawung	4	0	1
Scratch 2 (With augment)	Batik Parang	5	0	0
	Batik Mega Mendung	3	3	0
	Batik Kawung	5	0	0
VGG-16	Batik Parang	3	0	2
	Batik Mega Mendung	0	0	0
	Batik Kawung	1	0	4
Xception	Batik Parang	3	0	2
	Batik Mega Mendung	0	6	0
	Batik Kawung	1	0	4
Inception V3	Batik Parang	4	0	1
	Batik Mega Mendung	0	6	0
	Batik Kawung	1	0	4
ResNet50	Batik Parang	1	1	3
	Batik Mega Mendung	0	6	0
	Batik Kawung	0	0	5
DenseNet 121	Batik Parang	3	0	2
	Batik Mega Mendung	0	6	0
	Batik Kawung	1	0	4
Mobile NetV2	Batik Parang	5	0	0
	Batik Mega Mendung	0	6	0
	Batik Kawung	1	0	4
Efficient NetV2	Batik Parang	0	0	5
	Batik Mega Mendung	0	0	6
	Batik Kawung	1	0	4

Setelah melatih dan mengevaluasi semua model, baik yang menggunakan pendekatan scratch maupun transfer learning, kami memperoleh hasil yang bervariasi dalam hal loss, accuracy, dan performa pada confusion matrix. Berdasarkan analisis dari tabel hasil loss dan accuracy

(tabel 1) serta confusion matrix (tabel 2) di atas, kami menemukan bahwa model dengan performa terbaik adalah model transfer learning MobileNetV2.

Model MobileNetV2 menunjukkan akurasi tertinggi dibandingkan dengan model-model lainnya. Performanya yang unggul terlihat dari nilai test accuracy yang lebih tinggi dan distribusi yang lebih baik dalam confusion matrix.

Berikut adalah analisa lebih rinci dan detail dari hasil model MobileNetV2:

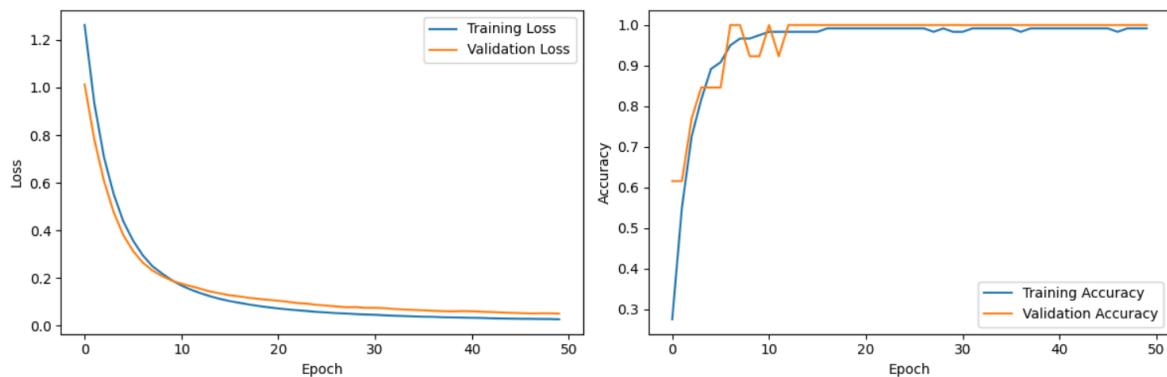
Model MobileNetV2 dilatih dengan train data tanpa augmentasi menggunakan optimizer Adam dengan learning rate 0.001 dan loss function categorical crossentropy selama 50 epoch.

Hasil training:

- Epoch 1/50:
Loss: 1.2624, Accuracy: 0.2750, Val Loss: 1.0132, Val Accuracy: 0.6154
- Epoch 50/50:
Loss: 0.0274, Accuracy: 0.9917, Val Loss: 0.0513, Val Accuracy: 1.0000

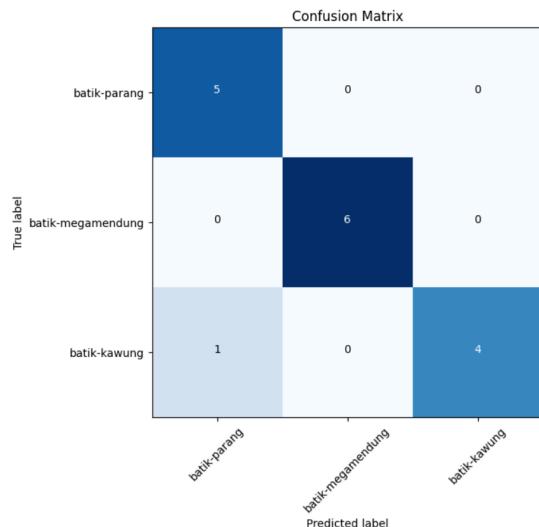
Evaluasi pada test set:

- Test Loss: 0.2980
- Test Accuracy: 0.9375



Grafik di atas menunjukkan bahwa model MobileNetV2 menunjukkan peningkatan yang signifikan dalam akurasi dari awal hingga akhir pelatihan. Training loss dan validation loss terus menurun secara konsisten, menunjukkan bahwa model belajar dengan baik dari data tanpa overfitting. Akurasi pelatihan dan validasi juga meningkat dengan cepat dalam beberapa epoch pertama dan mencapai nilai mendekati 1, menunjukkan performa yang sangat baik.

Confusion matrix:



- True Positive (TP): Batik Mega Mendung diklasifikasikan dengan benar 6 kali.
- False Positive (FP): Batik Parang diklasifikasikan sebagai Batik Kawung 1 kali.
- False Negative (FN): Batik Kawung diklasifikasikan sebagai Batik Parang 1 kali.
- True Negative (TN): Semua prediksi lainnya benar, menunjukkan bahwa model sangat akurat dalam membedakan antara kelas-kelas tersebut.

Hasil confusion matrix di atas juga menunjukkan bahwa model MobileNetV2 memiliki kinerja yang sangat baik dalam mengklasifikasikan setiap kelas motif batik. Model ini berhasil mengenali dan membedakan motif-motif dengan akurasi yang tinggi, mengurangi kesalahan prediksi yang signifikan pada kelas-kelas yang lebih sulit.

Optimization

Setelah melakukan analisis menyeluruh terhadap berbagai model yang dilatih, kami menemukan bahwa model MobileNetV2 menunjukkan kinerja terbaik dalam klasifikasi motif batik. Untuk lebih meningkatkan performa model ini, kami melakukan proses fine-tuning yang ekstensif.

Proses fine-tuning dimulai dengan memuat bobot dari model MobileNetV2 yang telah dilatih sebelumnya selama 50 epoch. Kami membekukan semua lapisan dari base model dan hanya melatih 23 lapisan terakhir (15% dari total 155 layer). Pendekatan ini memungkinkan model untuk menyesuaikan diri dengan dataset motif batik tanpa kehilangan fitur-fitur umum yang telah dipelajari sebelumnya. Kami menggunakan Adam optimizer dengan learning rate yang lebih rendah ($1e-4$) untuk proses fine-tuning ini, yang merupakan praktik umum untuk mencegah perubahan yang terlalu drastis pada bobot model.

Selama proses fine-tuning, kami menerapkan beberapa teknik optimasi pelatihan. Callback ModelCheckpoint digunakan untuk menyimpan bobot model terbaik berdasarkan akurasi validasi. EarlyStopping diterapkan untuk menghentikan pelatihan jika tidak ada peningkatan pada validation loss selama 5 epoch berturut-turut. Kami juga menggunakan

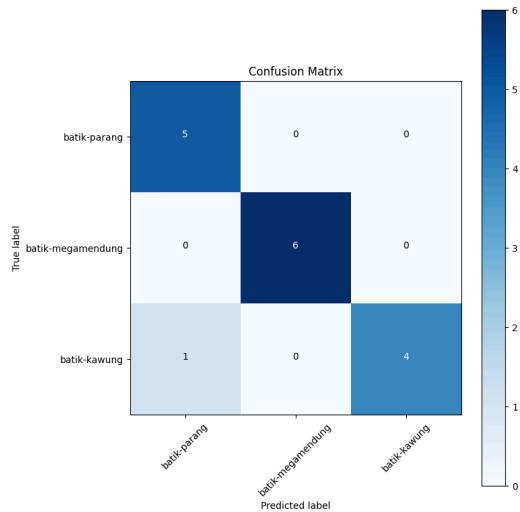
ReduceLROnPlateau untuk mengurangi learning rate jika validation loss tidak membaik, membantu model menemukan minimum lokal yang lebih baik.

Fine-tuning berlanjut dari epoch ke-51 hingga epoch ke-66, di mana proses berhenti karena ReduceLROnPlateau mengurangi learning rate menjadi sekitar 2e-5. Ini menandakan bahwa model telah mencapai titik di mana perbaikan lebih lanjut memerlukan penyesuaian yang lebih halus.

Hasil fine-tuning menunjukkan peningkatan yang signifikan dalam performa model. Akurasi training mencapai hampir 100% pada beberapa epoch, sementara akurasi validasi konsisten mencapai 100%. Loss pelatihan dan validasi juga menurun secara konsisten, menunjukkan bahwa model belajar dengan baik tanpa overfitting yang signifikan.

Evaluasi akhir pada test set menunjukkan hasil yang sangat memuaskan. Model fine-tuned mencapai akurasi 93.75% pada test set, dengan loss sebesar 0.3744. Ini merupakan peningkatan dari performa awal model MobileNetV2 sebelum fine-tuning.

Confusion matrix hasil fine-tuning juga menunjukkan kinerja model yang sangat baik:



- Batik Parang: 5 dari 5 sampel diprediksi dengan benar
- Batik Mega Mendung: 6 dari 6 sampel diprediksi dengan benar
- Batik Kawung: 4 dari 5 sampel diprediksi dengan benar, dengan 1 sampel salah diklasifikasikan sebagai Batik Parang

Model menunjukkan akurasi sempurna untuk Batik Parang dan Batik Mega Mendung, dengan sedikit kesalahan pada Batik Kawung.

Berikut gambar sample Batik Parang yang salah diklasifikasikan sebagai Batik Kawung
actual: batik-parang, pred: batik-kawung
prob: 1.00



Berdasarkan gambar tersebut, kami menganalisis beberapa faktor yang mungkin menyebabkan kesalahan klasifikasi:

1. Kompleksitas pola:

Batik Parang biasanya memiliki pola yang lebih linier dan berulang, sementara Batik Kawung memiliki pola yang lebih geometris dan berbentuk oval. Jika pola Batik Parang dalam gambar ini memiliki elemen yang sangat kompleks dan menyerupai pola geometris, ini bisa menyebabkan algoritma kesulitan membedakannya.

2. Elemen garis:

Batik Parang cenderung memiliki garis-garis panjang yang berulang secara diagonal, sementara Batik Kawung lebih fokus pada bentuk oval yang disusun berulang kali. Jika garis dalam Batik Parang ini tidak terlihat jelas atau tumpang tindih dengan elemen lainnya, algoritma bisa saja salah menginterpretasikannya.

3. Warna dan kontras:

Algoritma *Image Classification* sering kali dipengaruhi oleh warna dan kontras. Jika Batik Parang dalam gambar ini memiliki warna yang tidak umum atau kontras yang rendah, ini bisa mengaburkan pola linier khas Parang dan membuatnya terlihat lebih mirip dengan pola Kawung yang memiliki kontras dan warna berbeda.

4. Variasi bentuk:

Batik Parang yang memiliki variasi bentuk yang lebih tidak konvensional bisa menyebabkan kesalahan klasifikasi. Bentuk yang lebih variatif dan tidak mengikuti pola linier tradisional bisa menyebabkan algoritma salah mengklasifikasikan.

5. Elemen tambahan:

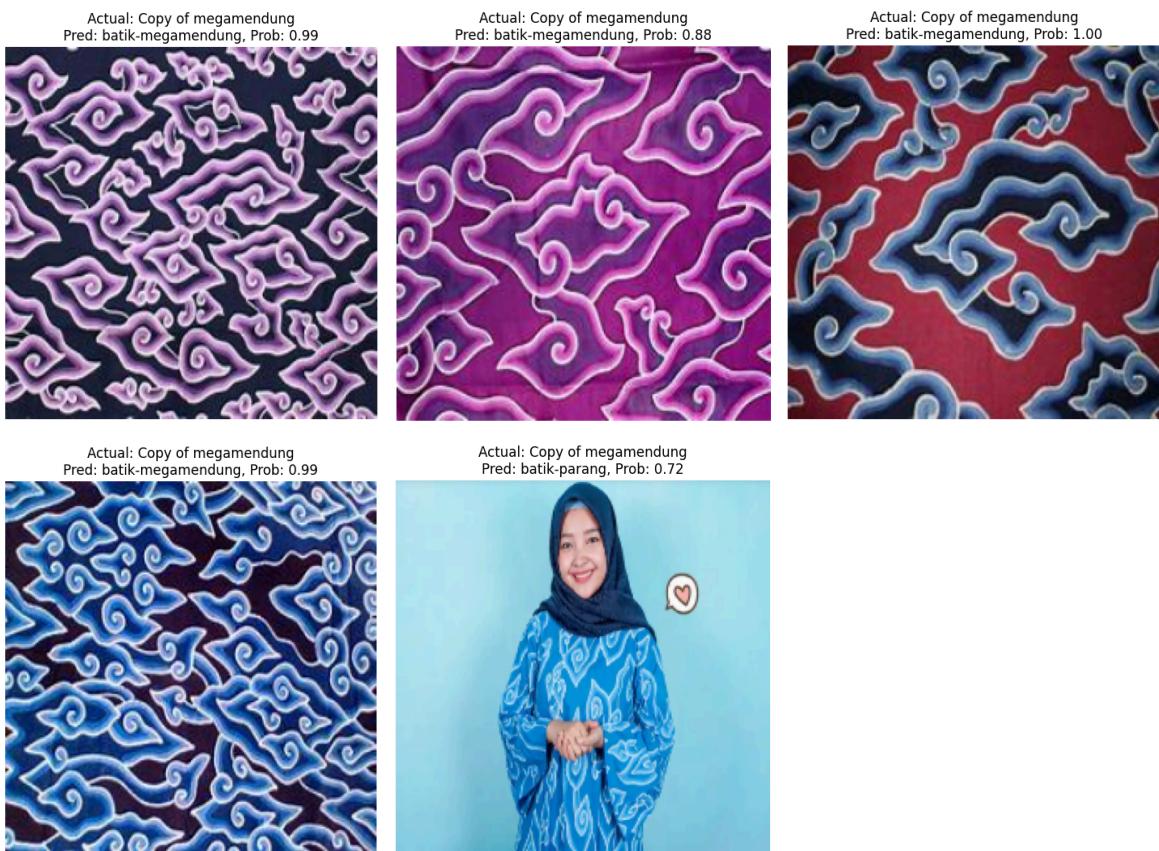
Elemen tambahan seperti hiasan atau motif yang bukan bagian dari pola utama bisa mengganggu algoritma pengenalan. Jika Batik Parang ini memiliki elemen tambahan yang menyerupai elemen pada Batik Kawung, seperti bentuk oval atau elemen geometris lainnya, algoritma bisa terpengaruh oleh elemen tersebut.

Kesalahan klasifikasi ini, meskipun dengan probabilitas yang tinggi (1.00), menunjukkan bahwa model mungkin terlalu yakin dalam prediksinya. Hal ini mengindikasikan bahwa model cenderung overfitting terhadap fitur-fitur tertentu yang mirip dengan Batik Kawung.

Inference

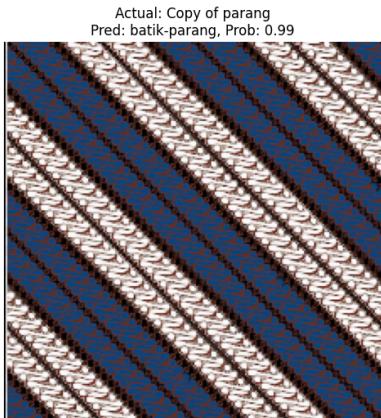
Setelah melakukan fine-tuning dan evaluasi model MobileNetV2, kami melakukan inference terhadap tiga kelas batik dengan masing-masing 5 gambar: 4 gambar motif batik penuh dan 1 gambar orang mengenakan baju batik. Hasil inference ini memberikan wawasan baru tentang kinerja model dalam situasi nyata.

Batik Mega Mendung:

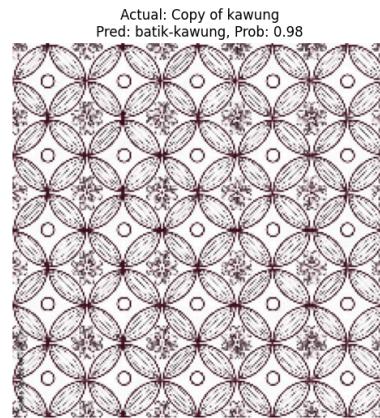
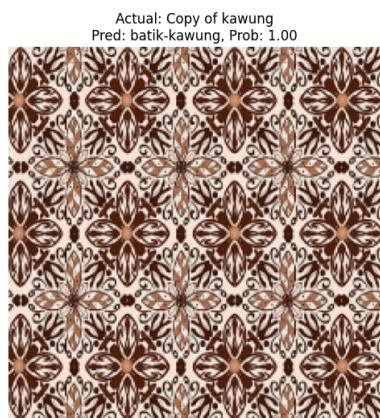
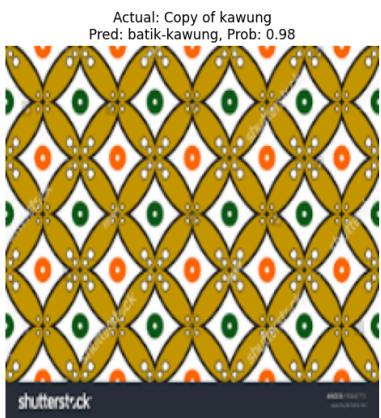


Batik Parang:





Batik Kawung:



Model memiliki akurasi sebesar 93.75%, yang menunjukkan kemampuannya untuk memprediksi dengan benar sebagian besar gambar dalam dataset evaluasi. Meskipun gambar diambil dari Google, model tetap mampu mengenali pola gambar dengan baik. Model berhasil memprediksi dengan akurat semua gambar motif Mega Mendung dan Kawung. Namun, model mengalami kesulitan saat memprediksi gambar orang yang mengenakan baju batik Mega Mendung dan Kawung.

Di sisi lain, hasil menunjukkan bahwa model terkadang masih salah mengklasifikasi motif Parang. Namun, model justru berhasil mengenali pola pada baju batik Parang dengan baik. Hal ini mengindikasikan bahwa model lebih mampu menangkap motif Parang ketika motif

tersebut diaplikasikan pada pakaian yang dikenakan, meskipun ada beberapa kesalahan dalam klasifikasi motif Parang pada gambar-gambar lainnya.

Perbedaan performa ini dapat disebabkan oleh beberapa faktor:

1. Kemiripan Data:

- **Data Latihan:** Jika dataset latihan model lebih banyak mengandung gambar motif batik yang jelas dan tanpa gangguan, model akan lebih mahir dalam mengenali pola tersebut. Sebaliknya, jika ada gambar yang lebih kompleks atau memiliki variasi tinggi, model mungkin mengalami kesulitan dalam memprediksi dengan benar.
- **Data Evaluasi:** Gambar dari Google mungkin memiliki kualitas dan karakteristik yang berbeda dari dataset latihan, seperti variasi dalam resolusi, pencahayaan, dan komposisi, yang dapat mempengaruhi akurasi prediksi.

2. Variasi Terbatas:

- **Pola Batik:** Gambar motif Mega Mendung dan Kawung mungkin memiliki pola yang lebih konsisten dan mudah dikenali dibandingkan motif Parang, yang bisa lebih bervariasi dalam desain dan warna.
- **Kondisi Pakaian:** Ketika motif batik diaplikasikan pada pakaian yang dikenakan, variasi dalam bentuk, lipatan, dan posisi kain dapat mengubah pola yang terlihat dan membuatnya lebih sulit dikenali oleh model.

3. Kondisi Ideal vs Realistik:

- **Kondisi Ideal:** Gambar yang diambil dalam kondisi ideal dengan pencahayaan yang baik dan latar belakang yang bersih cenderung lebih mudah dikenali oleh model.
- **Kondisi Realistik:** Gambar orang yang mengenakan baju batik mungkin diambil dalam berbagai kondisi pencahayaan, sudut, dan situasi, yang dapat mempengaruhi kemampuan model dalam mengenali pola.

4. Konteks Aplikasi:

- **Gambar Statis vs Dinamis:** Gambar motif batik pada kain datar lebih mudah dikenali dibandingkan motif pada pakaian yang dikenakan, yang mungkin memiliki gerakan dan distorsi.

Performa yang lebih baik pada Batik Mega Mendung dapat dijelaskan oleh karakteristik khasnya yang konsisten, seperti pola awan yang bergelombang dan kontras warna yang jelas. Motif ini cenderung lebih mudah dikenali bahkan dalam berbagai variasi dan konteks. Disatu sisi, Performa yang lebih baik pada Batik Kawung dapat dijelaskan oleh karakteristik khasnya yang konsisten, seperti pola simetris berbentuk lingkaran atau elips yang tersusun rapi dan sering kali diatur dalam pola berulang. Motif ini cenderung lebih mudah dikenali bahkan dalam berbagai variasi dan konteks karena keseragaman dan keteraturan pola yang membentuknya, serta kontras warna yang membantu memperjelas bentuk-bentuk geometris tersebut.

Sebaliknya, kesalahan dalam prediksi Batik Parang mungkin disebabkan oleh karakteristik motifnya yang lebih bervariasi dan kompleks. Motif Parang sering kali memiliki pola

diagonal yang terdiri dari elemen-elemen yang lebih kecil dan detail, yang bisa berbeda dalam ukuran, bentuk, dan warna. Variasi ini membuat motif Parang lebih sulit dikenali oleh model, terutama jika motif tersebut tidak konsisten atau terdistorsi dalam gambar. Selain itu, Batik Parang sering kali memiliki elemen-elemen tambahan atau variasi desain yang lebih tinggi, yang dapat menyebabkan kebingungan pada model saat mencoba mengklasifikasikan motif tersebut.

Hasil ini menekankan pentingnya memiliki dataset yang beragam dan representatif, serta kebutuhan untuk terus mengevaluasi model dalam kondisi yang menyerupai aplikasi dunia nyata. Meskipun model menunjukkan potensi yang besar dalam klasifikasi motif batik, masih ada ruang untuk perbaikan, terutama dalam meningkatkan kemampuan generalisasi model pada variasi motif yang lebih luas dan dalam konteks yang berbeda-beda.

Hasil ini juga menunjukkan bahwa evaluasi model tidak boleh hanya bergantung pada metrics dari test set, tetapi juga harus mempertimbangkan kinerja pada data dari sumber eksternal yang mungkin lebih mencerminkan variasi dan kompleksitas dunia nyata. Dengan penyempurnaan lebih lanjut, model ini berpotensi menjadi alat yang sangat berharga dalam pelestarian dan promosi warisan budaya Indonesia, khususnya dalam konteks identifikasi dan klasifikasi motif batik.

KESIMPULAN

Kami telah berhasil melakukan klasifikasi multi class pada motif batik Indonesia menggunakan teknik deep learning. Fokus kami adalah pada tiga motif batik yang terkenal: Parang, Mega Mendung, dan Kawung. Kami menggunakan dataset dari Kaggle yang terdiri dari 142 gambar motif batik. Setelah melakukan preprocessing data, kami membandingkan performa berbagai model deep learning, baik yang dibangun dari awal (scratch) maupun yang menggunakan transfer learning. Hasil eksperimen kami menunjukkan bahwa model transfer learning, khususnya MobileNetV2, memberikan performa terbaik dengan akurasi 93.75% pada test set.

Kami melakukan fine-tuning pada model MobileNetV2 untuk lebih meningkatkan kinerjanya. Proses ini melibatkan pembekuan lapisan-lapisan awal dan pelatihan ulang lapisan-lapisan akhir dengan learning rate yang lebih rendah. Hasil fine-tuning menunjukkan peningkatan performa yang signifikan. Dalam tahap inferensi, kami menguji model dengan gambar-gambar baru dari internet. Hasilnya menunjukkan bahwa model kami memiliki kemampuan yang baik dalam mengenali motif Mega Mendung dan Kawung, namun terkadang masih mengalami kesulitan dengan motif Parang. Kami juga menemukan bahwa model mampu mengenali motif batik pada pakaian yang dikenakan, meskipun dengan tingkat akurasi yang bervariasi.

Percobaan yang kami lakukan mendemonstrasikan potensi penggunaan deep learning dalam klasifikasi motif batik Indonesia. Meskipun demikian, kami menyadari bahwa masih ada ruang untuk perbaikan, terutama dalam meningkatkan kemampuan model untuk mengenali variasi motif yang lebih luas dan dalam konteks yang lebih beragam. Kami berharap percobaan ini dapat berkontribusi pada upaya pelestarian dan promosi warisan budaya Indonesia, khususnya batik. Ke depannya, kami menyarankan untuk memperluas dataset dengan lebih banyak variasi motif dan kondisi pengambilan gambar untuk meningkatkan kehandalan dan aplikabilitas model dalam situasi dunia nyata.

REFERENSI

- Setyaningrum, P. (2022, December 6). *Mengenal Motif Batik Parang, Batik Larangan yang Tidak Boleh Sembbarang Digunakan.* KOMPAS. <https://yogyakarta.kompas.com/read/2022/12/06/135804178/mengenal-motif-batik-parang-batik-larangan-yang-tidak-boleh-sembbarang?page=all>
- Qothrunnada, K. (2023, September 21). *Mengenal batik Mega Mendung: Sejarah, Filosofi, dan Asal Daerahnya.* detikjabar. <https://www.detik.com/jabar/budaya/d-6943931/mengenal-batik-mega-mendung-sejarah-filosofi-dan-asal-daerahnya>
- Setyaningrum, P. (2022, January 10). *Mengenal motif batik Kawung, Sejarah, filosofi, Dan Jenisnya.* KOMPAS.com. <https://regional.kompas.com/read/2022/01/10/172259678/mengenal-motif-batik-kawung-sejarah-filosofi-dan-jenisnya>
- Ajmera, G. (2024, February 16). *Feature Extraction of Images using GCLM (Gray Level Cooccurrence Matrix).* Medium. <https://medium.com/@girishajmera/feature-extraction-of-images-using-glcm-gray-level-cooccurrence-matrix-e4bda8729498>
- GeeksforGeeks. (2024, March 21). *VGG-16: CNN model.* <https://www.geeksforgeeks.org/vgg-16-cnn-model/>
- Rohini, G. (2021, September 23). *Everything you need to know about VGG16.* Medium. <https://medium.com/@mygreatlearning/everything-you-need-to-know-about-vgg16-7315defb5918>
- Hesaraki, S. (2023, December 29). *Xception.* Medium. <https://medium.com/@saba99/xception-cd1adc84290f>
- Tsang, S. H. (2018, September 25). *Review: Xception — With Depthwise Separable Convolution, Better Than Inception-v3 (Image Classification).* Medium. <https://towardsdatascience.com/review-xception-with-depthwise-separable-convolution-better-than-inception-v3-image-dc967dd42568>
- Szegedy, C. (2015, December 11). *Rethinking the Inception Architecture for Computer Vision.* <https://doi.org/10.48550/arXiv.1512.00567>
- Kundu, N. (2023, Jan 23). *Exploring ResNet50: An In-Depth Look at the Model Architecture and Code Implementation.* Medium. https://medium-com.translate.goog/@nitishkundu1993/exploring-resnet50-an-in-depth-look-at-the-model-architecture-and-code-implementation-d8d8fa67e46f?_x_tr_sl=en&_x_tr_tl=id&_x_tr_hl=id&_x_tr_pto=tc&_x_tr_hist=true

Das, S. (2023, Mar 19). *Implementing DenseNet121 in PyTorch: A Step-by-Step Guide*. Medium.

<https://medium.com/deepkapha-notes/implementing-densenet-121-in-pytorch-a-step-by-step-guide-c0c2625c2a60>

Akay, M., Du, Y., Sershen, C. L., Wu, M., et al. (2021, Mar). Deep Learning Classification of Systemic Scelerosis Skin Using the MobileNetV2 Model. *IEEE Open Journal of Engineering in Medicine and Biology*, PP(99), 1-1. <https://doi.org/10.1109/OJEMB.2021.3066097>.

AlTakrouri, S., Noor, N. M., Ahmad, N., Justinia, T., et al. (2023, January). Image super-resolution using generative adversarial networks with EfficientNetV2. *International Journal of Advanced Computer Science and Applications*, 14(2). <https://doi.org/10.14569/IJACSA.2023.01402100>.

Krüger, F. (2016, December). *Activity, context, and plan recognition with computational causal behaviour models* (PhD thesis). https://www.researchgate.net/figure/Confusion-matrix-for-multi-class-classification-The-confusion-matrix-of-a_fig7_314116591