

CLASSIFICATION OF BATIK MOTIF USING TRANSFER LEARNING ON CONVOLUTIONAL NEURAL NETWORK (CNN)

Riqqah Fadiyah Alya^{*1}, Merlinda Wibowo², Paradise³

^{1,2,3}Teknik Informatika, Fakultas Informatika, Institut Teknologi Telkom Purwokerto, Indonesia
Email: ¹18102031@ittelkom-pwt.ac.id, ²merlinda@ittelkom-pwt.ac.id, ³paradise@ittelkom-pwt.ac.id

(Naskah masuk: 25 Juni 2022, Revisi: 6 Juli 2022, Diterbitkan: 10 Februari 2023)

Abstract

The number of batik motifs in Indonesia is not comparable to the knowledge possessed by the Indonesian people about batik motifs. The diversity of batik motifs can be a problem because classifying them can only be done by those who are familiar with batik in depth, both the pattern and the philosophy behind the motif, most of which are elderly people. To classify batik accurately and quickly is to use image classification technology. In this study, data were obtained from the previous researchers' GitHub repository, google images, and camera shots with a total dataset of 3,534 images. The data only focused on five batik motifs, namely Ceplok, Kawung, Parang, Megamendung, and Sidomukti. Before the batik motif is processed, preprocessing is carried out to obtain various quality data. Then the dataset was trained using the CNN model then the results were retrained using the VGG-16 and Xception Transfer Learning models. The researcher made several model scenarios, namely the CNN model without Transfer Learning and the model with Transfer Learning which took into account the effect of the learning rate values of 0.0004 and 0.0001. Therefore, the results of the CNN model without Transfer Learning (M0) obtained training accuracy results of 89.64%. While the results of the model with the best Transfer Learning is the M2 model (CNN + VGG-16, learning rate = 0.0001) with an accuracy of 91.23%, a loss of 24.48%, and the test results obtained an accuracy of 89%. Based on the results of the classification method, it can be concluded that the CNN model with Transfer Learning performs classification better in terms of accuracy and computation time than the CNN model.

Keywords: Batik, CNN, Transfer Learning, VGG-16, Xception

KLASIFIKASI MOTIF BATIK MENGGUNAKAN TRANSFER LEARNING PADA CONVOLUTIONAL NEURAL NETWORK (CNN)

Abstrak

Banyaknya motif batik yang ada di Indonesia tidak sebanding dengan pengetahuan yang dimiliki oleh masyarakat Indonesia mengenai motif batik. Keberagaman motif batik ini dapat menjadi masalah karena untuk mengklasifikasikannya hanya dapat dilakukan oleh pihak yang sudah mengenal batik secara mendalam baik pola maupun filosofi di balik motif tersebut, dimana kebanyakan adalah pihak yang sudah lanjut usia. Untuk mengklasifikasikan batik secara akurat dan cepat adalah menggunakan teknologi klasifikasi citra. Pada penelitian ini, data diperoleh dari repository *GitHub* peneliti terdahulu, *google image* dan jepretan melalui kamera dengan total dataset berjumlah 3.534 citra. Data yang diambil hanya berfokus pada lima motif batik yaitu Ceplok, Kawung, Parang, Megamendung dan Sidomukti. Sebelum motif batik itu diolah, dilakukan *preprocessing* untuk mendapatkan data yang beragam dan berkualitas. Kemudian dataset tersebut dilatih menggunakan model CNN yang kemudian hasilnya dilatih ulang dengan model *Transfer Learning* VGG-16 dan Xception. Peneliti membuat beberapa skenario model yaitu model CNN tanpa *Transfer Learning* dan model dengan *Transfer Learning* yang memperhatikan pengaruh dari nilai *learning rate* 0,0004 dan 0,0001. Sehingga didapatkan hasil model CNN tanpa *Transfer Learning* (M0) memperoleh hasil akurasi pelatihan sebesar 89,64%. Sedangkan hasil model dengan *Transfer Learning* terbaik yaitu model M2 (CNN + VGG-16, *learning rate* = 0,0001) dengan hasil akurasi 91,23%, *loss* 24,48%, dan hasil pengujian diperoleh hasil akurasi sebesar 89%. Berdasarkan hasil dari metode klasifikasi dapat ditarik kesimpulan bahwa model CNN dengan *Transfer Learning* melakukan performa klasifikasi lebih baik dari nilai akurasi dan waktu komputasi daripada hanya menggunakan model CNN.

Kata kunci: Batik, CNN, Transfer Learning, VGG-16, Xception

1. PENDAHULUAN

Batik merupakan ekspresi budaya yang memiliki makna simbolis dan nilai estetika yang tinggi bagi masyarakat Indonesia. Pada 2 Oktober 2009, UNESCO menetapkan keberadaan dan keunikan batik sebagai Warisan Kemanusiaan untuk Budaya Lisan dan Nonbendawi (*Masterpieces of the Oral and Intangible Heritage of Humanity*) [1] [2]. Batik merupakan warisan budaya yang tidak hanya disebabkan oleh batik itu sendiri, namun juga seni pembuatannya.

Budaya batik yang sifatnya budaya turun menurun menyebabkan beberapa motif batik dapat dikenali berdasarkan asal daerah atau asal keluarganya, misalnya motif yang dibuat oleh kalangan bangsawan dengan motif yang dibuat oleh masyarakat biasa. Sehingga hampir setiap daerah memiliki motif khasnya sendiri dan banyak seniman yang mengembangkan motif baru untuk membuat batik berkelanjutan [3].

Namun banyak masyarakat Indonesia modern yang tidak mengetahui jenis atau klasifikasi batik yang mereka gunakan atau lihat karena batik diklasifikasikan oleh nenek moyang Indonesia tetapi kebanyakan pengetahuan akan hal tersebut tidak selalu diturunkan kepada anak cucu mereka. Dokumentasi lengkap untuk batik Indonesia diperlukan untuk menghindari masalah ini [4]. Citra batik perlu diklasifikasikan karena memiliki motif yang beragam [5], dapat diambil dari berbagai macam sumber [4] dan motif batik terus berkembang dari waktu ke waktu [1].

Tantangan klasifikasi tidak hanya berasal dari beragamnya motif batik, namun masalah pada citra batik yang memiliki berbagai kondisi karena dapat berasal dari internet atau hasil jepretan sendiri juga melatarbelakangi dibutuhkannya teknologi yang dapat melakukan klasifikasi batik [6].

Convolutional Neural Network (CNN) merupakan salah satu algoritma deep learning yang merepresentasikan perkembangan multi-layer perceptrons (MLP) [7]. CNN merupakan salah satu model yang paling populer dan akurat yang diimplementasikan dalam klasifikasi citra [8]. Berdasarkan penelitian [9], [10] menunjukkan bahwa model CNN dapat digunakan untuk klasifikasi batik namun model CNN yang dibuat sendiri belum dapat memperoleh akurasi yang andal dalam klasifikasi batik. Maka dari itu, untuk mengatasi hal tersebut peneliti menerapkan transfer learning.

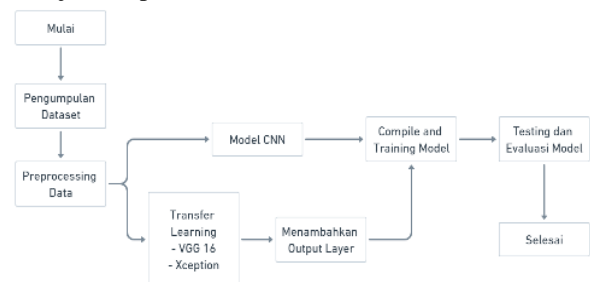
Transfer learning merupakan teknik deep learning yang menggunakan model yang dilatihkan pada satu masalah untuk memecahkan masalah yang lain. Pembelajaran transfer memungkinkan pelatihan pembelajaran mendalam dengan sejumlah kecil sampel dengan akurasi tinggi [11]. Model pra-pelatihan yang telah digunakan untuk mengklasifikasikan batik adalah VGG-16 dan VGG-19 [12], [4], [6], [13]. Berdasarkan penelitian tersebut dapat disimpulkan bahwa dengan menggunakan

model pra latih menghasilkan akurasi yang lebih tinggi daripada hanya menggunakan model CNN yang dibuat sendiri, sehingga peneliti mengklasifikasikan batik menggunakan model pra latih VGG16 dan Xception sesuai dengan referensi yang diberikan oleh penelitian terdahulu.

Bersumber pada masalah yang ada dan penelitian-penelitian yang telah dilakukan, masih terdapat beberapa kekurangan seperti permasalahan dataset, baik dalam jumlah dataset hingga variasi dari dataset yang digunakan, ekstraksi fitur yang digunakan, pelabelan data, *preprocessing* yang dilakukan hingga arsitektur model yang dibangun. Berdasarkan penjelasan di atas, maka peneliti mengambil judul penelitian "Klasifikasi Motif Batik Menggunakan *Transfer Learning* Pada *Convolutional Neural Network* (CNN)". Hasil dari penelitian ini diharapkan dapat menjadi acuan pembangunan model klasifikasi motif batik agar menghasilkan akurasi dan performa model yang baik serta dapat berkontribusi dalam pelestarian warisan budaya batik.

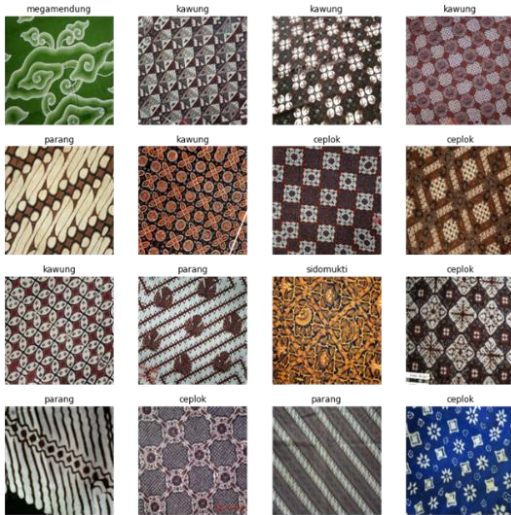
2. METODE PENELITIAN

Metode penelitian yang dilakukan pada penelitian ini dapat dilihat dari diagram yang ditunjukkan pada Gambar 1.



Gambar 1. Diagram Alir Penelitian

Berdasarkan diagram alir penelitian di atas, diketahui bahwa penelitian ini akan melalui empat tahapan. Langkah pertama adalah pengumpulan data. Data yang digunakan peneliti dalam penelitian ini adalah citra batik digital yang terdiri dari lima motif yang berbeda yaitu Ceplok, Megamendung, Parang, Kawung, dan Sidomukti. Data dikumpulkan dari berbagai sumber, termasuk *crowdsourcing*, kumpulan data dari peneliti sebelumnya, unduhan gambar *Google*, dan rekaman kamera. Hasilnya, ada 3.534 gambar yang terdiri dari 850 dataset untuk Ceplok, 760 dataset Kawung, 1097 dataset Parang, 560 dataset Megamendung, dan 267 dataset Sidomukti. Gambar 2 di bawah ini adalah contoh tampilan citra motif batik pada dataset yang digunakan.:



Gambar 2. Contoh Citra Motif Batik yang Digunakan

Langkah kedua adalah pengolahan data (*preprocessing*). Tahap *preprocessing* ini terdiri dari tiga tahap: pembagian dataset, *input image cropping*, dan augmentasi data. Pembagian dataset dibagi menjadi 80% data pelatihan, 10% data pengujian, dan 10% data validasi. Proses *cropping* citra input dilakukan pada ukuran 224 x 224 piksel, dengan *batch size* 4 untuk data latih, 2 untuk data uji, dan 1 untuk data validasi. Pemotongan gambar dilakukan karena perbedaan ukuran gambar yang terdapat dalam dataset. *Batch size* adalah jumlah sampel data yang didistribusikan melalui jaringan saraf. Seluruh kumpulan data tidak dapat dilewatkan ke jaringan saraf sekaligus, sehingga kumpulan data harus dibagi menjadi bagian-bagian atau kumpulan bagian. *Batch size* ditentukan oleh faktor jumlah gambar di setiap data pelatihan, pengujian, dan validasi. Kemudian untuk augmentasi data dilakukan dengan memanfaatkan Keras API untuk data *preprocessing* yaitu *Image Data Generator*. Terdapat delapan parameter augmentasi data yang digunakan yaitu: *rescale* 1/255, *rotation* 45 derajat, *width shift* 0.25, *height shift* 0.25, *shear* 0.25, *zoom* 0.25, *horizontal flip*, dan *fill mode*.

Selanjutnya yaitu pembangunan model. Pada penelitian ini ada lima model yang dibuat, untuk lebih jelasnya dapat lihat Tabel 1 di bawah ini.

Tabel 1. Model yang Dibuat Pada Penelitian

Kode model	Fungsi optimizer	Learning Rate	Fungsi loss	arsitektur
M0	Adam	0.0004	Categorical crossentropy	CNN
M1	Adam	0.0001	Categorical crossentropy	CNN, VGG-16
M2	Adam	0.0004	Categorical crossentropy	CNN, VGG-16
M3	Adam	0.0001	Categorical crossentropy	CNN, Xception
M4	Adam	0.0004	Categorical crossentropy	CNN, Xception

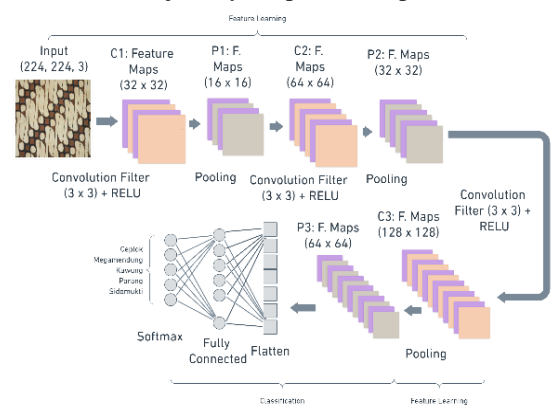
Pada tahap ini, membangun model deep learning dimulai dengan menerapkan teknik transfer learning menggunakan arsitektur CNN, VGG-16, dan Xception. Metode *pre-trained model* yang digunakan pada penelitian ini adalah Fine Tuning. Fine Tuning menggunakan pre-trained model yang akan dilatih ulang dengan kebutuhan yang baru [14]. Kemudian menambahkan layer diakhir pre-trained dengan menambahkan layer DNN seperti Flatten dan Dense layer untuk mengubah output menjadi array satu dimensi yang selanjutnya dapat digunakan sebagai model klasifikasi. Selain itu, peneliti juga menambahkan teknik regularisasi yang disebut dropout untuk mengurangi overfitting dalam proses pelatihan.

Tahap terakhir yaitu pengujian model. Dalam melakukan pengujian terhadap model yang sudah dibangun, peneliti menguji model dengan data uji menggunakan confusion matrix untuk mengukur performa model dalam melakukan klasifikasi. Peneliti menggunakan metrik evaluasi yang lain seperti akurasi, presisi, *recall*, dan *F1-score*.

3. HASIL DAN PEMBAHASAN

3.1. Arsitektur Model CNN Tanpa Transfer Learning

Model pertama yang dibuat oleh peneliti yaitu model CNN yang belum mengimplementasikan transfer learning. Arsitektur CNN ini terbagi dalam 2 tahapan yaitu feature learning dan classification. Ukuran citra yang digunakan sebagai data input berukuran 224x224 dengan menggunakan channel RGB sehingga terdiri dari 3 channel. Penggambaran arsitektur lebih jelasnya dapat dilihat pada Gambar 3.



Gambar 3. Arsitektur Model CNN

Layer konvolusi diawali dengan proses konvolusi awal dengan menggunakan filter sejumlah 32 dan ukuran kernel 3x3. menginputkan gambar dengan ukuran 224x224x3, menunjukkan bahwa tinggi dan lebar dari suatu gambar adalah 224 piksel dan 3 menunjukkan channel warna yaitu red, green, dan blue yang biasa disebut sebagai RGB. Fungsi aktivasi yang digunakan pada konvolusi pertama adalah ReLu (*Rectified Linear Unit*).

Proses pooling, merupakan pengurangan ukuran matriks dengan menggunakan operasi pooling. Penelitian ini menggunakan max-pooling untuk mendapatkan nilai matriks yang baru berukuran 2x2 maka setiap maksimum nilai yang berada pada area 2x2 piksel akan dipilih setiap pergeseran filter.

Proses konvolusi kedua meneruskan hasil dari proses pooling pertama yakni dengan input matriks gambar sebesar 64x64 dengan jumlah filter sebanyak 64 dan dengan ukuran kernel 3x3. Proses konvolusi kedua ini sama-sama menggunakan fungsi aktivasi ReLU.

Proses selanjutnya adalah proses pooling yang kedua, proses ini hampir sama dengan proses pooling yang pertama, namun ada perbedaan pada nilai output akhir dari matriksnya. Output yang dihasilkan memiliki ukuran gambar 32x32.

Proses konvolusi ketiga menuruskan hasil dari proses pooling kedua yakni dengan input matriks citra sebesar 128x128 dan dengan ukuran kernel 3x3. Hasil dari proses konvolusi yang dilakukan mendapatkan output feature maps yang berbentuk multidimensional array.

Proses selanjutnya adalah proses pooling yang ketiga, proses ini hampir sama dengan proses pooling kedua, namun ada perbedaan pada nilai output akhir dari matriksnya. Output yang dihasilkan memiliki ukuran 64x64.

Selanjutnya dilakukan tahap fully connected. Pada tahap ini dilakukan proses Flatten yang digunakan untuk merubah final output pooling layer menjadi sebuah vektor. Pada dense layer menggunakan nilai 512 ini menunjukkan bahwa neuron yang digunakan pada jaringan konvolusi sebanyak 512. Sebelum melakukan proses klasifikasi, pada tahap ini memanfaatkan teknik regularisasi jaringan saraf yang biasa disebut dengan Dropout sebesar 0.2.

Pada layer terakhir, menggunakan dense layer dengan nilai 5 dan fungsi aktivasi Softmax. Angka 5 menunjukkan banyaknya kelas gambar yang akan diklasifikasi, yaitu Ceplok, Kawung, Parang, Megamendung, dan Sidomukti. Sementara penggunaan fungsi softmax digunakan pada proses klasifikasi dengan cara menghitung probabilitas tiap kelas target dari semua kelas target.

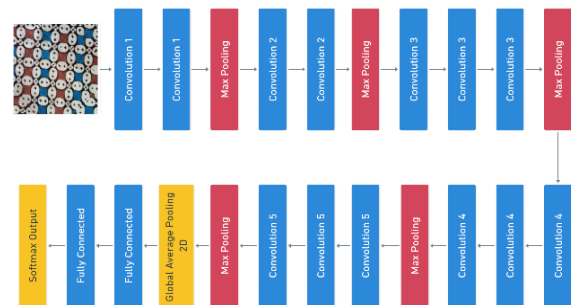
Setelah membuat arsitektur Convolutional Neural Network selanjutnya melakukan fungsi kompilasi untuk menentukan *optimizer*, *loss*, dan *metrics* yang akan digunakan. Seperti yang telah dijabarkan pada Tabel 1 dengan kode model M0, parameter optimasi yang digunakan adalah *Adam* dengan *learning rate* 0.0004. Parameter fungsi *loss* yang digunakan adalah *categorical_crossentropy*. *Loss categorical_crossentropy* digunakan untuk masalah klasifikasi multi-kelas. Kemudian parameter metrik yang digunakan untuk mengukur performa model adalah metrik akurasi.

3.2. Arsitektur Model CNN dengan Transfer Learning

Implementasi CNN dengan Transfer Learning pada penelitian ini menggunakan dua arsitektur berbeda yaitu arsitektur VGG-16 dan Xception serta menggunakan salah satu cara pre-trained model untuk transfer learning yaitu Fine Tuning.

3.2.1. Arsitektur CNN dengan VGG-16

Model kedua yang dibuat oleh peneliti yaitu model CNN yang mengimplementasikan pre-trained model VGG-16.



Gambar 4. Arsitektur Model CNN dengan VGG-16

Berdasarkan Gambar 4. pada base model yang dirancang dengan pre-trained model VGG-16, ada beberapa lapisan yang digunakan, yaitu lapisan masukan, lapisan Convolution, dan beberapa lapisan deep learning seperti lapisan Flatten, lapisan Dense sebagai fully-connected layers, dan lapisan luaran berupa lapisan Dense dengan fungsi aktivasi Softmax. Pre-trained model VGG-16 yang digunakan pada penelitian ini berasal dari model yang sudah dilatih dengan parameter weight yang diisi dengan imagenet. Hal tersebut dipilih dengan alasan jika tidak menggunakan pre-trained weight, proses training akan lebih lama karena initial value dari *weight* nya acak sehingga perlu banyak penyesuaian terhadap data yang akan digunakan. Kemudian include top diatur *False* artinya tidak menggunakan layer terakhir (prediksi) dalam *fully-connected layer* dari model VGG-16 yang mana menggunakan dataset dari imagenet karena jumlah kelas pada imagenet adalah 1000, ini tidak cocok dengan jumlah kelas yang digunakan pada penelitian ini yaitu hanya 5 kelas motif batik.

Dalam mengubah output layer yang dihasilkan oleh pre-trained model dari array multidimensi menjadi array satu dimensi atau vektor tunggal, peneliti memanfaatkan layer *Global Average Pooling* setelah layer pooling terakhir. Selanjutnya masuk ke fully connected layer yaitu lapisan dense berukuran 256 dengan fungsi aktivasi ReLU. Untuk mengurangi kemungkinan terjadinya overfitting, peneliti memanfaatkan dropout layer sebesar 0.5 atau 50% sehingga dapat menghilangkan neuron secara acak (random) ketika model memasuki proses pelatihan. Layer dense kedua sekaligus menjadi layer output

berisi nilai 5 dan menggunakan fungsi aktivasi softmax yang digunakan untuk proses klasifikasi.

3.2.2. Arsitektur CNN dengan Xception

Model ketiga yang dibuat oleh peneliti yaitu model CNN yang mengimplementasikan pre-trained model Xception. Summary custom model dapat dilihat pada Tabel 2.

Tabel 2. Summary Custom Model

model: "sequential 1"			
Layer (type)	Output shape	Param#	Connected to
xception (Functional)	(None, 7, 7, 2048)	20861480	Categorical crossentropy
global_average_pooling2d (GlobalAveragePooling2D)	(None, 2048)	0	Categorical crossentropy
dense_2 (Dense)	(None, 256)	524544	Categorical crossentropy
dropout_1 (Dropout)	(None, 256)	0	Categorical crossentropy
dense_3 (Dense)	(None, 5)	1285	
Total params: 21,387,309			
Trainable params: 525,829			
Non-trainable params: 20,861,480			

Pre-trained model Xception yang digunakan pada penelitian ini berasal dari model yang sudah dilatih dengan parameter weight yang diisi dengan imagenet. Sebelum melakukan proses training, terdapat modifikasi pada model dasar Xception model yaitu dengan membuat dan menambahkan kepala model pada model dasar Xception model. Modifikasi tepatnya dilakukan pada *Exit Flow Xception Model* bagian output.

Dalam mengubah output layer yang dihasilkan oleh pre-trained model, custom layer yang dibangun sama dengan custom layer yang ada pada model VGG-16 yaitu memanfaatkan layer *Global Average Pooling* setelah layer pooling terakhir, lapisan dense berukuran 256 dengan fungsi aktivasi ReLu, dropout layer sebesar 0.5 atau 50%, dan layer dense kedua sekaligus menjadi layer output berisi nilai 5 dan menggunakan fungsi aktivasi softmax yang digunakan untuk proses klasifikasi.

3.3. Hasil Pelatihan Model

Setelah membuat model, selanjutnya dilakukan proses kompilasi dan fitting untuk melakukan proses pelatihan menggunakan data training dan data validation.

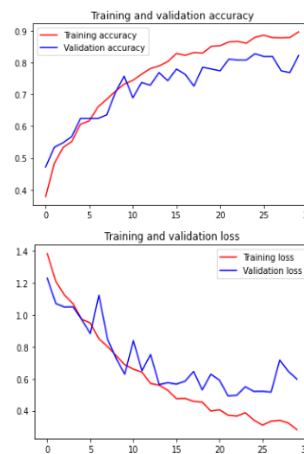
3.3.1 Hasil Pelatihan Model CNN Tanpa Transfer Learning

Proses kompilasi yang dilakukan pada arsitektur CNN menggunakan beberapa parameter yaitu optimizer, loss, dan metrics. Parameter optimisasi yang digunakan adalah Adam dengan learning rate 0,0004. Parameter loss digunakan untuk menentukan loss function. *Loss categorical_crossentropy*

digunakan untuk masalah klasifikasi multi-kelas. Parameter metrics digunakan untuk menentukan performa *metrics*, dalam penelitian ini metrics yang digunakan adalah *accuracy*.

Selanjutnya melakukan proses fitting yaitu data training dimasukkan ke dalam model dengan menggunakan fungsi model fit. Dalam melakukan proses fitting, epochs yang digunakan sebanyak 30 dan steps per epoch 707. Dalam proses fitting juga digunakan train generator, validation data, validation steps, dan verbose. Train generator merupakan data generator yang telah diproses untuk keperluan pelatihan menggunakan data training. Validation data hampir sama penggunaannya dengan training generator hanya saja penggunaannya untuk data validation. Validation steps sama seperti steps per epoch hanya saja digunakan untuk data validation. Verbose digunakan untuk menampilkan progress bar hasil dari setiap steps yang dilakukan.

Setelah melakukan proses fitting, didapatkan hasil pelatihan sebagai berikut:



Gambar 5. Grafik Hasil Pelatihan Model M0

Gambar 5. merupakan hasil dari pelatihan data training dan data validation menggunakan epoch (iterasi) sebanyak 30 dan learning rate 0.0004 dengan waktu pelatihan selama 1 jam 16 menit 20 detik. Akurasi dari data training pada epoch ke-30 mencapai 0.8964 dengan nilai loss sebesar 0.2816. Sedangkan akurasi untuk data validation pada epoch ke-30 mencapai 0.8220 dan nilai loss sebesar 0.5973. Garis berwarna merah menunjukkan grafik pergerakan akurasi dan loss untuk data training, sedangkan garis berwarna biru menunjukkan nilai pergerakan akurasi dan loss untuk data validation. Gambar diatas menunjukkan nilai akurasi dan loss dari kedua data yaitu data training dan data validation sama-sama menghasilkan nilai yang sudah cukup baik namun kurang stabil. Hal tersebut dapat dipengaruhi oleh jumlah data yang tersedia masih tergolong sedikit dan distribusi dataset tiap kelas yang kurang merata sehingga dapat mempengaruhi nilai akurasi dan waktu komputasi. Oleh karena itu, peneliti memanfaatkan metode Transfer Learning.

3.3.2. Hasil Pelatihan Model CNN dengan Transfer Learning

Proses pelatihan pada model CNN dengan Transfer Learning menggunakan parameter dan nilai yang sama seperti sebelumnya saat melakukan pelatihan pada model CNN. Namun pada model ini, untuk memperoleh model terbaik dan mengurangi kemungkinan overfitting, peneliti juga melakukan percobaan dengan nilai learning rate 0.0004 dan 0.0001, dropout layer sebesar 0.5 atau 50%, serta menerapkan fungsi *earlystopping* dan checkpoint pada proses training. Fungsi *earlystopping* digunakan untuk menghentikan pelatihan saat metrik val_loss berhenti meningkat. dan fungsi checkpoint digunakan untuk menyimpan model terbaik yang didapat selama pelatihan.

Percobaan pertama adalah model M1. Model M1 yaitu model CNN + VGG-16 dengan nilai learning rate 0.0004. Setelah melakukan proses fitting, didapatkan hasil pelatihan seperti pada Tabel 3. berikut:

Tabel 3. Hasil Fitting Model M1

Epic h	Data Training		Data Validation	
	Acc	Loss	Val Acc	Val Loss
1	0.6040	1.0268	0.6412	0.8875
2	0.7351	0.7106	0.7119	0.7470
3	0.7606	0.6272	0.7486	0.6903
4	0.7900	0.5746	0.7599	0.6885
5	0.7900	0.5430	0.7712	0.6408
⋮	⋮	⋮	⋮	⋮
28	0.9042	0.2595	0.8390	0.5049
29	0.9038	0.2498	0.8305	0.5011
30	0.9084	0.2469	0.8220	0.4961

Tabel 3. merupakan hasil dari pelatihan data training dan data validation menggunakan epoch (iterasi) sebanyak 30 dan learning rate 0.0004 dengan waktu pelatihan selama 1 jam 18 menit 47 detik. Nilai akurasi untuk data train pada epoch pertama menghasilkan nilai sebesar 0.6040 dan untuk nilai loss yang didapatkan adalah 1.0268. Sedangkan untuk data validation, nilai akurasi yang dihasilkan pada epoch pertama adalah 0.6412 dan nilai loss yaitu sebesar 0.8875. Nilai akurasi hasil iterasi pada kedua data baik data training maupun data validation sama-sama naik hingga epoch terakhir. Sementara nilai loss untuk kedua data terus menurun hingga epoch ke-24 lalu naik kembali hingga epoch terakhir. Akurasi dari data training pada epoch ke-30 mencapai 0.9084 dengan nilai loss sebesar 0.2469. Sedangkan akurasi untuk data validation pada epoch ke-30 mencapai 0.8220 dan nilai loss sebesar 0.4961. Namun model terbaik bukan dihasilkan pada epoch terakhir yaitu epoch ke-30 melainkan pada epoch ke-24, karena nilai loss pada data validasi mengalami kenaikan dari 0.49843 menjadi 0.46290. Epoch ke-24 memiliki akurasi pada data training sebesar 0.8879 dengan nilai loss sebesar 0.2900. Sedangkan akurasi untuk data validation mencapai 0.8220 dengan nilai loss sebesar 0.4629.



Gambar 6. Grafik Hasil Pelatihan Model M1

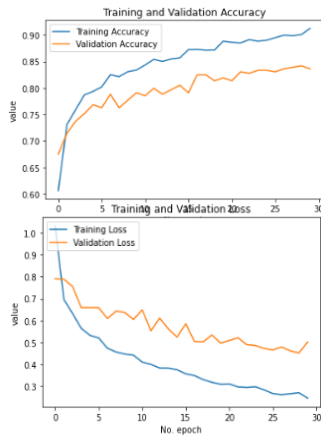
Gambar 6. merupakan hasil dari pelatihan data training dan data validation pada model M1. Nilai akurasi dan loss dari kedua data yaitu data training dan data validation sama-sama menghasilkan nilai yang sudah cukup baik dan lebih stabil dibandingkan pergerakan proses latih pada model M0.

Percobaan kedua pada model M2. Model M2 yaitu model CNN + VGG-16 dengan nilai learning rate 0.0001. Setelah melakukan proses fitting, didapatkan hasil pelatihan seperti pada Tabel 4. berikut:

Tabel 4. Hasil Fitting Model M2

Epic h	Data Training		Data Validation	
	Acc	Loss	Val Acc	Val Loss
1	0.6064	1.0319	0.6751	0.7915
2	0.7309	0.6961	0.7147	0.7887
3	0.7592	0.6318	0.7373	0.7551
4	0.7868	0.5637	0.7514	0.6585
5	0.7935	0.5320	0.7684	0.6586
⋮	⋮	⋮	⋮	⋮
28	0.8985	0.2650	0.8390	0.4612
29	0.9006	0.2695	0.8418	0.4516
30	0.9123	0.2448	0.8362	0.5012

Tabel 4. merupakan hasil dari pelatihan data training dan data validation menggunakan epoch (iterasi) sebanyak 30 dan learning rate 0.0001 dengan waktu pelatihan selama 55 menit 7 detik. Nilai akurasi untuk data train pada epoch pertama menghasilkan nilai sebesar 0.6064 dan untuk nilai loss yang didapatkan adalah 1.0319. Sedangkan untuk data validation, nilai akurasi yang dihasilkan pada epoch pertama adalah 0.6751 dan nilai loss yaitu sebesar 0.7915. Nilai akurasi hasil iterasi pada kedua data baik data training maupun data validation sama-sama naik hingga epoch terakhir. Sementara nilai loss untuk kedua data terus menurun hingga epoch ke-29 lalu naik kembali pada epoch terakhir. Akurasi dari data training pada epoch ke-30 mencapai 0.9123 dengan nilai loss sebesar 0.2448. Sedangkan akurasi untuk data validation pada epoch ke-30 mencapai 0.8362 dan nilai loss sebesar 0.5012. Namun model terbaik bukan dihasilkan pada epoch terakhir yaitu epoch ke-30 melainkan pada epoch ke-29, karena nilai loss pada data validasi mengalami kenaikan dari 0.46117 menjadi 0.45165.



Gambar 7. Grafik Hasil Pelatihan Model M2

Gambar 7. merupakan hasil dari pelatihan data training dan data validation pada model M2. Nilai akurasi dan loss dari kedua data yaitu data training dan data validation sama-sama menghasilkan nilai yang sudah cukup baik dan lebih stabil dibandingkan pergerakan proses latih pada model M1.

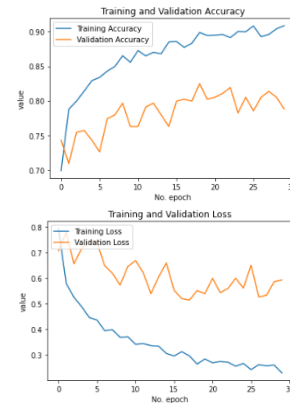
Percobaan ketiga pada model M3. Model M3 yaitu model CNN + Xception dengan nilai learning rate 0.0004. Setelah melakukan proses fitting, didapatkan hasil pelatihan seperti pada Tabel 5. berikut:

Tabel 5. Hasil Fitting Model M3

Epoc h	Data Training		Data Validation	
	Acc	Loss	Val Acc	Val Loss
1	0.6987	0.7932	0.7429	0.7067
2	0.7878	0.5791	0.7090	0.7781
3	0.7995	0.5250	0.7542	0.6560
4	0.8140	0.4887	0.7571	0.7147
5	0.8292	0.4459	0.7429	0.7412
⋮	⋮	⋮	⋮	⋮
28	0.8989	0.2642	0.8136	0.5327
29	0.8946	0.2842	0.8051	0.5859
30	0.8950	0.2693	0.7881	0.5924

Tabel 5. merupakan hasil dari pelatihan data training dan data validation menggunakan epoch (iterasi) sebanyak 30 dan learning rate 0.0004 dengan waktu pelatihan selama 57 menit 47 detik. Nilai akurasi untuk data train pada epoch pertama menghasilkan nilai sebesar 0.6987 dan untuk nilai loss yang didapatkan adalah 0.7932. Sedangkan untuk data validation, nilai akurasi yang dihasilkan pada epoch pertama adalah 0.7429 dan nilai loss yaitu sebesar 0.7067. Nilai akurasi hasil iterasi pada kedua data baik data training maupun data validation sama-sama naik hingga epoch terakhir. Sementara nilai loss untuk kedua data terus menurun hingga epoch terakhir. Akurasi dari data training pada epoch ke-30 mencapai 0.9088 dengan nilai loss sebesar 0.2299. Sedangkan akurasi untuk data validation pada epoch ke-30 mencapai 0.7881 dan nilai loss sebesar 0.5924. Namun model terbaik bukan dihasilkan pada epoch terakhir yaitu epoch ke-30 melainkan pada epoch ke-18, karena nilai loss pada data validasi mengalami kenaikan dari 0.51973 menjadi 0.51403. Akurasi dari

data training pada epoch ke-18 mencapai 0.8833 dengan nilai loss sebesar 0.2962.



Gambar 8. Grafik Hasil Pelatihan Model M3

Gambar 8. merupakan hasil dari pelatihan data training dan data validation pada model M3. Nilai akurasi dan loss dari kedua data yaitu data training dan data validation lebih tidak stabil dibandingkan pergerakan proses latih pada model M0, M1, dan M2.

Percobaan ketiga pada model M4. Model M4 yaitu model CNN + Xception dengan nilai learning rate 0.0001. Setelah melakukan proses fitting, didapatkan hasil pelatihan seperti pada Tabel 6. berikut:

Tabel 6. Hasil Fitting Model M4

Epoc h	Data Training		Data Validation	
	Acc	Loss	Val Acc	Val Loss
1	0.6648	0.8889	0.7316	0.7397
2	0.7687	0.6065	0.7006	0.7493
3	0.8002	0.5286	0.7260	0.7441
4	0.8200	0.4781	0.7627	0.6699
5	0.8342	0.4601	0.7429	0.7530
⋮	⋮	⋮	⋮	⋮
28	0.9105	0.2388	0.7966	0.5246
29	0.9176	0.2282	0.7938	0.5426
30	0.9176	0.2314	0.8249	0.5050

Tabel 6. merupakan hasil dari pelatihan data training dan data validation menggunakan epoch (iterasi) sebanyak 30 dan learning rate 0.0001 dengan waktu pelatihan selama 1h 5m 23s. Nilai akurasi untuk data train pada epoch pertama menghasilkan nilai sebesar 0.6064 dan untuk nilai loss yang didapatkan adalah 1.0319. Sedangkan untuk data validation, nilai akurasi yang dihasilkan pada epoch pertama adalah 0.6751 dan nilai loss yaitu sebesar 0.7915. Nilai akurasi hasil iterasi pada kedua data baik data training maupun data validation sama-sama naik hingga epoch terakhir. Sementara nilai loss untuk kedua data terus menurun hingga epoch ke-29 lalu naik kembali pada epoch terakhir. Akurasi dari data training pada epoch ke-30 mencapai 0.9123 dengan nilai loss sebesar 0.2448. Sedangkan akurasi untuk data validation pada epoch ke-30 mencapai 0.8362 dan nilai loss sebesar 0.5012. Namun model terbaik bukan dihasilkan pada epoch terakhir yaitu epoch ke-30 melainkan pada epoch ke-29, karena

nilai loss pada data validasi mengalami kenaikan dari 0.46117 menjadi 0.45165.

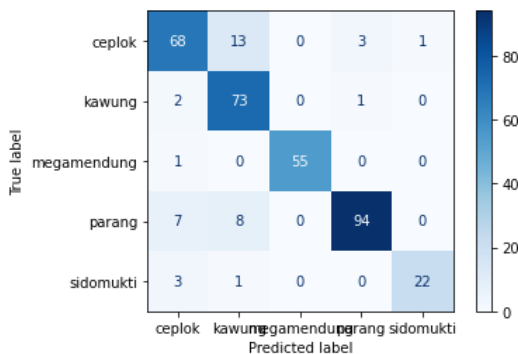


Gambar 9. Grafik Hasil Pelatihan Model M4

Gambar 9. merupakan hasil dari pelatihan data training dan data validation pada model M4. Nilai akurasi dan loss dari kedua data yaitu data training dan data validation sama-sama menghasilkan nilai yang sudah cukup baik dan lebih stabil dibandingkan pergerakan proses latih pada model M1.

3.4. Hasil Pengujian Model

Pada tahap ini dilakukan uji coba dan evaluasi model menggunakan data testing dimana data tersebut tidak termasuk kedalam pelatihan. Confusion matrix adalah salah satu metode pengukuran keputusan yang banyak digunakan dalam supervised machine learning. Informasi yang ditampilkan dalam confusion matrix merupakan angka-angka probabilistik [15]. Data testing terdiri dari 5 jenis motif batik, yaitu Ceplok, Kawung, Megamendung, Parang, dan Sidomukti. Data uji yang digunakan sebanyak 85 citra Ceplok, 76 citra Kawung, 56 citra Megamendung, 109 citra Parang, dan 26 citra Sidomukti. Berikut ini adalah hasil confusion matrix:



Gambar 10. Hasil Confusion Matrix Model Terbaik (M2)

Berdasarkan Gambar 10. hasil evaluasi dengan Confusion Matrix didapatkan akurasi 89%, precision 90%, recall 89%, dan F1-score 89%.

Pada Tabel 7. adalah perbedaan hasil antara penelitian penulis dengan penelitian-penelitian terdahulu:

Tabel 7. Perbandingan Akurasi Klasifikasi Motif Batik

Dataset	References	Feature Extraction	Acc
5 kelas & 900 citra	Agastya et al. [4]	CNN (VGG-19)	89.3%
5 kelas & 150 citra	Irawan et al. [5]	GLCM	73.3%
15 kelas & 3000 citra	Bariyah et al. [8]	CNN	91.4%
3 kelas & 120 citra	Mawan et al. [9]	CNN	85.9%
13 kelas & 967 citra	Tristanto et al. [10]	CNN	56%
5 kelas & 2,092 citra	Gultom et al. [12]	CNN (VGG-16)	89%
5 kelas & 3.534 citra	Proposed Method	CNN (VGG-16, Xception)	91.76%

4. DISKUSI

Hasil akurasi dan waktu komputasi yang didapat pada penelitian ini antara model satu dengan yang lain hampir sama yaitu akurasi berada direntang 0.8833 sampai 0.9176. Kemudian penelitian ini juga menjawab saran dari penelitian yang dilakukan Yohanes Gultom untuk menggunakan model pra-pelatihan lain seperti Xception [12].

Sebelumnya, penulis sudah melakukan tinjauan pustaka terhadap beberapa penelitian terdahulu yang menjadi referensi bagi penulis dalam melakukan penelitian ini. Berikut adalah perbedaan hasil antara penelitian penulis dengan penelitian-penelitian tersebut :

Perbedaan dengan hasil referensi ke-1 [12], yaitu "Batik Classification Using Deep Convolutional Network Transfer Learning" oleh Yohanes Gultom, dkk. adalah pada jenis motif, akurasi dan model transfer learning. Pada penelitian tersebut, akurasi yang didapatkan ketika menggunakan VGG-16 89,7%, ketika menggunakan SIFT 88,10% dan SURF 88,8%. Sedangkan pada penelitian ini ketika menggunakan VGG-16 91,23%.

Perbedaan dengan hasil referensi ke-2, yaitu "Classification of Indonesian Batik Using Deep Learning Techniques and Data Augmentation" oleh I Made Artha Agastya adalah jenis motif, akurasi, skenario percobaan. Pada penelitian tersebut didapat hasil akurasi sebesar 90,90% [4].

Perbedaan dengan hasil referensi ke-3, yaitu "Pengaruh Dimensi Gambar Pada Klasifikasi Motif Batik Menggunakan Convolutional Neural Network" oleh Riski Mawan adalah dimensi gambar. Pada penelitian tersebut terdapat tiga dimensi gambar berbeda. Pada dimensi 64x64 diperoleh akurasi sebesar 92,85%, dimensi 128x128 diperoleh akurasi sebesar 85%, dan dimensi 256x256 diperoleh akurasi sebesar 80%. Rata-rata akurasi yang didapat yaitu 85%. Sedangkan pada penelitian ini, diperoleh akurasi sebesar 89,64% pada model CNN [9].

5. KESIMPULAN

Berdasarkan penelitian dan pengujian yang telah dilakukan, model dengan hasil terbaik dalam mengklasifikasikan lima motif batik menggunakan Convolutional Neural Network dan transfer learning dalam penelitian ini didapatkan pada Model M2 (CNN + VGG-16 dengan Learning Rate 0,0001) dengan Akurasi Latih 0.9123 dan Akurasi Uji 0,89 dalam waktu 55 menit 7 detik.

Selain itu dalam mengimplementasikan Transfer Learning pada CNN untuk klasifikasi batik, peneliti menggunakan metode fine-tuning yang dimana pada metode ini menggunakan pre-trained model (VGG-16 dan Xception) yang telah dilatih namun tidak dibekukan. Jadi meskipun model sudah dilatih, nantinya pre-trained model ini akan dilatih lagi untuk kebutuhan yang baru yaitu untuk klasifikasi motif batik dengan menambahkan custom layer (top layer) yang ditentukan sendiri oleh peneliti. Dampak dari penerapan transfer learning ini membuat proses pelatihan jauh lebih cepat karena model sudah sedikit cerdas. Adapun hasil dari penelitian ini selanjutnya akan diterapkan pada aplikasi berbasis *mobile* agar aplikasi lebih interaktif dan lebih terasa manfaatnya.

6. DAFTAR PUSTAKA

- [1] A. Minarno, Y. Azhar, F. Sumadi, and Y. Munarko, "A Robust Batik Image Classification using Multi Texton Co-Occurrence Descriptor and Support Vector Machine," in *The 3rd International Conference on Intelligent Autonomous Systems*, 2020, pp. 51–55.
- [2] T. Suliyati and D. Yuliati, "Pengembangan Motif Semarang Untuk Penguatan Identitas Budaya Semarang," *Jurnal Sejarah Citra Lekha*, vol. 4, no. 1, pp. 61–73, 2019.
- [3] A. A. Trixie, "Filosofi Motif Batik Sebagai Identitas Bangsa Indonesia," *folio*, vol. 1, no. 1, pp. 1–9, 2020.
- [4] I. M. A. Agastya and A. Setyanto, "Classification of Indonesian Batik Using Deep Learning Techniques and Data Augmentation," in *ICITISEE 2018: the 3rd 2018 International conferences on Information Technology, Information Systems and Electrical Engineering*, 2018, pp. 27–31.
- [5] C. Irawan, E. N. Ardyastiti, D. Rosal, I. M. Setiadi, E. H. Rachmawanto, and C. A. Sari, "A Survey: Effect of the Number of GLCM Features on Classification Accuracy of Lasem Batik Images using K-Nearest Neighbor," in *International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, 2018, pp. 33–38.
- [6] C. U. Khasanah, E. Utami, and S. Raharjo, "Implementation of Data Augmentation Using Convolutional Neural Network for Batik Classification," Oct. 2020. doi: 10.1109/CITSM50537.2020.9268890.
- [7] M. Mardiyah, "Implementasi Deep Learning Untuk Image Classification Menggunakan Algoritma Convolutional Neural Network (CNN)," Universitas Islam Indonesia, 2020.
- [8] T. Bariyah, M. A. Rasyidi, and N. Ngatini, "Convolutional Neural Network untuk Metode Klasifikasi Multi-Label pada Motif Batik," *Techno.Com*, vol. 20, no. 1, pp. 155–165, Feb. 2021, doi: 10.33633/tc.v20i1.4224.
- [9] R. Mawan and H. al Fatta, "Pengaruh Dimensi Gambar Pada Klasifikasi Motif Batik Menggunakan Convolutional Neural Network," *Jurnal Teknologi Informasi*, vol. 4, no. 2, pp. 218–223, 2020, [Online]. Available: <https://fasnina.com>,
- [10] J. Tristanto, J. Hendryli, and D. E. Herwindiati, "Classification of Batik Motifs Using Convolutional Neural Networks," 2018. [Online]. Available: <http://ssrn.com/link/ITES-2018.html>
- [11] Rismiyati and A. Luthfiarta, "VGG16 Transfer Learning Architecture for Salak Fruit Quality Classification," *Telematika: Jurnal Informatika dan Teknologi Informasi*, vol. 18, no. 1, pp. 37–48, 2021, doi: 10.31515/telematika.v18i1.4025.
- [12] Y. Gultom, A. M. Arymurthy, and R. J. Masikome, "Batik Classification using Deep Convolutional Network Transfer Learning," *Jurnal Ilmu Komputer dan Informasi*, vol. 11, no. 2, pp. 59–66, Jun. 2018, doi: 10.21609/jiki.v11i2.507.
- [13] D. Arsa and A. Susila, "VGG16 in Batik Classification based on Random Forest," in *Proceedings of 2019 International Conference on Information Management and Technology (ICIMTech)*, 2019, pp. 295–299.
- [14] R. Ribani and M. Marengoni, "A Survey of Transfer Learning for Convolutional Neural Networks," in *The 32nd Conference on Graphics, Patterns and Images Tutorials (SIBGRABI-T)*, Oct. 2019, pp. 47–57. doi: 10.1109/SIBGRABI-T.2019.00010.
- [15] B. Purnama, *Pengantar Machine Learning*. Bandung: Penerbit Informatika, 2019.

