



## Penerapan Algoritma Convolutional Neural Network Untuk Klasifikasi Motif Citra Batik Solo

Tungki Ari Bowo<sup>1</sup>, Hadi Syaputra\*<sup>2</sup>, Muhammad Akbar<sup>3</sup>

<sup>1,2,3</sup>Informatics Departement , Bina Darma University, Palembang, Indonesia

Email: <sup>1</sup>aribowo.ab7@gmail.com, <sup>2</sup>hadisyaputra@binadarma.ac.id,

<sup>3</sup>muhammad.akbar@binadarma.ac.id

### Abstrak

Batik adalah warisan kebudayaan Indonesia yang telah menjadi ciri khas masyarakat Indonesia. Batik memiliki banyak motif dan corak dengan ciri yang berbeda disetiap corak atau motifnya. Sebagai upaya melestarikan batik, penelitian mengenai klasifikasi batik digunakan untuk mengenali motif citra batik. Untuk itu penelitian ini dilakukan menggunakan metode pembelajaran dengan data citra batik diharapkan dapat memberikan hasil klasifikasi yang akurat dan efisien dengan tingkat akurasi tinggi. Convolutional Neural Network (CNN) merupakan gabungan dari jaringan syaraf tiruan dan metode deep learning. CNN terdiri 3 layer utama yaitu Convolutional Layer, Pooling Layer, Fully Connected Layer. Pada penelitian ini menggunakan arsitektur 3 Convolutional Neural Network dan 2 Fully Connected Layer pada pembuatan system terdapat beberapa tahapan utama yaitu pengumpulan data, perancangan system, training, dan testing. Penelitian ini menggunakan data citra batik solo sebanyak 2.256 gambar yang terdiri dari 7 kelas yaitu motif Parang, Semenrante, Sidomukti, Ceplok, Kawung, Truntum, dan Buketan. Hasil dari proses learning didapatkan akurasi sebesar 99.07% dengan loss sebesar 0,2%. Pada proses pengujian menggunakan 745 sampel batik solo didapatkan akurasi sebesar 95%. Sehingga dapat disimpulkan bahwa metode CNN dapat mengklasifikasi citra dengan baik.

**Kata Kunci:** batik, batik solo, convolutional neural network, image, deep learning.

### 1. PENDAHULUAN

Batik adalah warisan kebudayaan Indonesia yang telah menjadi ciri khas masyarakat Indonesia. Kain batik banyak digunakan untuk pakaian resmi maupun sehari-hari. Batik sendiri dianggap sebagai pakaian tradisional yang digunakan oleh masyarakat Indonesia. Pada dasarnya seni batik adalah seni melukis lilin atau cara menghias kain dengan menutupi bagian kain dengan lapisan lilin kemudian mencelupkan kain (Lestari, 2012). Oleh karena itu batik memang sangat istimewa,



karena pada setiap motif mempunyai makna filosofi yang memiliki nilai sejarah yang panjang dan corak maupun motif batik tersebut tidak lepas dari wilayah asal daerahnya.

Untuk melestarikan warisan budaya Indonesia ini, diperlukan penelitian yang berkesinambungan. Salah satu topik penelitian yang banyak dikembangkan dalam suatu bidang ilmu komputer yaitu klasifikasi batik. Topik penelitian ini juga relevan dengan topik yang masih aktual yaitu ekstraksi fitur digunakan untuk mewakili karakteristik suatu gambar batik sehingga dapat mengidentifikasi gambar dengan akurat. Akan tetapi, kain batik dengan motif yang bervariasi akan menyebabkan kesulitan dalam mengidentifikasi gambar batik itu sendiri. Sehingga diperlukan suatu metode yang dapat digunakan untuk mengekstraksi fitur batik berdasarkan motifnya (Putri & Rochmawati, 2019).

Pengetahuan tentang pengenalan jenis motif batik hanya dimiliki oleh orang-orang tertentu hal ini dikarenakan batik memiliki motif yang bervariasi dan hampir serupa di setiap daerahnya. Berdasarkan permasalahan tersebut salah satu cara untuk mengenali motif batik adalah dengan pengenalan pola. Metode tersebut sangatlah bisa diterapkan pada pengklasifikasian motif batik.

Dengan demikian, salah satu tantangan terpenting pada computer vision adalah pengenalan objek dimana komputer diberi gambar untuk dianalisa dan menerapkan algoritma pengenalan tertentu, tujuan utamanya adalah untuk mendeteksi objek didalam gambar. Salah satu metode yang dapat digunakan untuk pengenalan pola adalah menggunakan metode Deep Learning yaitu Convolutional Neural Network (CNN). Convolutional Neural Network (CNN) telah berhasil melakukan pengklasifikasian dalam membedakan salak yang lolos ekspor dan tidak. Input yang digunakan dalam penelitian tersebut adalah citra salak (RISMIYATI & M.t, 2016).

Selaras dengan itu penelitian ini mencoba merancang sebuah sistem pengklasifikasian motif citra batik solo yang menggunakan algoritma Convolutional Neural Network (CNN) sebagai metode pengenalan pola khususnya klasifikasi gambar batik. Oleh karena itu penulis bertujuan memperkaya riset mengenai klasifikasi gambar citra batik solo menggunakan metode CNN. Diharapkan Convolutional Neural Network (CNN) dapat mengenali serta mengklasifikasi batik solo berdasarkan tujuh kelas motif batik yaitu, motif Parang, Buketan, Ceplok, Kawung, Truntum, Semen Rante,

dan Sidomukti yang diharapkan dapat memberikan hasil yang lebih efisien dan akurat dengan akurasi yang cukup baik dan berguna bagi pihak yang membutuhkan.

## 2. METODOLOGI PENELITIAN

### 2.1 Metode pengumpulan data

#### 2.1.1 Populasi dan sampel

Populasi merupakan suatu wilayah generalisasi yang terdiri atas obyek atau subyek yang mempunyai kualitas dan karakteristik tertentu yang telah ditetapkan oleh peneliti untuk dipelajari dan kemudian ditarik kesimpulannya (Sugiyono, 2012). Populasi dalam penelitian ini adalah motif citra batik yaitu semua data motif citra batik yang ada di Indonesia. Sedangkan untuk sampel dalam penelitian ini adalah motif citra batik solo yang terdiri dari tujuh(7) jenis motif yaitu motif Parang, Buketan, Ceplok, Kawung, Truntum, Semen Rante, dan Sidomukt.

#### 2.1.2 Teknik Sampling

Data primer adalah data yang digunakan dalam penelitian ini, data tersebut dikumpulkan dari berbagai sumber dari internet. Pengumpulan data primer adalah dengan mengunduh data citra yang telah ada di internet. Dalam penentuan sampel adapun cara yang digunakan yaitu menggunakan cara *purposive sampling*. Hal ini dilakukan dengan cara mengambil subjek bukan didasarkan atas tujuan tertentu. Purposive sampling adalah teknik penentuan sampel dengan pertimbangan tertentu (Sugiyono, 2016).

### 2.2 Metode simulasi

Pada penelitian ini metode yang digunakan adalah metode simulasi. Berikut ini adalah langkah-langkah metode simulasi :

#### 2.2.1 Formulasi Masalah (*Problem Formulation*)

Di dalam tahapan ini, penulis melakukan identifikasi masalah berdasarkan penelitian-penelitian yang berkaitan dengan metode *Convolutional Neural Network* (CNN) dan *Image Classification* pada *Deep Learning*. Pada penelitian yang sudah ada belum ada yang membahas permasalahan sistem pengklasifikasian motif citra batik solo menggunakan metode (CNN)

## 2.2.2 Model Pengkonsepian (*Conceptual Model*)

Pada tahap ini, penulis merencanakan model konsep pada penelitian ini, pada penelitian ini akan melakukan pemodelan untuk mengenali pola motif batik solo pada citra gambar batik solo dengan arsitektur *Convolutional Neural Network* pada proses klasifikasinya. Proses pembuatan sistem nantinya akan terdapat proses ekstraksi fitur pada dataset gambar, mendapat nilai akurasi tinggi, membuat model sistem untuk mengenali motif batik, lalu dihubungkan dengan *interface* yang telah dibuat. Setelah itu mendapat *output* hasil sistem berupa motif citra yang telah di *input* dan dikenali oleh sistem.

## 2.2.3 Pengumpulan Data Masukan/Keluaran (*Collection input/Output Data*)

Dalam tahapan ini, penulis melakukan pengumpulan data yang dibutuhkan untuk membuat model sistem untuk pengklasifikasian motif batik ini. Data yang dikumpulkan berupa gambar batik solo berdasarkan 7 kelas yaitu motif batik Parang, Buketan, Ceplok, Kawung, Truntum, Semen Rante, dan Sidomukti. Data kemudian dikumpulkan dari berbagai sumber dari internet. Data awal berupa gambar batik. Kemudian penulis memperbanyak data dengan memotong, memutar rotasi dan dibolak balik menjadi motif yang diperlukan sesuai kelas yang telah di tentukan. Setelah seluruh gambar dikumpulkan dan dilakukan pemilihan dari gambar yang berkualitas baik, data gambar tersebut nantinya akan dibagi menjadi data latih dan data uji. Pada proses pembuatan model, untuk mencapai nilai akurasi yang diinginkan, setelah data dilatih dan diuji. Dan telah mendapat jumlah yang diinginkan kemudian dibuat model dengan metode *Convolutional Neural Network* (CNN).

## 2.2.4 Preprocessing

Pada tahap ini data yang sudah ada kemudian masuk ketahap preprocessing. Data yang masuk kemudian diresize, yang bertujuan untuk menyamakan ukuran gambar. Data yang sudah diresize kemudian melalui proses segmentasi citra dimana data diproses untuk menentukan deteksi tepi, yang bertujuan untuk mengenali objek yang ada dalam data.

## 2.2.5 Pemodelan

Dalam tahapan ini penulis melakukan pembuatan model dari dataset yang ada untuk mengenali gambar dan proses klasifikasi menggunakan metode *deep learning* yang menggunakan algoritma *Convolutional Neural network* agar sistem dapat mengenali citra motif batik.

## 2.2.6 Pelatihan Pengujian

pada tahap pelatihan dan pengujian yaitu menggunakan metode *Convolutional Neural Network*. Pada tahapan ini, data yang dikumpulkan dalam bentuk citra dibagi menjadi menjadi data latih dan data uji. Untuk mendapatkan model yang ingin dibuat dibutuhkan suatu analisis dari data latih. Setelah data dilatih, dilakukan tahap selanjutnya dengan menghitung nilai loss dan akurasi dari model-model yang dihasilkan dari data uji. Data uji berguna untuk mengetahui nilai akurasi yang dihasilkan dari model yang telah dilatih. Penentuan nilai dari data *Epoch* dimana suatu perulangan dari proses pengujian pada satu kali sesi lewatkan guna untuk mendapatkan error terkecil dan menghasilkan kinerja yang baik untuk model yang dibuat.

## 2.2.7 Eksperimental (*Eksperimentation*)

Selanjutnya penulis melakukan tahap eksperimentasi yaitu mencoba keberhasilan model yang dibuat. Penulis menginput dokumen batik yang berformat jpg yang telah disiapkan oleh penulis, lalu sistem akan memproses inputan tersebut. Setelah diproses melalui model yang dibuat, keluarlah hasil dari inputan tersebut berupa kelas yang telah ditentukan.

## 2.2.8 Analisis Output (*Output Analysis*)

Pada tahap ini penulis melakukan analisa terhadap output atau keluaran berdasarkan skenario yang dilakukan apakah sesuai dengan inputan. Selain itu juga pada tahap ini akan menampilkan hasil keluaran dengan inputan dokumen.

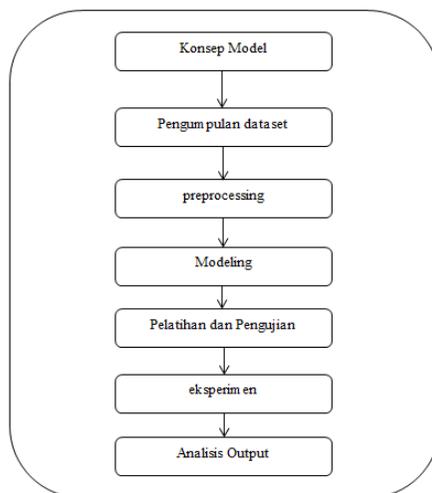
## 2.3 Metode Analisis data

dalam metode analisis data yang digunakan dalam penelitian ini adalah menggunakan metode *Deep Learning* yaitu *Convolutional Neural Network (CNN)*, metode ini digunakan untuk klasifikasi *objec recognition* pada motif batik solo.

## 2.4 Diagram Alur Penelitian

Seperti yang diperlihatkan pada gambar 1 alur pengerjaan penelitian ini dimulai dari konsep model yaitu menentukan konsep yang dibuat. Kemudian pengumpulan data sampel yang digunakan sebagai input untuk data training, validasi dan pengujian/tes. Kemudian merancang jaringan dengan metode

CNN untuk melakukan klasifikasi pada motif citra batik solo. Rancangan jaringan CNN diaplikasikan dengan data training. Sehingga komputer dapat belajar mengenali objek. Untuk mengetahui pembelajaran jaringan yang dihasilkan menghasilkan nilai yang sangat baik dalam membedakan jenis objek batik solo, maka jaringan tersebut kemudian di uji coba terhadap data validasi. Jika data validasi mendapatkan hasil yang baik maka jaringan dapat digunakan untuk klasifikasi pada data tes.



Gambar 1. Diagram Alur

## 3. RESULTS AND DISCUSSION

### 3.1 Pelabelan

Setelah dataset terkumpul dan dirapikan dalam folder sesuai kelas yang telah melewati tahap preprocessing, selanjutnya dataset diberi label sesuai kelas. Pelabelan dimulai dari label 1 sampai 7 label dengan dtype=float32 kemudian di *mapping* berdasarkan kelas. Sehingga menghasilkan label dan kelas sebagai berikut:

```
[ ] mapping = {  
    "batik truntum": "batik truntum",  
    "batik semen rante": "batik semenrante",  
    "batik kawung": "batik kawung",  
    "batik parang": "batik parang",  
    "batik ceplok": "batik ceplok",  
    "batik sidomukti": "batik sidomukti",  
    "batik buketan": "batik buketan"  
}
```

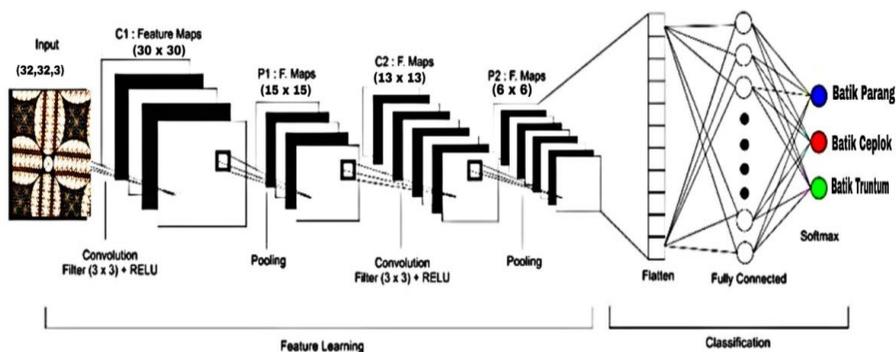
Gambar 2. Pelabelan

### 3.2 Membagi Dataset Untuk Dilatih dan Diuji dari Dataset

Proses selanjutnya adalah dataset yang telah ada akan dibagi menjadi data latih dan data uji menggunakan model sklearn yang sudah ada pada *library* python. pada penelitian ini digunakan *library* sklearn karena *library* ini memudahkan pemrograman karena lebih sedikit baris kode dibandingkan dengan *library* lainnya seperti tensorflow atau numpy. *Library* sklearn juga dapat digunakan untuk *clustering* atau pengelompokan data dengan karakteristik yang sama ke suatu wilayah yang sama dan data dengan karakteristik yang berbeda ke wilayah yang lain. Fungsi `train_test_split` di masukkan pada program guna memisahkan array atau matrik menjadi data latih dan data test secara acak. Pada proses ini digunakan `test_size` sebesar 0.33 dan `random size` sebanyak 42.

### 3.3 Arsitektur Jaringan

Setelah melalui tahap pelabelan dan pengambilan data training dari dataset, selanjutnya dataset akan dilatih menggunakan algoritma *Convolutional Neural Network* (CNN). Pada algoritma *Convolutional Neural Network* (CNN) pembentukan arsitektur dapat mempengaruhi hasil dan akurasi model.



Gambar 3. Arsitektur Jaringan

Pada gambar 3 adalah arsitektur jaringan pada proses training untuk menghasilkan model yang optimal. Penelitian ini menggunakan input berukuran  $32 \times 32 \times 3$ , yang bertujuan untuk membandingkan nilai akurasi berdasarkan ukuran gambarnya. Penjelasan untuk gambar arsitektur diatas dapat dijelaskan seperti seperti penjelasan dibawah ini:

- 1) Pada proses pertama yaitu proses konvolusi digunakan kernel berukuran  $3 \times 3$  dan jumlah dengan jumlah filter sebanyak 32 filter, proses konvolusi ini adalah

proses kombinasi antara dua buah matrik yang berbeda untuk menghasilkan suatu nilai matriks yang baru. Setelah proses konvolusi, maka ditambahkan sebuah aktivasi fungsi RELU (*Retrified Linier Unit*). Fungsi aktivasi ini bertujuan untuk mengubah nilai negatif menjadi nol (menghasilkan nilai negative dalam sebuah matriks hasil konvolusi). Hasil konvolusi ini memiliki ukuran baru yakni 30x30.

- 2) Proses *pooling* pertama. *Pooling* merupakan pengurangan ukuran matriks dengan menggunakan operasi *pooling*. Pada dasarnya *pooling layer* terdiri dari suatu filter dengan ukuran tertentu yang secara bergantian bergeser pada seluruh area *feature map*. Penelitian ini menggunakan *maxpooling* untuk mendapatkan nilai matriks yang baru hasil dari proses *pooling*. Berdasarkan hasil *pooling* menghasilkan matriks baru berukuran 15x15 dari input hasil konvolusi pertama yang berukuran 30x30. Proses ini menggunakan kernel *pooling* 2x2. Untuk cara kerja *max-pooling* adalah mengambil nilai paling maksimum berdasarkan pergerakan kernelnya sebanyak nilai *stride* nya yaitu 2.
- 3) Proses konvolusi kedua adalah meneruskan hasil dari proses *pooling* pertama yakni dengan input matriks dengan gambar sebesar 15x15 dengan jumlah filter yang sama dengan konvolusi pertama yakni sebanyak 32 filter dan dengan ukuran kernel 3x3. Proses konvolusi kedua sama-sama menggunakan fungsi RELU. Proses ini menghasilkan gambar berukuran 13x13.
- 4) Proses selanjutnya adalah proses *pooling* ke dua, proses ini hampir sama dengan proses *pooling* pertama akan tetapi memiliki perbedaan di nilai output akhir dan matriksnya. Output yang dihasilkan dari proses *pooling* kedua memiliki ukuran gambar sebesar 6x6
- 5) Proses selanjutnya adalah proses *Flatten* dan *fully connected*. Untuk tahap ini hanya menggunakan satu hidden layer pada jaringan MLP (*Multi Layer perceptron*). Fungsi *Flatten* disini berfungsi untuk mengubah *output pooling layer* menjadi sebuah vector. Sebelum melakukan klasifikasi atau memprediksi gambar, pada tahap ini digunakan nilai *dropout*. *Dropout* adalah sebuah teknik regulasi jaringan syaraf dengan tujuan untuk memilih beberapa neuron secara acak dan tidak akan dipakai selama proses pelatihan. Oleh sebab itu neuron-neuron dibuang secara acak. Tujuan dari penelitian ini adalah mengurangi *overfitting* pada proses *training*. Nilai *Droupout* yang digunakan sebesar 0,5 engan mengeluarkan hasil yang sama 6x6
- 6) Proses terakhir adalah menggunakan aktivasi fungsi *Softmax*. Fungsi ini secara spesifik bisa digunakan pada metode klasifikasi *multinomial logistic regression* dan *multiclass linier discriminant analiys*.

Pada penjelasan uraian arsitektur jaringan diatas, arsitektur tersebut digunakan untuk proses trining. Sehingga dari proses training didapatkan model dari arsitektur tersebut. Berikut model yang terbentuk :

```

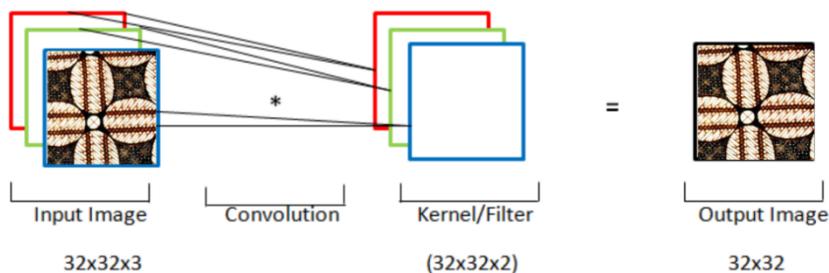
Model: "sequential_2"
-----
Layer (type)                Output Shape              Param #
-----
conv2d_3 (Conv2D)           (None, 30, 30, 32)       896
max_pooling2d_3 (MaxPooling2 (None, 15, 15, 32)       0
conv2d_4 (Conv2D)           (None, 13, 13, 32)       9248
max_pooling2d_4 (MaxPooling2 (None, 6, 6, 32)       0
dropout_3 (Dropout)         (None, 6, 6, 32)         0
flatten_2 (Flatten)         (None, 1152)              0
dense_3 (Dense)             (None, 128)               147584
dropout_4 (Dropout)         (None, 128)               0
dense_4 (Dense)             (None, 7)                 903
-----
Total params: 158,631
Trainable params: 158,631
Non-trainable params: 0
-----
None
    
```

Gambar 4. Model CNN

Gambar 4. adalah model yang terbentuk dari hasil *training*. Total parameter yang terbentuk dari model sebanyak 158,631 neuron

### 5.3.1 Proses *Convulusi Layer*

Proses selanjutnya adalah proses konvolusi yang terjadi berdasarkan arsitektur jaringan yang telah dibahas diatas.



Gambar 5. Proses Konvolusi

Konvolusi adalah proses penggabungan dua digit angka pada suatu gambar sehingga menghasilkan digit angka yang baru, dengan kata lain gambar tersebut dimanipulasi

yang menggunakan *external mask / subwindows* untuk menghasilkan citra yang baru, angka konvolusi pada citra tersebut berbentuk matriks *array*. Berdasarkan gambar 3.4 digunakan input gambar sebesar 32 pixel dan gambar input tersebut mempunyai *channel* yaitu *red, green, dan blue* atau sering disebut dengan RGB.

Disetiap *channel* pixel mempunyai nilai matriks yang berbeda-beda. Input akan di konvolusi dengan nilai filter yang sudah ditentukan. Filter adalah blok lain atau kubus engan tinggi dan lebar yang lebih kecil namun mempunyai kedalaman yang sama yang tersapu di atas gambar dasar atau gambar asli. Filter digunakan untuk menentukan pola apa yang akan dideteksi yang selanjutnya akan dikonvolusi atau dikalikan dengan nilai pada input matriks, nilai masing-masing kolom dan baris pada matriks sangat bergantung pada jenis pola yang akan dideteksi. Jumlah filter pada konvolusi ini sebanyak 32 dengan ukuran (3x3) yang artinya gambar dihasilkan dari hasil konvolusi akan sebanyak 32 fitur map.

Berdasarkan arsitektur jaringan diatas digunakan aktivasi ReLU (*Rectified Linear Unit*) setelah proses konvolusi berlangsung. Fungsi aktivasi sendiri adalah melakukan "threshold" dari 0 hingga infinity. Nilai yang ada pada hasil konvolusi yang bernilai negatif akan diubah dengan aktivasi ini menjadi nol dan yang lainnya sampai infinity.

### 3.3.2 Proses Pooling

*Pooling* atau penggabungan merupakan proses dari CNN dimana matriks input disederhanakan menjadi matriks yang baru. Dalam penelitian ini digunakan metode *max pooling* pada proses *pooling*nya dimana akan diambil nilai terbesar dari matriks *input* yang kemudian menghasilkan matriks baru sebagai *output*nya. Berdasarkan penelitian (Hutchison et al., 2010) berjudul "*Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition*" yang menunjukkan bahwa *max pooling* lebih efektif menghasilkan hasil yang lebih baik dibandingkan dengan metode lain seperti subsampling untuk data berupa gambar. Untuk itu pada penelitian ini menggunakan *max pooling* didalam proses *pooling*nya

### 3.3.3 Proses Fully Connected

Proses terakhir adalah *Fully-Connected*. Hasil dari proses terakhir dari model yang ada di dalam *feature extraction layer* yang masih berbentuk *multiimentional array*, sehingga harus dilakukan "*flatten*" atau *reshape feature map* menjadi sebuah vector

agar bisa digunakan sebagai *input* dari *fully-connected layer* yang kemudian diberi fungsi *dense*. *Dense* adalah fungsi untuk menambahkan *layer* pada *fully-connected*.

```
model.add(Dense(128, activation='relu')) #fully connected la
```

**Gambar 6.** Fully Connected Layer

Pada gambar diatas *dense* yang digunakan adalah sebanyak 128 unit. Unit tersebut merupakan jumlah node yang harus ada di *hidden layer* yang mempunyai nilai antara jumlah input node dan *output* node. Sedangkan untuk fungsi aktivasinya digunakan fungsi ReLU. Fungsi dari *fully Connected layer* ini adalah untuk melakukan transformasi pada dimensi data agar data dapat diklasifikasikan secara linear.

```
model.add(Dense(7, activation='softmax')) #softmax works
```

**Gambar 7** Fully Connected Layer Akhir

### 3.4 Akurasi

Setelah melewati beberapa tahap pemodelan, kemudian model tersebut diuji nilai akurasi dan lossnya dengan menggunakan adam optimizer

```
epochs = 100
model.compile(loss='categorical_crossentropy', optimizer='Adam', metrics=['accuracy'])
history = model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=epochs, batch_size=32)
#Final evaluation of the model
```

**Gambar 8.** Uji Akurasi

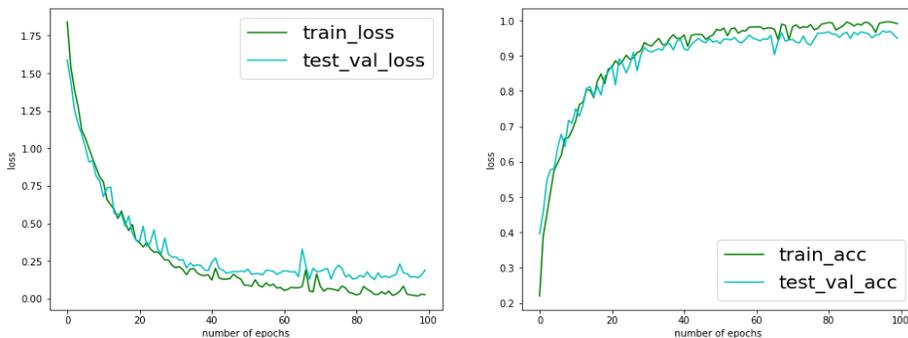
Dengan menggunakan iterasi epoch sebanyak 100 selajutnya hasil akurasi validasi yang didapatkan adalah sebagai berikut:

```
loss: 0.0259 - accuracy: 0.9907 - val_loss: 0.1895 - val_accuracy: 0.9490
Accuracy: 94.90%
```

**Gambar 9.** Hasil Akurasi

Berdasarkan gambar 5.8 yang menunjukkan hasil akurasi model. Sehingga dapat diketahui besar loss 0.2% dengan akurasi yang tinggi sebesar 99,07% sedangkan

untuk data validasi, nilai loss nya tidak terlalu tinggi yaitu sebesar 18,95% dan akurasi sebesar 94,90% yang menunjukkan akurasi model sebesar 94,90%. Untuk klasifikasi dengan menggunakan banyak kelas yaitu sebanyak 7 kelas nilai akurasi yang didapat sebesar 94,90% tersebut sudah bagus untuk dijalankan dan sistem sudah cukup pintar untuk mendeteksi batik sesuai dengan kelas yang ada. Dibawah ini merupakan paparan grafik gambar akurasi dan loss dari model *training* dan *validasi*:



Gambar 10. Grafik Training Akurasi dan Training Loss

Pada gambar sebelah kanan diatas menunjukkan perbedaan akurasi antara hasil training dan validasi tidak terlalu signifikan. Pada *epoch* ke 30 menunjukkan data akurasi untuk data train cukup stabil hingga pada epoch ke 100. Sedangkan untuk data validasi mulai *epoch* ke 40 hingga *epoch* ke 100, nilai akurasi yang dihasilkan cukup stabil. Untuk gambar sebelah kiri nilai kesalahan untuk data *training* dimulai pada kisaran 1,50 untuk data train dan 0,50 untuk data validasi dan keduanya mulai stabil pada *epoch* ke 30 sampai ke *epoch* 100.

### 3.5 Precision Recall

	precision	recall	f1-score	support
batik truntum	0.99	0.94	0.97	108
batik semenrante	0.89	0.99	0.94	121
batik kawung	0.97	0.76	0.86	97
batik parang	0.96	0.95	0.95	111
batik ceplok	0.95	0.99	0.97	95
batik sidomukti	0.98	1.00	0.99	97
batik buketan	0.93	0.99	0.96	116
accuracy			0.95	745
macro avg	0.95	0.95	0.95	745
weighted avg	0.95	0.95	0.95	745

Gambar 11. Precision Recall

Pada gambar diatas dapat kita lihat bahwa *precesion* (tingkat ketepatan) menghasilkan ketepatan yaitu untuk batik truntum sebesar 99%, batik semenrante 89%, batik kawung 97%, batik parang 96%, batik ceplok 95%, batik sidomukti 98%, dan batik buketan 93%. Sedangkan untuk recall (tingkat keberhasilan) menghasilkan keberhasilan yaitu untuk batik truntum sebesar 99%, batik semenrante 89%, batik kawung 97%, batik parang 96%, batik ceplok 95%, batik sidomukti 100%, dan batik buketan 99%.

### 3.6 Hasil Prediksi untuk Deteksi Persatu Gambar



▶ y

```
array([[3.7572228e-08, 9.6105159e-09, 9.3996841e-06, 5.0368293e-15,  
3.6882419e-16, 9.9998987e-01, 7.5951971e-07]], dtype=float32)
```

```
[39] print( labels[np.argmax(y)], np.max(y) )
```

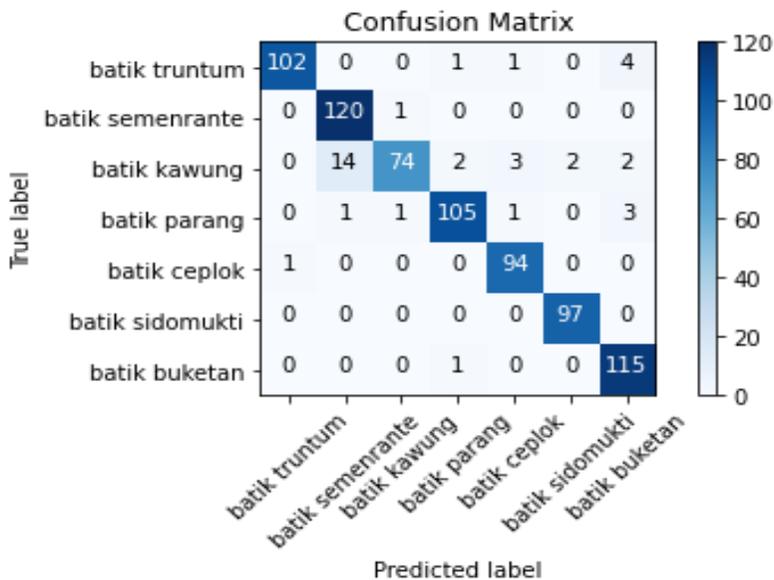
```
▶ batik sidomukti 0.99998987
```

**Gambar 12.** Hasil Prediksi gambar

Hasil prediksi untuk deteksi persatu gambar bisa dilihat pada gambar 5.11. Variabel *y* merupakan nilai hasil dari prediksi yang menghasilkan banyak nilai dari masing-masing kategori. Ada 7 kategori kelas yang telah di labelkan bisa kita lihat nilai terbesarnya ada pada kolom ke 6 yang merupakan kelas batik sidomukti berdasarkan label yang sudah di definisikan. Hasil dari diprediksi untuk mendeteksi gambar tersebut adalah benar bahwa gambar tersebut merupakan batik sidomukti dengan nilai probabilitas tertinggi yaitu 0,99%.

### 3.7 Evaluasi dan Prediksi Hasil *Train* dan *Test* dari Seluruh Kelas

Untuk melihat hasil prediksi dari data *training* dan data *testing* akan ditunjukkan melalui tabel *confusion matrix* seperti gambar dibawah ini.



Gambar 13. Hasil Klasifikasi Data Baru

Gambar diatas merupakan hasil dari prediksi data baru, dimana dapat dilihat prediksi yang didapat bahwa dari 745 data gambar yang dimasukkan komputer semua data berhasil di klasifikasi dengan baik. Pada batik truntum data yang dimasuk sebanyak 103 gambar yang berhasil di klasifikasi adalah sebanyak 102 benar dan 1 salah, batik semenrante data yang dimasukkan sebanyak 135 yang berhasil diklasifikasi dengan benar sebanyak 120 dan 15 salah, batik kawung data yang dimasukkan sebanyak 76 yang berhasil diklasifikasi dengan benar sebanyak 74 dan 2 salah, batik parang data yang dimasukkan sebanyak 109 yang berhasil diklasifikasi dengan benar sebanyak 105 dan 4 salah, batik ceplok data gambar yang dimasukkan sebanyak 99 yang berhasil diklasifikasin dengan benar sebanyak 94 dan 5 salah, batik sidomukti data yang dimasukkan sebanyak 99 yang berhasil diklasifikasi dengan benar sebanya 97 dan salah, dan batik buketan data yang dimasukkan sebanyak 124 yang berhasil diklasifikasi dengan benar sebanyak 115 dan 9 salah. Dari penelitian yang di lakukan , hasil akurasi yang didapatkan dari data *trining* dan *testing* menghasilkan akurasi yang cukup bagus. Maka dapat dikatakan bahwa metode ini sangat baik untuk melakukan klasifikasi gambar batik.

## 4. KESIMPULAN

Berdasarkan pembahasan yang telah dilakukan, maka dapat disimpulkan bahwa Pembuatan model klasifikasi data citra motif batik solo telah berhasil dilakukan dengan menggunakan metode *deep learning* yang berarsitektur Convolutional Neural Network (CNN). Model CNN pada penelitian ini menggunakan input shape berukuran 32x32x3, ukuran filter 3x3, jumlah *epoch* sebanyak 100. Data yang digunakan untuk proses training model sebanyak 2256 menghasilkan tingkat akurasi training dan testing dalam melakukan deteksi gambar citra batik solo sebesar 99% untuk *accuracy* dan 94% untuk validasi *accuracy*. Penelitian ini menggunakan data testing baru sebanyak 745 gambar dimana perkelas terdapat 96 sampai 127 gambar untuk diuji kedalam model yang telah dibuat. Hasil testing menghasilkan tingkat akurasi baru dalam mengklasifikasi motif citra batik solo yaitu sebesar 95%.

## DAFTAR PUSTAKA

- Goodfellow, I., Bengio, Y., dan Courville, A. 2016. *Deep Learning*. Diperoleh pada 25 Januari 2018 dari <http://goodfeli.github.io/dlbook/>.
- Lestari, S. D. (2012). *Mengenal Aneka Batik*. PT Balai Pustaka (Persero).
- Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J. M., Mattern, F., Mitchell, J. C., Naor, M., Nierstrasz, O., Pandu Rangan, C., Steffen, B., Sudan, M., Terzopoulos, D., Tygar, D., Vardi, M. Y., Weikum, G., Scherer, D., Müller, A., & Behnke, S. (2010). Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition. In K. Diamantaras, W. Duch, & L. S. Iliadis (Eds.), *Artificial Neural Networks – ICANN 2010* (Vol. 6354, pp. 92–101). Springer Berlin Heidelberg. [https://doi.org/10.1007/978-3-642-15825-4\\_10](https://doi.org/10.1007/978-3-642-15825-4_10)
- Putri, R. A., & Rochmawati, N. (2019). *Penerapan Algoritma Support Vector Machine untuk Klasifikasi Motif Citra Batik Solo Berdasarkan Fitur Multi-Autoencoders. 01*, 8.
- RISMIYATI, & M.t, D. A. S. (2016). *IMPLEMENTASI CONVOLUTIONAL NEURAL NETWORK UNTUK SORTASI MUTU SALAK EKSPOR BERBASIS CITRA DIGITAL* [Universitas Gadjah Mada]. [http://etd.repository.ugm.ac.id/index.php?mod=penelitian\\_detail&sub=PenelitianDetail&act=view&typ=html&buku\\_id=101746&obyek\\_id=4](http://etd.repository.ugm.ac.id/index.php?mod=penelitian_detail&sub=PenelitianDetail&act=view&typ=html&buku_id=101746&obyek_id=4)
- Sugiyono. 2016. *Metode Penelitian Kuantitatif, Kualitatif dan R&D*. Bandung: PT Alfabet.