

IMPLEMENTASI CONVOLUTIONAL NEURAL NETWORK (CNN) KLASIFIKASI MOTIF BATIK MENGGUNAKAN EFFICIENTNET-B1

Aryo Hardirega, Irsan Jaelani, Minarto

Teknik Informatika, Sekolah Tinggi Teknologi Wastukencana
Jalan Cikopak No.53, Mulyamekar, Kec. Babakancikao Kab. Puwakarta, Jawa Barat (41151)
ryozero11@mail.com

ABSTRAK

Penelitian ini bertujuan untuk memudahkan pengenalan motif batik di Indonesia. Pengolahan citra digital dalam klasifikasi gambar bertujuan membuat komputer dengan meniru kemampuan manusia dalam membedakan atau mengelompokkan gambar berdasarkan informasi yang terkandung didalamnya melalui *Convolutional Neural Network* (CNN), bagian dari *Deep Learning*, untuk mendeteksi dan mengenali objek pada citra digital. Metode penelitian menggunakan CRISP-DM dengan arsitektur *EfficientNet-B1*, yang menunjukkan akurasi *training* 0,89 dan *loss* 0,33, serta akurasi testing 0,98 dan *loss* 0,70. Faktor seperti jumlah *strides*, nilai *dropout*, dan pemilihan aktivasi sangat berpengaruh pada kinerja model. Model ini dapat diimplementasikan dalam aplikasi *mobile* berbasis *android*.

Kata kunci : *Convolutional Neural Network* (CNN), Klasifikasi, Motif, Batik, *EfficientNet-B1* dan *Android*

1. PENDAHULUAN

Indonesia merupakan negara kepulauan yang terdiri dari berbagai suku bangsa, budaya dan bahasa. Salah satu budaya yang melekat dari Indonesia adalah kain batik sehingga batik dapat ditemukan di setiap wilayah Indonesia. Pada tanggal 2 Oktober 2009, UNESCO mengakui batik Indonesia sebagai warisan budaya dunia dengan memasukkannya ke dalam daftar representatif sebagai warisan budaya non-benda warisan manusia (*Representative List of the Intangible Cultural Heritage of Humanity*). Pada hari yang sama, itu juga diperingati sebagai Hari Batik Nasional. [1] Semua warga negara harus bertanggung jawab untuk melestarikan batik sebagai warisan seni dan budaya. Agar batik tidak punah dan terus berkembang, semua orang di negeri ini harus bersatu untuk melindunginya. Dengan adanya penelitian ini, akan lebih mudah untuk mengenal motif batik Indonesia yang ada saat ini.

Di era digital saat ini, dunia telah mengalami perkembangan teknologi yang sangat cepat. Berbagai teknologi telah dikembangkan di berbagai bidang kehidupan untuk mempermudah pekerjaan manusia. Pengolahan gambar digital adalah salah satu teknologi yang sedang berkembang. Ini adalah bidang ilmu informatika yang berfokus pada proses mengubah gambar menjadi gambar lain. Klasifikasi gambar adalah salah satu dari banyak teknik baru yang dikembangkan untuk pengolahan citra. Tujuan dari klasifikasi gambar saat ini adalah untuk membuat komputer atau mesin yang dapat meniru kemampuan manusia untuk membedakan atau mengelompokkan gambar digital berdasarkan informasi yang terkandung di dalamnya.

Pembuatan sistem berbasis *mobile* untuk klasifikasi motif batik menggunakan algoritma *Convolutional Neural Network* (CNN) salah satu metode dalam pengklasifikasiannya adalah dengan mengumpulkan data gambar sebagai sampel dari beberapa motif batik yang ada di Indonesia. Dalam

metode pengajaran mendalam, *Convolutional Neural Network* (CNN) digunakan untuk mendeteksi dan mengenali objek pada gambar digital. CNN juga digunakan dalam proses ekstraksi ciri menggunakan konvolusi dan klasifikasi. Pada dasarnya, metode pengajaran mendalam menggunakan konsep jaringan syaraf manusia yang disebut *neural network* dengan lapis. Penelitian CNN terus berkembang, mencapai titik di mana pengenalan citra digital dengan akurasi yang menyaingi manusia dapat dilakukan pada data tertentu [2].

Memanfaatkan arsitektur *EfficientNet-B1* untuk melakukan pemodelan arsitektur baru yang menghasilkan kinerja yang sangat baik dengan memanfaatkan *depth*, *width*, dan resolusi *network*. *EfficientNet* terdiri dari beberapa model yang diambil dari jaringan dasar dan dikonfigurasi sesuai dengan penskalaannya. Struktur arsitektur *EfficientNet* terdiri dari *Floating Point Operations Per Second* (FLOPS), akurasi yang tinggi, dan kemampuan untuk meningkatkan efektivitas model melalui pengurangan parameter. [3]. Dalam prosesnya algoritma CNN akan melakukan *training* dan *testing* terhadap motif batik sehingga kumpulan model batik yang telah terklasifikasi.

2. TINJAUAN PUSTAKA.

Berdasarkan referensi dari penelitian sebelumnya, yaitu implementasi metode *Convolutional Neural Network* (CNN) dalam klasifikasi motif batik, penelitian tersebut menggunakan arsitektur *EfficientNet-B0* dan menghasilkan nilai akurasi sebesar 79,62%. Diharapkan penelitian selanjutnya menghasilkan hasil yang lebih baik lagi serta dapat membantu meningkatkan dan memiliki tingkat akurasi yang lebih tinggi.[4]

2.1. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) adalah jenis jaringan saraf yang dirancang untuk memproses data dengan topologi grid yang mirip. CNN menggunakan konvolusi daripada penerapan matriks umum, di mana paling tidak satu konvolusi ditemukan di setiap lapisan. [5].

2.2. Batik

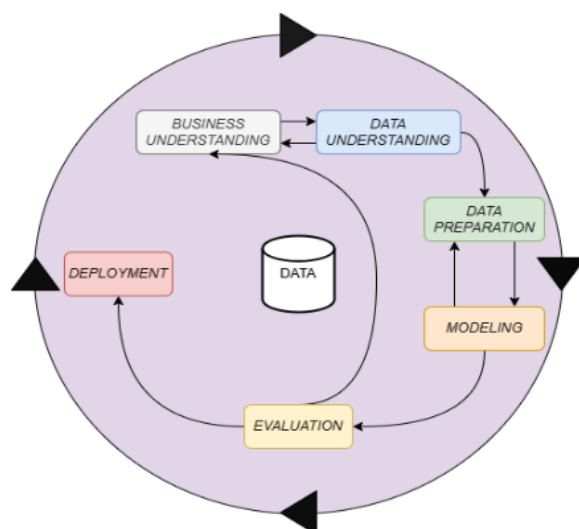
Indonesia merupakan negara kepulauan yang terdiri dari berbagai suku bangsa, budaya dan bahasa. Salah satu budaya yang melekat dari Indonesia adalah kain batik sehingga batik dapat ditemukan di setiap wilayah Indonesia. Pada tanggal 2 Oktober 2009, UNESCO mengakui batik Indonesia sebagai warisan budaya dunia dengan memasukkannya ke dalam daftar representatif sebagai warisan budaya non-benda warisan manusia (Representative List of the Intangible Cultural Heritage of Humanity). Pada hari yang sama, itu juga diperingati sebagai Hari Batik Nasional. [1].

2.3. EfficientNet-B1

EfficientNet-B1 adalah salah satu arsitektur dari keluarga EfficientNet yang dirancang untuk meningkatkan efisiensi dan kinerja model deep learning melalui teknik *compound scaling*. Arsitektur ini memperbesar dimensi kedalaman, lebar, dan resolusi jaringan secara seimbang untuk mencapai hasil yang optimal [6].

2.4. Cross Industry Standart For Data Mining (CRISP-DM)

Menurut [7] Dalam enam tahapan proses pengembangannya, CRISP-DM dianggap sebagai metodologi data mining terlengkap untuk memenuhi kebutuhan proyek industri. Tahapan tersebut mencakup, *Business Understanding*, *Data Understanding*, *Data Preparation*, *Modeling*, *Evaluation*, dan *Deployment*.



Gambar 1. Siklus CRIPS-DM

Berikut 6 tahapan CRISP-DM [8] :

- Business Understanding**
Tahap ini mencakup memahami kebutuhan dan tujuan dari sudut pandang bisnis, mendefinisikan masalah data mining, dan menetapkan rencana dan strategi untuk mencapai tujuan data mining..
- Data Understanding**
Pemahaman data mencakup persiapan, melakukan pengecekan terhadap data yang digunakan, mengumpulkan data awal, dan mengidentifikasi kualitas data. Dalam pemahaman data, data yang digunakan akan mengalami proses yang menjelaskan setiap fiturnya.
- Data Preparation**
Setelah data dikumpulkan, proses persiapan data melibatkan identifikasi, pemilihan, pembersihan, dan transformasi.
- Modelling**
Tahap implementasi algoritma yang akan digunakan untuk pencarian, identifikasi, dan pembuatan pola yang akan diterapkan pada data penelitian disebut modeling.
- Evaluation**
Evaluasi adalah suatu proses untuk mengukur hasil evaluasi dari model yang telah diimplementasikan pada tahap modeling sebelumnya. Hasil evaluasi tersebut menggambarkan proses data mining yang telah dilakukan dan menentukan model mana yang paling cocok untuk digunakan.
- Deployment**
Pembuatan laporan atau penerapan proses data mining

2.5. Android

Android adalah sistem operasi untuk perangkat *mobile* berbasis *Linux* yang menawarkan sistem operasi, aplikasi, *middleware*, dan *tool* pengembang yang lengkap, dengan dukungan komunitas *Open Source global* yang luas [9].

2.6. Machine Learning

Machine Learning adalah Mesin yang memiliki kemampuan untuk belajar secara mandiri tanpa bantuan manusia dikenal sebagai teknologi pembelajaran mesin. Berdasarkan pada disiplin ilmu lainnya seperti data mining, matematika, dan statistika, pembelajaran mesin memungkinkan mesin untuk belajar melalui analisis data tanpa perlu diprogram ulang atau diinstruksikan. Dalam hal ini, machine learning dapat memperoleh data dengan perintahnya sendiri dan juga dapat mempelajari data yang sudah ada sehingga dapat melakukan tugas tertentu. Tugas yang dapat dilakukan oleh machine learning sangat beragam tergantung pada apa yang dipelajari. [10].

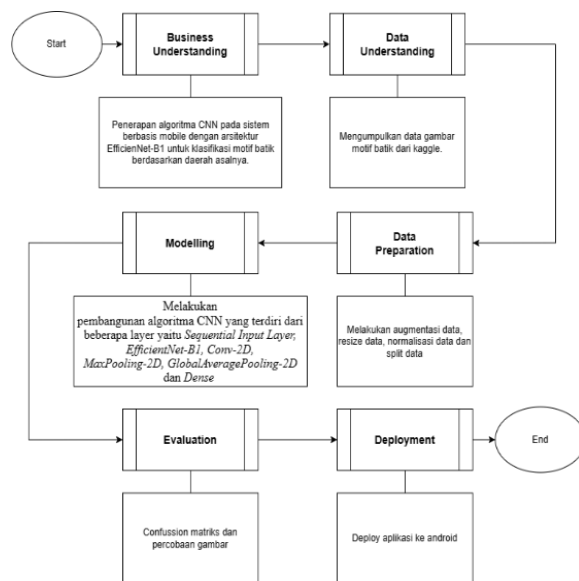
2.7. Machine Learning

Deep Learning merupakan metode *learning* yang menggunakan artificial neural network yang berlapis-lapis (multilayer). Artificial Neural Network

ini mirip dengan otak manusia karena jaringan neuronnya sangat rumit [11].

3. METODE PENELITIAN

Dalam kerangka penelitian ini, penulis menjelaskan segala sesuatu yang telah dan akan mereka lakukan selama penelitian, mulai dari *Business Understanding*, *Data Understanding*, *Data Preparation*, *Modelling*, *Evaluation* dan *Deployment* meng ikuti aturan metode CRISP-DM. Kerangka penelitian ini terdiri dari:



Gambar 2. Kerangka Penelitian

3.1. Business Understanding

Penelitian ini bertujuan untuk membuat sebuah aplikasi *mobile* yang dapat melakukan klasifikasi jenis motif batik dengan menggunakan metode *Convolutional Neural Network (CNN)* dan arsitektur *EfficientNet-B1*. Aplikasi akan melakukan pengenalan pada gambar motif batik dengan cara mengambil gambar menggunakan kamera handphone dan mengunggah gambar batik kemudian akan menghasilkan output klasifikasi motif batik dari gambar motif batik yang telah diambil atau di unggah.

3.2. Data Understanding

Data yang digunakan berupa gambar motif batik yang didapatkan dari website *Kaggle.com*. Dengan nama data *Indonesian Batik Motifs (Corak App)* diunggah oleh Ahmad Alfian pada tahun 2021 dari beberapa motif batik yang ada di Indonesia berjumlah sebanyak 300 gambar.

3.3. Data Preparation

Melakukan proses persiapan data meliputi augmentasi data, *resize* (mengubah ukuran gambar), menormalisasi data dan *split* data atau membagi menjadi dua kategori data yaitu data *training* untuk pelatihan model dan data testing untuk menguji model yang telah dibangun. Dengan tujuan mempersiapkan

proses ini adalah untuk meningkatkan akurasi model dan data mentah yang telah dikumpulkan sebelumnya.

3.4. Modelling

Pada tahap ini, melakukan pembangunan model *Convolutional Neural Network (CNN)* yang terdiri dari beberapa *layer* yaitu *Sequential Input Layer*, *EfficientNet-B1*, *Conv-2D*, *MaxPooling-2D*, *GlobalAveragePooling-2D* dan *Dense* untuk mendukung kinerja model.

3.5. Evaluation

Pada tahap ini, kinerja model dievaluasi. Ini dilakukan dengan metode *Matriks Confussion* untuk mengetahui bagaimana model berfungsi pada seluruh dataset. Selain itu, percobaan dilakukan menggunakan gambar yang sudah disiapkan untuk mengetahui apakah model dapat memprediksi kualitas motif batik berdasarkan daerah asalnya dengan baik atau tidak.

3.6. Deployment

Pada tahap *deployment*, model yang berhasil melewati tahap evaluasi dan memiliki kinerja yang baik akan disimpan dalam format *tflite* dan kemudian diimpor ke dalam aplikasi berbasis *android*. Dengan membuat aplikasi berbasis *android* akan mempermudah penggunaan sistem klasifikasi motif batik.

4. HASIL DAN PEMBAHASAN

4.1. Data Understanding

Pada tahap awal, penulis melakukan pengunduhan dataset menggunakan API dari situs *Kaggle.com*. Dataset yang diunduh berisi data gambar motif batik dengan 15 kategori masing-masing berisi 20 sampel gambar yaitu Bali, Betawi, Ceplok, Cendrawasih, Celup, Ciamis, Dayak, Gautan, Parang, Pekalongan, Priangan, Sekar jagad, Kawung, Mega mendung, dan Tambal berjumlah sebanyak 300 gambar. Proses pengunduhan dataset dapat dilihat pada gambar 3 dibawah ini.

```

!kaggle datasets download -d alfame/indonesian-batik-motifs-corak-app
Warning: Your Kaggle API key is readable by other users on this system! To fix this, you can run 'chmod 600 /root/.kaggle/kaggle.json'
Dataset URL: https://www.kaggle.com/datasets/alfame/indonesian-batik-motifs-corak-app
License(s): CC0-1.0
Downloading indonesian-batik-motifs-corak-app.zip to /content
94% 147M/156M [00:01<00:00, 97.8MB/s]
100% 156M/156M [00:01<00:00, 86.3MB/s]
  
```

Gambar 3. Baris Kode Pengunduhan Dataset

Melalui baris kode diatas, penulis mendapatkan file dataset yang masih menggunakan ekstensi ZIP. Untuk dapat menggunakan file dataset secara utuh dan keseluruhan maka perlu melakukan proses ekstraksi dataset yang masih berekstensi ZIP. Proses ekstraksi dataset dapat dilihat pada gambar 4 dibawah ini.

```

[5] import zipfile
zip_ref = zipfile.ZipFile('/content/indonesian-batik-motifs-corak-app.zip', 'r')
zip_ref.extractall('/content')
zip_ref.close()
  
```

Gambar 4. Proses Ekstraksi Dataset

Setelah dataset telah berhasil diekstraksi menggunakan baris kode diatas, maka selanjutnya penulis akan melakukan perapihan dataset pada tahapan *data preparation*.

4.2. Data Preparation

Setelah berhasil mendapatkan dataset yang diperlukan pada tahapan *data understanding*. Selanjutnya penulis akan melakukan perapihan dataset pada tahapan ini. Tujuan dari perapihan dataset adalah agar model yang akan dibangun nanti dapat memproses dataset dengan lebih optimal dan konsisten. Tahapan dalam melakukan perapihan dataset meliputi *resize data* dan *split dataset*.

4.2.1. Resize Data

Di awal proses perapihan dataset, langkah awal yang akan dilakukan adalah menyesuaikan ukuran data. Menyesuaikan data diperlukan agar model dapat memproses data secara konsisten dan optimal. Proses menyesuaikan dataset dilakukan dengan membuat variabel *img_height* dan *img_width* yang menampung nilai 240, variabel ini pada tahapan selanjutnya akan digunakan sebagai parameter ukuran data.

```
[11] batch_size = 32
     img_height = 240
     img_width = 240
```

Gambar 5. Variabel *img height* dan *img width*

4.2.2. Split Dataset

Proses selanjutnya dari tahapan perapihan dataset adalah membagi dataset menjadi dua kategori yaitu *data training* dan *data testing*. Proses pembagian dataset ini akan menggunakan suatu fungsi dari *library Tensorflow*. Baris kode untuk proses pembagian dataset dapat dilihat pada gambar 6 dibawah ini.

```
data_training = tf.keras.utils.image_dataset_from_directory(
    base_dir,
    seed=123,
    image_size=(img_height, img_width),
    batch_size=batch_size)
```

➡ Found 1050 files belonging to 15 classes.

```
[37] data_test = tf.keras.utils.image_dataset_from_directory(
    test_dir,
    seed=123,
    image_size=(img_height, img_width),
    batch_size=batch_size)
```

→ Found 300 files belonging to 15 classes.

Gambar 6. Baris Kode Proses *Split* Dataset

Pada baris kode diatas, penulis menggunakan beberapa parameter dalam membagi dataset, berikut adalah penjelasan dari masing-masing parameter yang digunakan:

- a. *validation_split*, Parameter ini menentukan proporsi dari dataset yang akan digunakan untuk validasi. Dalam baris kode diatas, 20% dari data akan digunakan untuk validasi, sementara 80% akan digunakan untuk pelatihan.

- b. *Subset*, Mengidentifikasi subset dari dataset yang akan dibuat. Di sini, penulis akan membuat data untuk '*training*' dan '*tesing*'.
- c. *Seed*, *Seed* adalah nilai acak yang digunakan untuk memastikan bahwa pengacakan data berulang kali akan menghasilkan hasil yang sama.
- d. *image_size*, Menentukan ukuran gambar yang akan diubah. *img_height* dan *img_width* adalah variabel atau nilai yang sudah ditentukan sebelumnya yang menunjukkan dimensi yang diinginkan untuk gambar.
- e. *Batch_size*, Menentukan jumlah gambar yang akan dimuat dalam setiap batch data. *Batch size* adalah parameter yang penting dalam pelatihan model, karena mempengaruhi kecepatan konvergensi model dan penggunaan memori. Jumlah *batch* pada masing-masing dataset adalah 32.

4.3. Modeling

Setelah berhasil melakukan proses perapihan dataset sehingga mendapatkan dua variable *data_training* dan *data_validation*, tahapan selanjutnya yang akan dilakukan penulis adalah membangun model *convolutional neural network* menggunakan arsitektur *EfficientNet-B1*. Tahapan awal dalam membangun model adalah dengan membuat variable *class_names* yang berisi jumlah kelas pada dataset yang digunakan.

```
[30] class_names = data_training.class_names
      print(class_names)
```

```
['Batik Bali', 'Batik Betawi', 'Batik Cendrawasih', 'Batik Dayak', 'Batik Geblok Rengas', 'Batik Ikat Celup', 'Batik Jombang', 'Batik Kase',
```

Gambar 7. Baris kode membuat *variabel class names*

Tahapan selanjutnya adalah melakukan proses *import* arsitektur *EfficientNet-B1* dari library *tf.keras.application*. Proses *import* arsitektur *EfficientNet-B1* dari library *tf.keras.application* dapat dilihat pada gambar 8 dibawah.

```
EfficientNetB1 = tf.keras.applications.EfficientNetB1(input_shape=img_size, include_top=False, weights='imagenet')  
  
EfficientNetB1.trainable = False
```

```

➔ Downloading data from https://storage.googleapis.com/keras-applications/efficientnetb1\_notop.h5
27018416/27018416 [=====] - 0s 0us/step

```

Gambar 8. Baris Kode proses *import* arsitektur *EfficientNet-B1* dari *library* *tf.keras.applications*

Setelah melakukan *import* arsitektur *EfficientNet-B1*, tahapan selanjutnya adalah mengimplementasikan arsitektur tersebut kedalam suatu model CNN. Dalam membangun model CNN, penulis menggunakan beberapa layer untuk memproses gambar. *Layer-layer* tersebut adalah:

- a. *Data augmentation*, layer pertama pada model ini adalah *augmentasi* dimana pertama akan meng-*augmentasi* dataset. *Layer* ini digunakan untuk meningkatkan keragaman data pelatihan.

- b. *EfficientNet-B1*, layer kedua pada model adalah arsitektur *EfficientNet-B1* yang sebelumnya sudah di-import.
- c. *layers.Conv2D*, layer ketiga pada model adalah layer konvolusi. Ini adalah layer konvolusi dengan 32 filter, ukuran kernel 3x3, dan fungsi aktivasi ReLU. *Padding 'same'* berarti ukuran *output* sama dengan ukuran input. Lapisan ini digunakan untuk mengekstraksi fitur dari gambar *input*.
- d. *layers.MaxPooling2D()*, layer keempat pada model adalah layer *maxPooling*. Layer ini bertujuan untuk mengurangi dimensi spasial dari fitur yang diekstraksi oleh lapisan konvolusi dengan mengambil nilai maksimum dalam setiap wilayah *pooling* (biasanya 2x2). Layer ini bertujuan untuk membantu mengurangi *overfitting* dan mengurangi kompleksitas komputasi.
- e. *Dropout*, layer kelima pada model adalah layer *dropout*, dimana tujuan dari layer ini adalah untuk efisiensi jumlah neuron sebanyak 0,1 untuk menghindari adanya *overfitting* pada model.
- f. *GlobalAveragePooling2D()*, layer keenam pada model adalah layer *pooling* yang mereduksi setiap peta fitur menjadi satu angka dengan mengambil rata-rata dari seluruh nilai dalam peta tersebut.
- g. *Dense*, layer ketujuh pada model adalah layer *dense (fully connected)* dengan 128 *neuron* dan fungsi aktivasi ReLU. Lapisan ini digunakan untuk membuat keputusan berdasarkan fitur yang diekstraksi sebelumnya.
- h. *Dense*, layer kedelapan pada model adalah layer *dense* terakhir yang jumlah *neuron* nya sesuai dengan jumlah kelas pada dataset.

```
[ ] num_classes = len(class_names)

model_withaug = Sequential([
    data_augmentation,
    EfficientNetB1,
    layers.Conv2D(filters=32, padding='same', kernel_size=3, activation='relu', strides=1),
    layers.MaxPooling2D(),
    tf.keras.layers.Dropout(rate=0.1),
    layers.GlobalAveragePooling2D(),
    layers.Dense(128, activation='relu'),
    layers.Dense(num_classes)
])
```

Gambar 9. Baris Kode Pembangunan Model

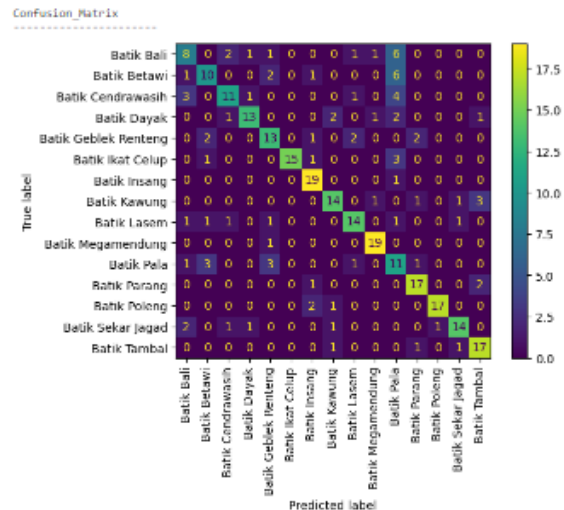
Setelah model berhasil dibangun, penulis selanjutnya akan melakukan pelatihan terhadap model. Pelatihan model dilakukan dengan metode *epoch*, metode *epoch* adalah suatu proses dimana model akan memproses seluruh data *training* yang sudah disiapkan dalam satu siklus yang terstruktur. Pada pelatihan model ini, penulis melakukan pelatihan dengan jumlah *epoch* sebanyak 10. Dibawah ini dapat dilihat hasil pelatihan model yang sudah dilakukan.

```
Epoch 1/10
10/10 [====] - 21s 6s/step - loss: 2.0252 - accuracy: 0.3524 - val_loss: 1.3007 - val_accuracy: 0.5567
Epoch 2/10
10/10 [====] - 179s 5s/step - loss: 1.1107 - accuracy: 0.6467 - val_loss: 1.0501 - val_accuracy: 0.6633
Epoch 3/10
10/10 [====] - 189s 6s/step - loss: 0.8438 - accuracy: 0.7343 - val_loss: 1.0899 - val_accuracy: 0.6780
Epoch 4/10
10/10 [====] - 186s 6s/step - loss: 0.6818 - accuracy: 0.7838 - val_loss: 0.9475 - val_accuracy: 0.7133
Epoch 5/10
10/10 [====] - 192s 6s/step - loss: 0.5745 - accuracy: 0.8162 - val_loss: 0.9911 - val_accuracy: 0.6880
Epoch 6/10
10/10 [====] - 192s 6s/step - loss: 0.5174 - accuracy: 0.8238 - val_loss: 1.0141 - val_accuracy: 0.7133
Epoch 7/10
10/10 [====] - 188s 6s/step - loss: 0.4548 - accuracy: 0.8535 - val_loss: 0.9792 - val_accuracy: 0.7180
Epoch 8/10
10/10 [====] - 188s 6s/step - loss: 0.3722 - accuracy: 0.8781 - val_loss: 1.0462 - val_accuracy: 0.7180
Epoch 9/10
10/10 [====] - 188s 6s/step - loss: 0.3147 - accuracy: 0.9095 - val_loss: 1.1383 - val_accuracy: 0.7000
Epoch 10/10
10/10 [====] - 187s 6s/step - loss: 0.3326 - accuracy: 0.8943 - val_loss: 0.9836 - val_accuracy: 0.7000
```

Gambar 10. Baris Kode Pelatihan Model

4.4. Evaluation

Tahapan selanjutnya adalah evaluasi dari model yang telah dibangun. Proses evaluasi dilakukan dengan menggunakan metode *confussion matrix*. *Confussion matrix* adalah alat yang digunakan untuk mengevaluasi kinerja model klasifikasi dengan memberikan gambaran yang jelas tentang prediksi yang benar dan salah untuk masing-masing kelas dalam dataset. *Confussion matrix* dari model 1 dapat dilihat pada gambar 11 dibawah ini.



Gambar 11. Confussion Matrix

Dari hasil *confussion matrix* diatas terlihat bahwa model memprediksi dengan benar jauh lebih banyak daripada memprediksi dengan salah pada setiap kelas. Sehingga dapat disimpulkan bahwa model yang dibangun dapat bekerja dengan baik atau dapat mengklasifikasikan jenis penyakit dengan baik.

4.5. Deployment

Setelah meng-evaluasi model dan model telah dinyatakan layak, tahapan selanjutnya adalah implementasi model kedalam suatu aplikasi *mobile* berbasis *Android*. Tujuan dari implementasi ini adalah agar model dapat digunakan oleh pengguna secara *portable* dan tanpa harus memerlukan spesifikasi komputasi yang tinggi. Dalam melakukan implementasi, model yang telah dibangun harus disimpan terlebih dahulu pada format *Tflite*. Dengan format ini, memungkinkan aplikasi untuk mengenali dan menjalankan model tanpa memerlukan spesifikasi yang tinggi. Proses menyimpan model pada format *Tflite* dapat dilihat pada gambar 12 dibawah ini.

```
[ ] from tensorflow.keras.models import save_model

save_model(model_withaug, 'batik_model.h5')
print("Model Saved")

!python -input-34-(7597284355): UserWarning: You are saving your model as an HDF5 file via "model.save()". This file format is save_model(model_withaug, 'batik_model.h5')
Model Saved

converter = tf.lite.TFLiteConverter.from_keras_model(model_withaug) # path to the SavedModel directory
tflite_model = converter.convert()

# Save the model.
with open('model_batik.tflite', 'wb') as f:
    f.write(tflite_model)
```

Gambar 12. Proses menyimpan model dalam format Tflite

Setelah model disimpan menjadi *TFLite*, langkah selanjutnya adalah mengimplementasikannya ke dalam aplikasi *Android*. Berikut ini adalah hasil dari implementasi model tersebut dalam sebuah aplikasi *Android*.

5. KESIMPULAN DAN SARAN

Berdasarkan penelitian, Model EfficientNet-B1 menunjukkan kinerja yang baik dalam mengklasifikasikan motif batik dengan akurasi training 0,89 dan loss training 0,33, serta akurasi testing 0,98 dan loss testing 0,70. Faktor seperti jumlah strides, nilai dropout, dan pemilihan aktivasi sangat berpengaruh pada tingkat akurasi dan loss model. Model ini juga dapat disimpan dan diimplementasikan dalam aplikasi mobile berbasis *Android* untuk memudahkan penggunaannya. Diharapkan penelitian ini membantu mengenali berbagai motif batik di Indonesia. Untuk penelitian selanjutnya, disarankan untuk meningkatkan jumlah data pelatihan dan kualitas gambar, menggunakan teknik augmentasi data, mengeksplorasi arsitektur model lain seperti MobileNet atau ResNet, serta mengintegrasikan model ini dengan sistem untuk edukasi pengenalan motif batik secara cepat dan akurat.

DAFTAR PUSTAKA

- [1] K. Azmi, S. Defit, and S. Sumijan, "Implementasi Convolutional Neural Network (CNN) Untuk Klasifikasi Batik Tanah Liat Sumatera Barat," *J. UNITEK*, vol. 16, no. 1, pp. 28–40, Jun. 2023, doi: 10.52072/unitek.v16i1.504.
- [2] L. Hakim, H. R. Rahmanto, S. P. Kristanto, and D. Yusuf, "KLASIFIKASI CITRA MOTIF BATIK BANYUWANGI MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK," *J. Teknoinfo*, vol. 17, no. 1, p. 203, Jan. 2023, doi: 10.33365/jti.v17i1.2342.
- [3] M. Khairunnas, "Penggunaan Metode Deep Learning Efficientnetb1 untuk Mengenali Sampah Organik dan Sampah Anorganik," p. 108, 2021, [Online]. Available: <https://repository.uin-suska.ac.id/57829/>
- [4] D. Anastasya, S. Fahri, and S. Situmorang, "Implementasi Metode Convolutional Neural Network (CNN) Dalam Klasifikasi Motif Batik," *NUANSA Inform.*, vol. 18, no. 1, pp. 1–5, Jan. 2024, doi: 10.25134/ilkom.v18i1.21.
- [5] A. E. Minarno, "Klasifikasi Jenis Batik Menggunakan Algoritma Convolutional Neural Network," *J. Repos.*, vol. 3, no. 2, pp. 278–284, 2021, doi: 10.22219/repositor.v3i2.1201.
- [6] A. Unique, "濟無No Title No Title No Title," no. 0, pp. 1–23, 2016.
- [7] D. Kurniawan and M. Yasir, "Optimization Sentimen Analysis using CRISP-DM and Naive Bayes Methods Implemented on Social Media," *Cybersp. J. Pendidik. Teknol. Inf.*, vol. 6, no. 2, p. 74, Oct. 2022, doi: 10.22373/cj.v6i2.12793.
- [8] M. A. Hasanah, S. Soim, and A. S. Handayani, "Implementasi CRISP-DM Model Menggunakan Metode Decision Tree dengan Algoritma CART untuk Prediksi Curah Hujan Berpotensi Banjir," *J. Appl. Informatics Comput.*, vol. 5, no. 2, pp. 103–108, 2021, doi: 10.30871/jaic.v5i2.3200.
- [9] A. Ramadhani, "PERANCANGAN DAN PEMBUATAN APLIKASI NOTES BERBASIS ANDROID (DESIGN AND DEVELOPMENT OF ANDROID-BASED NOTES APPLICATION)," Jul. 2021.
- [10] L. Utari and A. Zulfikar, "Penerapan Convolutional Neural Networks Menggunakan Edge Detection Untuk Identifikasi Motif Jenis Batik," *TeknoIS J. Ilm. Teknol. Inf. dan Sains*, vol. 13, no. 1, pp. 110–123, 2023, doi: 10.36350/jbs.v13i1.184.
- [11] P. A. Nugroho, I. Fenriana, and R. Arijanto, "Implementasi Deep Learning Menggunakan Convolutional Neural Network (CNN) Pada Ekspresi Manusia," *Algor*, vol. 2, no. 1, pp. 12–21, 2020.