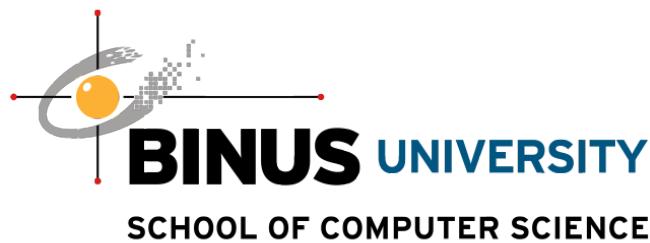
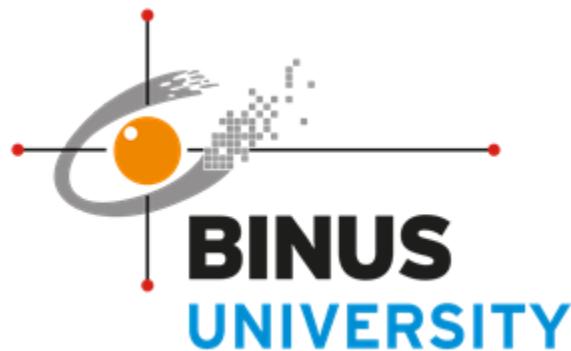


Klasifikasi Multiclass Batik Indonesia Menggunakan Model Deep Learning

CNN dan Transfer Learning dengan MobileNetV3Large



Oleh:

Brandon Agusta Wijaya 2602054146

Nicholas Sugijono 2602064714

Wilson Gregory Pribadi 2602071000

Mata Kuliah : Deep Learning

Tugas : Assignment

Class : LA09

Dosen : Lili Ayu Wulandhari, S.Si., M.Sc., Ph.D.

Tahun Ajaran : Semester Genap 2023/2024

I. Pendahuluan

A. Masalah

Batik memiliki beragam motif yang rumit, yang sering kali sulit diidentifikasi secara manual. Oleh karena itu, kami akan mengembangkan model Deep Learning untuk mengidentifikasi jenis batik menggunakan Image Classification. Dalam proyek ini, kami akan membangun dan melatih model deep learning untuk mengklasifikasikan gambar Batik Kawung, Batik Megamendung, dan Batik Parang.

Tujuan diadakannya pengklasifikasian ini untuk mengidentifikasi dan mengelompokkan batik berdasarkan pola yang terdeteksi di dalamnya dengan menerapkan teknik penganalisisan data, persiapan data, pembangunan model, eksperimen, evaluasi dan juga dokumentasi.

B. Penjelasan Dataset

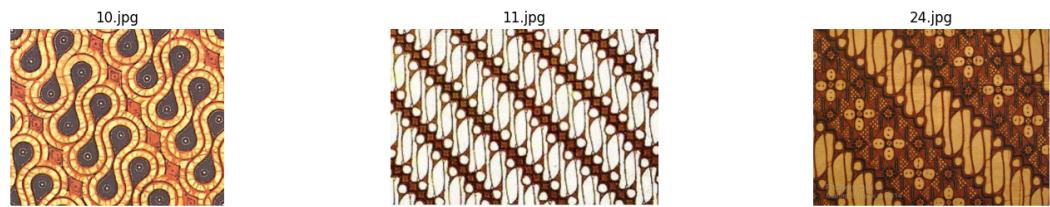
Dataset yang kami gunakan berasal dari Kaggle, dapat diakses di <https://www.kaggle.com/datasets/dionisiusdh/indonesian-batik-motifs>. Dataset ini mencakup 20 jenis batik Indonesia yang berbeda. Kami memilih tiga jenis batik untuk proyek ini: Batik Kawung, Batik Megamendung, dan Batik Parang. Ketiga kategori batik ini dipilih karena, setelah melihat dataset secara manual, kami menemukan bahwa pola pada ketiga jenis batik ini sangat jelas dan konsisten. Pola yang 'bersih' ini akan mempermudah proses pembelajaran model deep learning, meningkatkan akurasi dan efisiensi klasifikasi jenis batik.

Distribusi dari setiap kategori batik yang kami pilih dapat dilihat di dalam tabel dibawah ini :

Batik	Gambar
Batik Kawung	45
Batik Megamendung	46
Batik Parang	50

Kami juga telah melakukan visualisasi untuk melihat 3 gambar sampel batik dari masing-masing kategori batik dan dapat dilihat sebagai berikut :

BATIK-PARANG



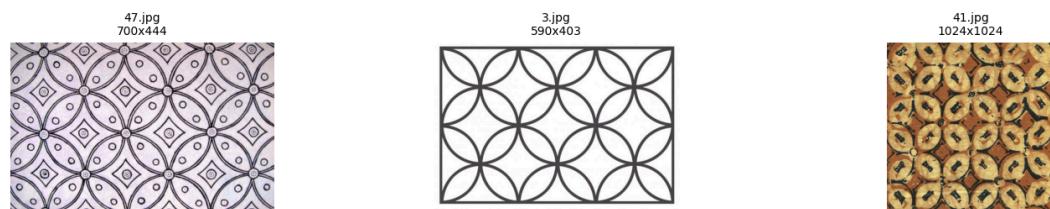
Gambar 1.1 Batik Parang

BATIK-MEGAMENDUNG



Gambar 1.2 Batik Megamendung

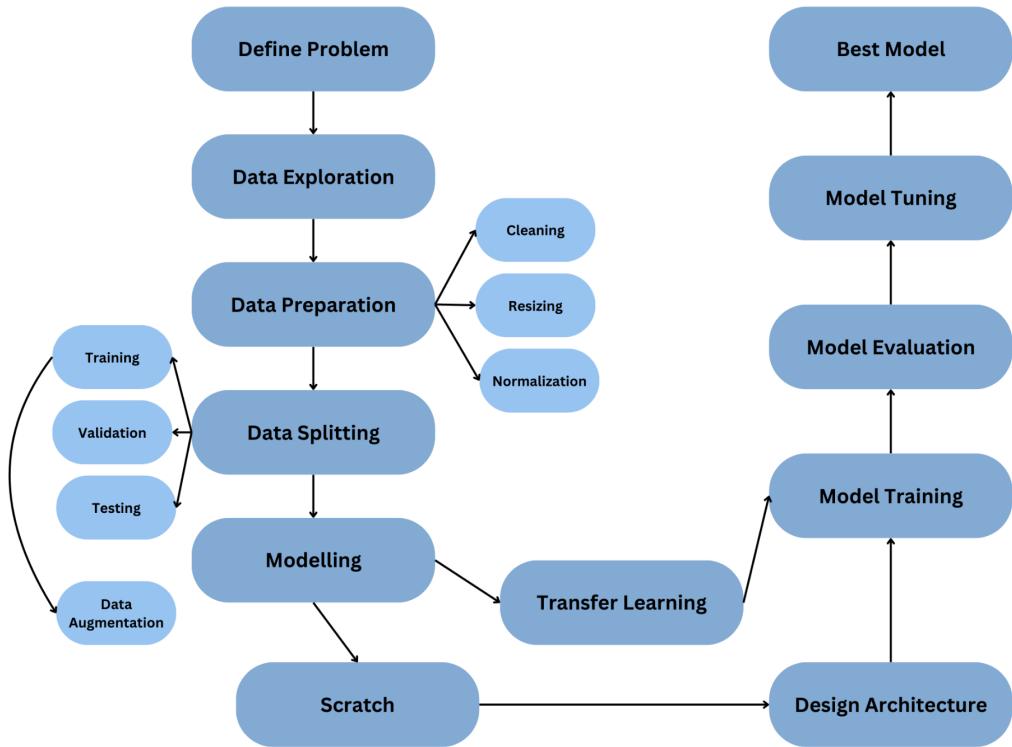
BATIK-KAWUNG



Gambar 1.3 Batik Kawung

Kita dapat melihat bahwa beberapa sampel dari batik ini memiliki perbedaan yang jelas, terlihat dari pola berulang dan simetris yang unik pada setiap jenis batik. Dengan mengamati sampel-sampel ini, kita dapat memahami karakteristik khusus dari setiap jenis batik dalam dataset yang digunakan. Pemahaman ini akan sangat membantu dalam proses pengembangan model deep learning berikutnya, memungkinkan model untuk lebih akurat dalam mengidentifikasi dan mengklasifikasikan jenis-jenis batik.

II. Metodologi



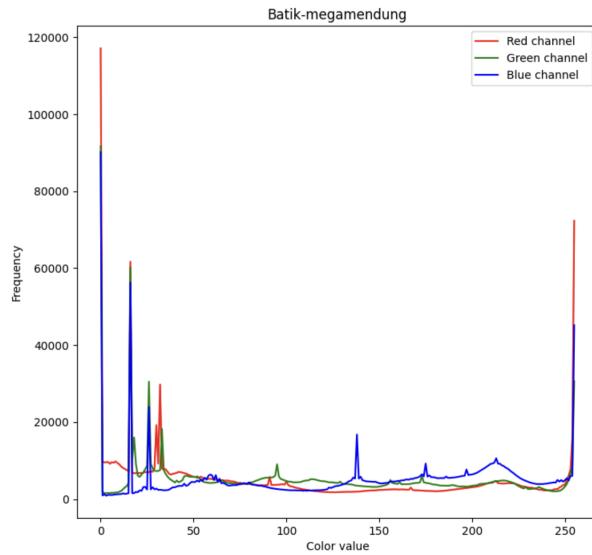
Gambar 2.1 Diagram Metodologi

A. Eksplorasi Data

Kami melakukan eksplorasi data untuk mendapatkan pemahaman yang lebih mendalam mengenai dataset yang akan kami gunakan. Langkah ini penting untuk mengenali karakteristik unik setiap jenis batik dan memastikan kualitas data sebelum memulai proses pengembangan model deep learning.

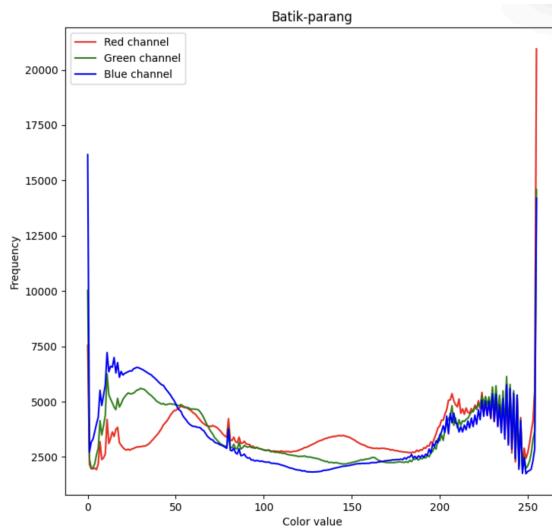
1. Distribusi warna

Kami melihat distribusi warna dengan cara menampilkan histogram warna. Sumbu-x itu menampilkan color value dan sumbu-y menampilkan frekuensi



Gambar 2.2 Distribusi warna batik megamendung

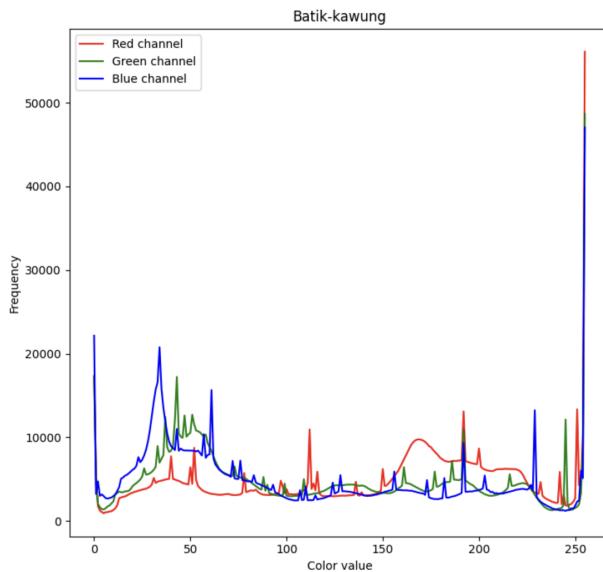
Gambar "Batik-megamendung" memiliki banyak piksel gelap (nilai warna rendah) dan piksel terang (nilai warna tinggi), dengan frekuensi tertinggi pada warna biru. Distribusi ini bisa menunjukkan karakteristik pola dan warna yang digunakan dalam motif batik tersebut, dengan dominasi warna biru yang signifikan.



Gambar 2.3 Distribusi warna batik parang

Gambar "Batik-parang" menunjukkan distribusi warna yang lebih seimbang dibandingkan dengan "Batik-megamendung," dengan puncak

signifikan pada nilai warna rendah (0-50) dan tinggi (sekitar 250) untuk semua saluran warna, menandakan banyak piksel gelap dan terang. Warna biru dan hijau dominan pada nilai warna rendah dan menengah, sementara warna merah memiliki puncak tinggi pada nilai warna tinggi, mencerminkan banyaknya piksel merah terang. Secara keseluruhan, "Batik-parang" memiliki variasi warna yang lebih beragam dengan distribusi warna yang merata, mencerminkan motif batik dengan pola warna yang kaya dan penggunaan warna terang yang signifikan.



Gambar 2.4 Distribusi warna batik kawung

Gambar "Batik-kawung" memperlihatkan distribusi warna dengan puncak signifikan pada nilai warna rendah (0-50) dan tinggi (sekitar 250) untuk semua saluran warna, menandakan keberadaan banyak piksel gelap dan terang. Saluran biru memiliki frekuensi tertinggi pada nilai rendah dan tinggi, sementara saluran hijau dan merah menunjukkan distribusi yang lebih merata di seluruh rentang warna. Puncak-puncak pada nilai warna menengah menunjukkan variasi warna yang kompleks, mencerminkan motif batik dengan pola warna yang kaya dan penggunaan warna terang yang signifikan. Secara keseluruhan, "Batik-kawung" memiliki variasi

warna yang seimbang dengan dominasi warna biru pada piksel gelap dan terang.

2. Perbedaan resolusi



Gambar 2.5 Perbedaan resolusi

Gambar di atas menunjukkan variasi resolusi yang signifikan di antara beberapa pola tekstil. Resolusi gambar berkisar dari yang terkecil 280x400 piksel (contoh: 5.jpg) hingga yang terbesar 1776x1776 piksel (contoh: 1.jpg). Gambar dengan resolusi tinggi seperti 1776x1776 dan 1600x1200 piksel memberikan detail yang lebih tajam dan jelas dibandingkan dengan gambar beresolusi lebih rendah seperti 280x400 dan 800x500 piksel. Variasi resolusi ini mencerminkan perbedaan kualitas dan tingkat detail yang dapat diperoleh dari masing-masing gambar.

Metode penyelesaian masalah pengklasifikasian tiga jenis batik ini menggunakan Convolutional Neural Network (CNN). Tahapan pertama ini merupakan pengklasifikasian ketiga jenis batik ini lalu melakukan preprocessing seperti resizing, normalisasi dan juga augmentasi untuk meningkatkan variasi pada data dan mengurangi overfitting. Data lalu dibagi menjadi data training, validasi dan juga testing. Model CNN ini juga menggunakan lapisan konvolusi, pooling, dan fully connected untuk

melakukan pengekstrakan fitur yang ada. Model juga dilatih dengan menggunakan data training dengan teknik optimisasi Adam dan hyperparameter yang disesuaikan melalui eksperimen. Evaluasi dilakukan dengan menggunakan data validasi dari akurasi, precision, recall, dan F1-Score untuk memahami datanya. Lalu terjadi optimisasi dan fine-tuning untuk meningkatkan akurasi dan generalisasi model.

B. Preparasi Data

1. Pembersihan Data

Pada proses preparasi data sebelum melakukan training data, langkah pertama yang harus dilakukan merupakan pembersihan data. Pembersihan ini meliputi pengecekan terhadap gambar yang ada pada dataset dan menghapus gambar yang tidak dapat dibaca dan memastikan kualitas datanya konsisten. Setelah melakukan pembersihan data, proses selanjutnya adalah mengubah seluruh ukuran gambar menjadi 224x224 piksel untuk memastikan seluruh gambar memiliki dimensi yang sama dan juga mengkonversikan gambar menjadi tiga warna utama yaitu merah, biru dan hijau sehingga gambar tersebut dapat diinterpretasikan dengan benar oleh model data visual berwarna.

2. Normalisasi

Setelah melakukan proses tersebut, normalisasi akan dilakukan dengan membagi nilai tensor dari setiap piksel dalam gambar dengan 255. Proses normalisasi ini penting untuk skala data piksel ke rentang 0 dan 1 untuk membantu model pada proses pembelajaran dengan mencegah nilai besar mengganggu gradien pada proses backpropagation. Selain itu, proses ini dapat membantu mempercepat konvergensi saat melakukan training model dan juga meningkatkan stabilitas numerik.

3. Splitting Data

Proses selanjutnya adalah pembagian data menjadi tiga subset yaitu, training, testing dan validation. Pembagian ini terjadi untuk memastikan model dapat dievaluasi dengan baik. Data training ini digunakan untuk melatih model, model validation untuk melakukan tuning

pada parameter model agar menghindari adanya overfitting pada data, sedangkan data testing akan digunakan untuk melakukan pengevaluasian terhadap performa akhir pada model.

4. Data Augmentasi

Proses yang terakhir terjadi merupakan proses augmentasi data yang diterapkan pada training data. Proses ini melibatkan rotasi, pemotongan, pencerminan, dan merubah skala dengan tujuan untuk meningkatkan jumlah serta variasi data training tanpa perlu mengumpulkan data baru lagi.

C. Model

1. Model from Scratch

Kami menggunakan arsitektur model neural network yang dikembangkan menggunakan TensorFlow dan Keras. Model ini dirancang untuk melakukan klasifikasi gambar pada dataset yang memiliki dimensi gambar 224x224 piksel dan 3 saluran warna (RGB). Arsitektur model terdiri dari lapisan-lapisan berikut:

- a. **Input Layer:** Lapisan ini adalah lapisan input yang menerima data gambar dengan dimensi yang telah ditentukan.
- b. **Convolutional Layers:** Terdapat dua lapisan konvolusi dengan masing-masing 32 dan 64 filter berukuran 3x3 piksel. Lapisan ini menggunakan fungsi aktivasi ReLU untuk mengekstraksi fitur-fitur dari gambar.
- c. **MaxPooling Layers:** Setelah setiap lapisan konvolusi, dilakukan operasi max pooling dengan filter berukuran 2x2 piksel untuk mengurangi dimensi gambar dan menjaga informasi fitur yang penting.
- d. **Flatten Layer:** Lapisan ini mengubah output dari lapisan konvolusi menjadi vektor satu dimensi untuk dimasukkan ke dalam lapisan-lapisan terhubung penuh (fully connected layers).

e. **Dense Layers:** Terdapat dua lapisan terhubung penuh dengan masing-masing 512 neuron dan 3 neuron (untuk output kelas yang dihasilkan oleh model). Fungsi aktivasi ReLU digunakan pada lapisan terhubung penuh pertama, sementara lapisan terhubung penuh terakhir menggunakan fungsi aktivasi softmax untuk menghasilkan probabilitas kelas output.

2. Model from Scratch with Hyperparameter Tuning

Kami menggunakan arsitektur model neural network yang dikembangkan menggunakan TensorFlow dan Keras yang diambil dari model pertama. Model ini dirancang untuk melakukan klasifikasi gambar pada dataset yang memiliki dimensi gambar 224x224 piksel dan 3 saluran warna (RGB). Arsitektur model terdiri dari lapisan-lapisan berikut:

Selain itu, kami mengganti dropout layer dengan menerapkan regularisasi L2 dengan parameter 0.01 pada lapisan-lapisan konvolusi dan lapisan terhubung penuh pertama untuk mencegah overfitting selama training model. Kami juga menggunakan teknik data augmentation pada dataset pelatihan untuk memperluas variasi data yang digunakan dalam proses pelatihan. Model ini diharapkan dapat memberikan hasil yang baik dalam klasifikasi gambar pada dataset yang digunakan.

3. Model pre-trained MobileNetV3Large

Dalam analisis model ini, kami mengadaptasi arsitektur dari model pretrained MobileNetV3Large sebagai model final dari poin 5. Namun, perbedaan utamanya terletak pada tahap percobaan pertama di mana kami **tidak menyertakan lapisan dropout** dalam arsitektur model. Hal ini kami lakukan untuk mengidentifikasi pengaruh langsung dari penambahan lapisan dropout terhadap performa dan generalisasi model dalam konteks klasifikasi gambar. Dengan demikian, percobaan ini memberikan wawasan yang lebih mendalam tentang kontribusi lapisan dropout terhadap efektivitas model yang kami kembangkan.

4. Model pre-trained MobileNetV3Large with Hyperparameter Tuning

Pada model kedua, kami melakukan pengembangan lebih lanjut dengan melakukan tuning hyperparameter dari model trial pertama. Meskipun demikian, terdapat kekurangan yang perlu diperbaiki, antara lain **absennya metode early stopping dan penurunan learning rate (reduced learning rate)**. Kedua teknik ini penting untuk mengoptimalkan pelatihan model dan mencegah overfitting. Dengan mengimplementasikan early stopping, kami dapat menghentikan pelatihan secara otomatis saat performa model sudah mencapai puncaknya pada dataset validasi, sehingga menghindari overfitting pada dataset pelatihan. Selain itu, penggunaan reduced learning rate akan membantu menyesuaikan laju pembelajaran model saat performa mulai menurun, sehingga memperbaiki konvergensi dan stabilitas model secara keseluruhan. Dengan melakukan perbaikan ini, kami berharap dapat meningkatkan performa dan generalisasi model pada tugas klasifikasi gambar yang kami hadapi.

5. Model pre-trained MobileNetV3Large Last Model

Dalam melakukan pembangunan dua model data scratch dan model yang menggunakan transfer learning akan menggunakan arsitektur MobileNet V3 Large dikarenakan sangat efisien dalam melakukan pengklasifikasian gambar seperti yang tertera pada tabel berikut

Classification Checkpoint	MACs(M)	Parameters(M)	Top1 Accuracy	Pixel1 CPU(ms)
mobilenet_v3_large_1.0_224	217	5.4	75.6	51.2

Gambar 2.6 Deskripsi MobileNet

Arsitektur ini memiliki 217 juta Multiply-Accumulate Operations (MACs), 5.4 juta parameter, dan mencapai akurasi Top-1 sebesar 75.6% dengan waktu inferensi rata-rata 51.2 milidetik pada CPU Pixel1.

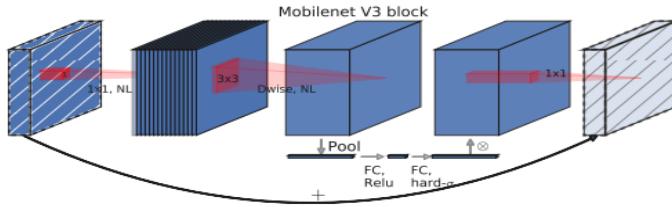


Figure 4. MobileNetV2 + Squeeze-and-Excite [20]. In contrast with [20] we apply the squeeze and excite in the residual layer. We use different nonlinearity depending on the layer, see section 5.2 for details.

Gambar 2.7 Blok Arsitektur MobileNetV3

- **Blok MobileNetV3:** Terdiri dari ekspansi (konvolusi 1x1), konvolusi *depthwise* (3x3), modul SE, dan proyeksi (konvolusi 1x1).
- **Squeeze-and-Excite:** Diintegrasikan ke dalam lapisan residual *bottleneck* untuk meningkatkan daya representasi jaringan dengan mengkalibrasi ulang respons fitur per saluran.
- **Non-linearitas:** Berbagai fungsi aktivasi digunakan (ReLU, h-swish) pada berbagai tahap dalam blok.

Input	Operator	exp size	#out	SE	NL	s
$224^2 \times 3$	conv2d	-	16	-	HS	2
$112^2 \times 16$	bneck, 3x3	16	16	-	RE	1
$112^2 \times 16$	bneck, 3x3	64	24	-	RE	2
$56^2 \times 24$	bneck, 3x3	72	24	-	RE	1
$56^2 \times 24$	bneck, 5x5	72	40	✓	RE	2
$28^2 \times 40$	bneck, 5x5	120	40	✓	RE	1
$28^2 \times 40$	bneck, 5x5	120	40	✓	RE	1
$28^2 \times 40$	bneck, 3x3	240	80	-	HS	2
$14^2 \times 80$	bneck, 3x3	200	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	184	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	184	80	-	HS	1
$14^2 \times 80$	bneck, 3x3	480	112	✓	HS	1
$14^2 \times 112$	bneck, 3x3	672	112	✓	HS	1
$14^2 \times 112$	bneck, 5x5	672	160	✓	HS	2
$7^2 \times 160$	bneck, 5x5	960	160	✓	HS	1
$7^2 \times 160$	bneck, 5x5	960	160	✓	HS	1
$7^2 \times 160$	conv2d, 1x1	-	960	-	HS	1
$7^2 \times 960$	pool, 7x7	-	-	-	-	1
$1^2 \times 960$	conv2d 1x1, NBN	-	1280	-	HS	1
$1^2 \times 1280$	conv2d 1x1, NBN	-	k	-	-	1

Gambar 2.8 Summary Arsitektur MobileNetV3

Input: Ukuran tensor input.

Operator: Jenis operasi yang dilakukan (misalnya, konvolusi, bottleneck).

exp size: Ukuran ekspansi sebelum konvolusi depthwise.

#out: Jumlah saluran output.

SE: Menunjukkan apakah Squeeze-and-Excite digunakan (✓ jika ada).

NL: Jenis non-linearitas (HS untuk h-swish, RE untuk ReLU).

s: Langkah (stride) konvolusi (mempengaruhi downsampling).

- **Implementasi Code**

MobileNetV3Large diimplementasikan melalui TensorFlow dan Keras. Inisiasi model dasar MobileNetV3Large dilakukan dengan mengatur parameter-parameter kritis seperti dimensi input (224x224x3), nilai alpha, serta konfigurasi tambahan. Model dasar ini tidak mencakup lapisan-lapisan teratas dan mengadopsi bobot pra-terlatih dari dataset ImageNet. Seluruh lapisan dalam model dasar dikonfigurasi sebagai non-trainable guna mempertahankan bobot pra-training.

Selanjutnya, lapisan input didefinisikan sesuai dengan spesifikasi yang telah ditentukan (224,224,3). Model dasar kemudian diaplikasikan dalam mode non-trainable (*freeze*) untuk menjaga integritas bobotnya. Untuk mengatasi potensi overfitting, sebuah lapisan dropout dengan tingkat 0.3 diintegrasikan, diikuti oleh lapisan dense yang terdiri dari tiga neuron dengan fungsi aktivasi softmax untuk tujuan klasifikasi. Kompilasi model dilakukan dengan menggunakan optimizer Adam, *learning rate* 0.1, *loss function* “categorical cross entropy”, serta metric “accuracy” sebagai indikator performa.

Untuk mengoptimalkan kinerja model, tiga mekanisme callback diimplementasikan: ModelCheckpoint, EarlyStopping, dan ReduceLROnPlateau.

ModelCheckpoint berfungsi untuk menyimpan model terbaik berdasarkan nilai loss validasi terendah. EarlyStopping diatur untuk menghentikan proses *training* apabila tidak terjadi perbaikan pada loss validasi selama 15 epoch berturut-turut. ReduceLROnPlateau berperan dalam menurunkan *learning rate* dengan faktor 0.1 jika tidak ada peningkatan pada loss validasi selama 15 epoch, dengan batas minimum *learning rate* ditetapkan pada 0.0001.

Proses training model dilaksanakan menggunakan data training dan validasi selama 50 epoch. Mekanisme callback yang telah diimplementasikan berfungsi untuk memonitor dan menyesuaikan parameter training sesuai dengan kriteria yang telah ditetapkan, sehingga memungkinkan optimalisasi performa model.

III. Hasil dan Analisis

Berikut ini adalah tabel akurasi dari keseluruhan model yang sudah kami buat.

Model	Accuracy
Model from Scratch	0.6428571343421936
Model from Scratch (Tuned)	0.8571428656578064
Model pre-trained MobileNetV3Large	0.8571428656578064
Model pre-trained MobileNetV3Large (Tuned)	0.6428571343421936
Model pre-trained MobileNetV3Large (Final Model)	0.8571428656578064

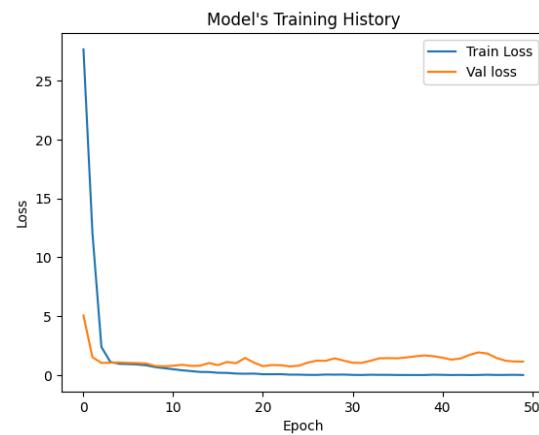
Tabel 1 Evaluasi Model

Dari Tabel 1, kita dapat melihat berbagai nilai akurasi dari model yang telah dibuat. Meskipun nilai-nilai tersebut memberikan gambaran awal, mereka tidak bisa dijadikan patokan tunggal untuk menilai kualitas model. Untuk melakukan analisis yang

komprehensif terhadap performa model, kita perlu mempertimbangkan evaluasi lainnya. Hal ini mencakup metrik seperti precision, recall, F1-score, serta analisis terhadap grafik loss dan validasi. Dengan melihat berbagai metrik dan evaluasi tersebut, kita dapat menentukan apakah model mengalami underfitting, overfitting, atau sudah optimal.

1. Model from Scratch

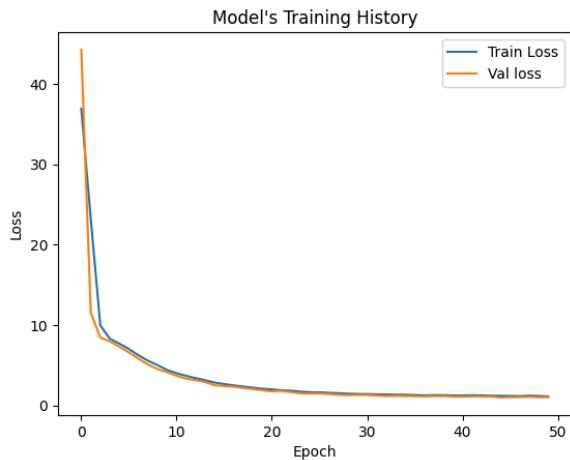
Kami melakukan analisis dari model yang sudah dijelaskan di bagian sebelumnya. Model yang kami gunakan ini adalah architecture dengan menggunakan dropout.



Gambar 3.1 Grafik Loss Model Scratch

Dari Gambar 3.1, kita dapat melihat bahwa validation loss lebih tinggi dibandingkan dengan training loss. Perbedaan ini menunjukkan bahwa model kita mengalami overfitting terhadap data training, yang berarti model terlalu menyesuaikan diri dengan data training dan tidak mampu menggeneralisasi dengan baik pada data baru.

2. Model from Scratch with Hyperparameter Tuning



Gambar 3.2 Grafik Loss Model Scratch with Tuning

Kita bisa melihat hasil dari grafik loss model scratch kami setelah dilakukan tuning. Hasilnya menunjukkan perbaikan yang signifikan karena training loss dan validation loss sudah lebih seimbang. Hal ini menandakan bahwa model kita tidak lagi mengalami overfitting dan mampu menggeneralisasi dengan baik pada data baru. Dengan training loss dan validation loss yang hampir setara, kita dapat menyimpulkan bahwa model telah berhasil belajar pola dari data training tanpa menghafal rincian spesifik yang tidak relevan.

	precision	recall	f1-score	support
Parang	0.00	0.00	0.00	2
Megamendung	0.86	1.00	0.92	6
Kawung	0.86	1.00	0.92	6
micro avg	0.86	0.86	0.86	14
macro avg	0.57	0.67	0.62	14
weighted avg	0.73	0.86	0.79	14
samples avg	0.86	0.86	0.86	14

Gambar 3.3 Classification Report Model Scratch

Kita bisa melihat hasil dari classification report dari Gambar 3.6 dan untuk penjelasan dari masing-masing valuenya adalah sebagai berikut”

a. Precision

Precision adalah metrik yang mengukur seberapa akurat model dalam memprediksi positif. Dalam konteks klasifikasi, precision mengacu pada kemampuan model untuk memprediksi kelas dengan benar. Precision dinyatakan sebagai rasio antara true positive (prediksi benar) dan total prediksi positif (true positive + false positive).

Precision untuk kelas 'Parang' adalah 0.50, yang berarti dari semua prediksi yang dilakukan untuk kelas 'Parang', hanya 50% yang benar. Precision untuk kelas 'Megamendung' adalah 0.86, artinya sebanyak 86% dari prediksi untuk kelas 'Megamendung' adalah benar. Sedangkan precision untuk kelas 'Kawung' adalah 1.00, yang menunjukkan bahwa model memprediksi kelas 'Kawung' dengan sempurna.

b. Recall

Recall, juga dikenal sebagai sensitivity, mengukur seberapa banyak data positif yang dapat diidentifikasi oleh model. Recall dinyatakan sebagai rasio antara true positive dan total data sebenarnya yang positif (true positive + false negative).

Recall untuk kelas 'Parang' adalah 0.50, yang berarti dari semua data sebenarnya yang termasuk ke dalam kelas 'Parang', model dapat mengidentifikasi 50% dari data tersebut. Untuk kelas 'Megamendung', recall mencapai 1.00, yang berarti model dapat mengidentifikasi semua data yang sebenarnya termasuk ke dalam kelas 'Megamendung'. Sedangkan untuk kelas 'Kawung', recall adalah 0.83, menunjukkan bahwa model dapat mengidentifikasi 83% dari data yang sebenarnya termasuk ke dalam kelas 'Kawung'.

c. F1-score

F1-score adalah harmonic mean dari precision dan recall, yang berguna untuk menilai keseimbangan antara precision dan recall. F1-score biasanya digunakan ketika kita ingin menemukan keseimbangan antara dua metrik ini.

F1-score untuk kelas 'Parang' adalah 0.50, menunjukkan keseimbangan antara precision dan recall untuk kelas 'Parang'. Untuk kelas 'Megamendung', F1-score mencapai 0.92, menunjukkan keseimbangan yang sangat baik antara precision dan recall. Sedangkan untuk kelas 'Kawung', F1-score adalah 0.91, menunjukkan keseimbangan yang baik antara precision dan recall juga.

d. Support

Support mengacu pada jumlah data yang sebenarnya termasuk dalam setiap kelas. Terdapat 2 data yang sebenarnya termasuk ke dalam kelas 'Parang', 6 data dalam kelas 'Megamendung', dan 6 data dalam kelas 'Kawung'.

e. Micro avg

Dalam Micro Average, nilai precision, recall, dan F1-score mencapai 0.86, menunjukkan keseimbangan yang baik antara kemampuan model dalam memprediksi dengan benar dan mengidentifikasi data dengan benar secara keseluruhan dari semua prediksi dan data sebenarnya yang diuji, yang jumlahnya mencapai 14 data.

f. Macro avg

Sementara itu, Macro Average memberikan rata-rata performa dari setiap kelas secara terpisah, dengan precision, recall, dan F1-score masing-masing mencapai 0.79, 0.78, dan 0.78.

g. Weighted avg

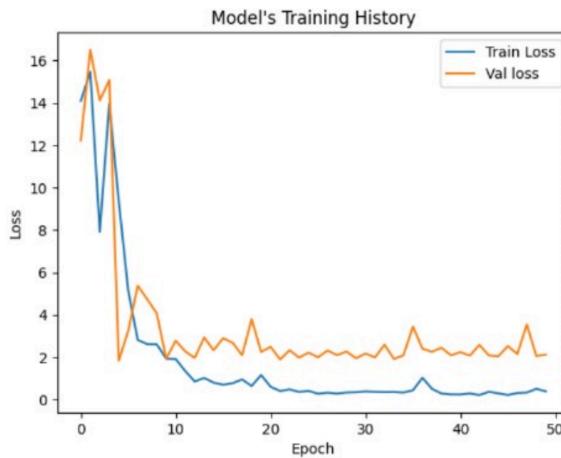
Weighted Average juga memberikan rata-rata performa dari setiap kelas, namun dengan mempertimbangkan jumlah data pada setiap kelas. Nilai precision, recall, dan F1-score pada Weighted Average adalah 0.87, 0.86, dan 0.86.

h. Samples avg

Samples Average memberikan gambaran performa dari sudut pandang setiap sampel data yang diuji. Dalam Samples Average,

nilai precision, recall, dan F1-score juga mencapai 0.86, dengan jumlah total sampel yang diuji mencapai 14 data. Dengan demikian, nilai-nilai evaluasi ini memberikan insight tentang keseimbangan performa model dalam melakukan klasifikasi untuk setiap kelas dan secara keseluruhan dari semua data yang diuji.

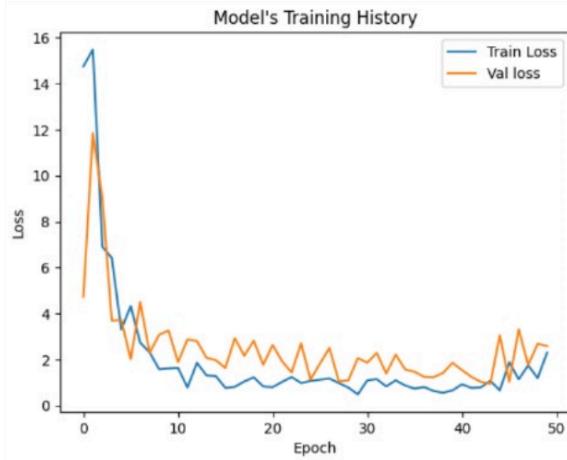
3. Model pre-trained MobileNetV3Large



Gambar 3.4 Grafik Loss Model pre-trained

Kita bisa melihat hasil dari grafik di atas yang masih menunjukkan tanda-tanda overfitting, karena validation loss lebih tinggi dibandingkan dengan training loss. Hal ini mungkin disebabkan oleh tidak adanya dropout layer dalam arsitektur model. Dropout layer dapat membantu mengurangi overfitting dengan mencegah model dari terlalu bergantung pada neuron-neuron tertentu selama training. Dengan menambahkan dropout layer, model dapat lebih baik dalam menggeneralisasi pola dari data training ke data validasi, sehingga mengurangi perbedaan antara training loss dan validation loss.

4. Model pre-trained MobileNetV3Large (Tuned)



Gambar 3.5 Grafik Loss Model pre-trained with Tuning

Kita bisa melihat bahwa grafik training loss dan validation loss menunjukkan perbaikan yang signifikan akibat tuning, karena keduanya sudah lebih setara. Namun, di akhir grafik terlihat bahwa validation loss dan training loss mulai meningkat. Hal ini menandakan bahwa model kita mulai overfit kembali terhadap data training. Kemungkinan penyebabnya adalah tidak adanya early stopping dan reduced learning rate. Early stopping dapat membantu menghentikan training sebelum model mulai overfit, sementara reduced learning rate dapat membantu model untuk terus belajar dengan lebih hati-hati pada tahap akhir training, mengurangi risiko overfitting.

5. Model pre-trained MobileNetV3Large (Last Model)



Gambar 3.6 Grafik Loss Model pre-trained Final Model

Kita bisa melihat pada grafik Train Loss dan data Validation Loss selama 20 epoch. Pada awal training, Train Loss dan Validation Loss mulai dari nilai yang tinggi dan mengalami penurunan tajam dalam beberapa epoch pertama, menunjukkan bahwa model dengan cepat belajar pola dasar dalam data. Seiring berjalananya waktu, laju penurunan menjadi lebih lambat dan mulai stabil. Setelah sekitar 10 epoch, perbedaan antara Train Loss dan Validation Loss menjadi lebih konsisten dan relatif rendah, menunjukkan bahwa model tidak mengalami overfitting yang signifikan dan performanya cukup stabil pada data validasi. Pada akhirnya, kedua loss mendekati nilai yang rendah dan konsisten, menandakan bahwa model telah mencapai konvergensi.

	precision	recall	f1-score	support
Parang	0.00	0.00	0.00	2
Megamendung	0.86	1.00	0.92	6
Kawung	0.86	1.00	0.92	6
micro avg	0.86	0.86	0.86	14
macro avg	0.57	0.67	0.62	14
weighted avg	0.73	0.86	0.79	14
samples avg	0.86	0.86	0.86	14

Gambar 3.7 Classification Report Model Scratch

Kita bisa melihat hasil dari classification report dari Gambar 3.6 dan untuk penjelasan dari masing-masing valuenya adalah sebagai berikut”

a. Precision

Precision adalah metrik yang mengukur seberapa akurat model dalam memprediksi positif. Pada kelas 'Parang', nilai precision mencapai 0.00, menunjukkan bahwa dari semua prediksi yang dilakukan untuk kelas 'Parang', tidak ada yang benar. Sementara itu, kelas 'Megamendung' dan 'Kawung' memiliki nilai precision yang baik, yaitu 0.86, yang berarti sebanyak 86% dari prediksi untuk kedua kelas tersebut adalah benar. Precision yang

tinggi menandakan kemampuan model dalam meminimalkan kesalahan prediksi positif.

b. Recall

Recall, juga dikenal sebagai sensitivity, mengukur seberapa banyak data positif yang dapat diidentifikasi oleh model. Pada kelas 'Parang', recall mencapai 0.00, mengindikasikan bahwa model tidak dapat mengidentifikasi data sebenarnya yang termasuk ke dalam kelas tersebut. Namun, kelas 'Megamendung' dan 'Kawung' mencapai nilai recall yang sangat baik, yaitu 1.00, menunjukkan bahwa model dapat mengidentifikasi semua data yang sebenarnya termasuk ke dalam kedua kelas tersebut. Recall yang tinggi menunjukkan kemampuan model dalam meminimalkan kesalahan prediksi negatif.

c. F1-score

F1-score adalah harmonic mean dari precision dan recall, yang memberikan gambaran tentang keseimbangan antara keduanya. Untuk kelas 'Parang', F1-score adalah 0.00, menunjukkan keseimbangan yang sangat rendah antara precision dan recall. Sementara itu, kelas 'Megamendung' dan 'Kawung' memiliki F1-score yang baik, yaitu 0.92, menunjukkan keseimbangan yang baik antara precision dan recall untuk kedua kelas tersebut. F1-score yang tinggi menunjukkan performa yang seimbang antara kemampuan model dalam memprediksi dan mengidentifikasi data dengan benar.

d. Support

Support dalam konteks evaluasi klasifikasi mengacu pada jumlah data yang sebenarnya termasuk dalam setiap kelas. Pada hasil evaluasi yang diberikan, support untuk kelas 'Parang' adalah 2, yang berarti terdapat 2 data yang termasuk ke dalam kelas 'Parang' pada data yang diuji. Sementara itu, kelas 'Megamendung' dan 'Kawung' masing-masing memiliki support sebesar 6,

menunjukkan bahwa terdapat 6 data yang termasuk ke dalam kedua kelas tersebut pada data yang diuji.

e. Micro avg

Micro Average mengukur performa keseluruhan dari model dalam memprediksi dan mengidentifikasi data dengan benar di antara semua prediksi positif yang dilakukan dan semua data sebenarnya yang positif. Nilai precision, recall, dan F1-score pada Micro Average mencapai 0.86, menunjukkan bahwa model memiliki keseimbangan yang baik antara precision dan recall secara keseluruhan dari seluruh data yang diuji, yang berjumlah 14 data.

f. Macro avg

Macro Average memberikan rata-rata performa dari setiap kelas secara terpisah. Dalam Macro Average, nilai precision, recall, dan F1-score masing-masing mencapai 0.57, 0.67, dan 0.62. Meskipun nilai recall untuk setiap kelas cukup baik (0.86 untuk 'Megamendung' dan 'Kawung'), nilai precision yang rendah untuk kelas 'Parang' (0.00) menyebabkan rata-rata precision secara keseluruhan menjadi rendah.

g. Weighted avg

Weighted Average juga memberikan rata-rata performa dari setiap kelas, dengan mempertimbangkan jumlah data pada setiap kelas. Nilai precision, recall, dan F1-score pada Weighted Average adalah 0.73, 0.86, dan 0.79. Weighted Average menghasilkan nilai yang lebih tinggi daripada Macro Average karena mempertimbangkan distribusi data yang lebih merata antara kelas-kelas.

h. Samples avg

Samples Average memberikan gambaran performa dari sudut pandang setiap sampel data yang diuji. Dalam Samples Average, nilai precision, recall, dan F1-score mencapai 0.86,

dengan jumlah total sampel yang diuji mencapai 14 data. Dengan demikian, nilai-nilai evaluasi ini memberikan insight tentang keseimbangan performa model dalam melakukan klasifikasi untuk setiap kelas dan secara keseluruhan dari semua data yang diuji, meskipun ada peningkatan yang dapat dilakukan terutama pada precision untuk kelas 'Parang'.

IV. Conclusion

Proyek ini menampilkan keefektifitasan model deep learning dalam mengklasifikasikan Batik Kawung, Batik Megamendung, dan Batik Parang. Kami menggunakan dua pendekatan: model scratch dan transfer learning dengan MobileNetV3Large. Proses preparasi data meliputi pembersihan, resizing, normalisasi, dan augmentasi. Hasil evaluasi menunjukkan bahwa model transfer learning dengan MobileNetV3Large dan model scratch setara dengan akurasi 85,71%

Namun, tuning hyperparameter tetap diperlukan untuk mengoptimalkan kinerja model dan mencegah overfitting. Penggunaan dropout layer, early stopping, dan penyesuaian learning rate efektif dalam mengurangi overfitting. Kesimpulannya, deep learning adalah alat yang kuat untuk klasifikasi gambar, dan pendekatan transfer learning dengan MobileNetV3Large adalah pilihan optimal untuk akurasi dan efisiensi dalam klasifikasi gambar yang kompleks.

V. Reference

MobileNet, MobileNetV2, and MobileNetV3. (n.d.).
<https://keras.io/api/applications/mobilenet/>

Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., Le, Q. V., & Adam, H. (2019). SearchingforMobileNetV3.