

Perbandingan Algoritma SVM Dan CNN Untuk Klasifikasi Citra Batik Nitik

Abstrak

Di Indonesia, berbagai motif batik tersebar di seluruh daerah sehingga menyulitkan untuk mengidentifikasi motif-motif tersebut. Kesalahan dalam klasifikasi motif batik akan menyebabkan misinformasi sehingga informasi tentang motif batik tidak tersampaikan dengan baik. Hal ini bisa menjadi penghambat dalam pengenalan berbagai motif batik secara digital. Motif batik dapat diklasifikasikan dengan metode machine learning atau deep learning. Algoritma yang banyak digunakan pada masing-masing metode tersebut adalah Support Vector Machine (SVM) dan Convolutional Neural Network (CNN) yang dapat digunakan untuk membantu mengenali dan mengidentifikasi motif batik. SVM dan CNN akan melakukan pemodelan dengan menggunakan dataset batik nitik 960 kemudian membandingkan performa kedua model. Dataset batik nitik terdiri dari 960 data citra yang terbagi kedalam 60 kelas. SVM dibangun dengan menggunakan ekstraksi fitur color moment dan MTCD, sedangkan CNN menggunakan arsitektur VGG16 pretrained. Berdasarkan hasil analisis matriks evaluasi seperti akurasi, presisi, recall, dan F1-score model CNN dengan arsitektur VGG16 mendapatkan akurasi konsisten di angka 100% pada data testing dengan nilai learning rate = 0,001. Sedangkan SVM mendapatkan akurasi yang bervariasi pada beberapa pengujian berdasarkan pembagian jumlah dataset.

Kata kunci—SVM, CNN, VGG16, Batik Nitik

Abstract

In Indonesia, various batik motifs are spread across different regions, making it difficult to identify them. Errors in batik motif classification can lead to misinformation, preventing accurate dissemination of information. This can hinder the digital recognition of batik motifs. Batik motifs can be classified using machine learning or deep learning methods. Commonly used algorithms include Support Vector Machine (SVM) and Convolutional Neural Network (CNN), which help recognize and identify batik motifs. SVM and CNN models were trained using the Batik Nitik dataset, which consists of 960 images divided into 60 classes. SVM was built using color moment and MTCD feature extraction, while CNN used a pretrained VGG16 architecture. Based on evaluation metrics such as accuracy, precision, recall, and F1-score, the CNN model with VGG16 achieved a consistent 100% accuracy on the test data with a learning rate of 0.001. Meanwhile, SVM showed varying accuracy depending on the dataset split.

Keywords—SVM, CNN, VGG16, Batik Nitik

1. PENDAHULUAN

Indonesia merupakan negara yang kaya akan sumber daya alam, budaya, dan pariwisata. Salah satu warisan budaya yang terkenal di Indonesia adalah batik. Beragam motif batik dapat dijumpai di Indonesia. Menurut penelitian Bandung Fe Institute dan Sobat Budaya (2015), ada sekitar 5.849 motif batik yang tersebar dari Aceh hingga Papua. Banyaknya motif batik menyebabkan sulitnya masyarakat mengenal motif-motif tersebut terlebih lagi untuk

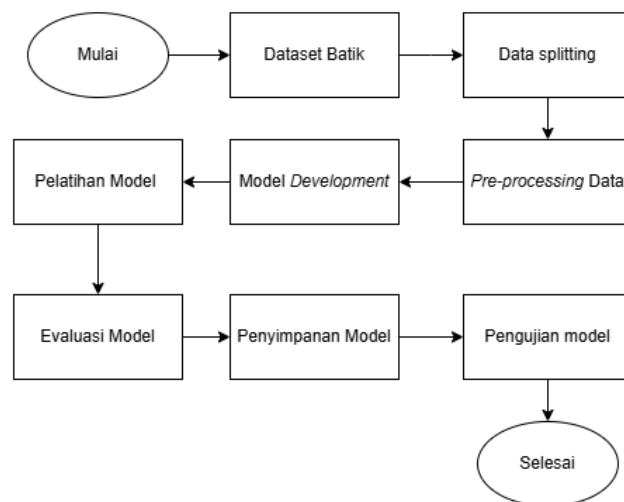
mengetahui makna filosofis dan keunikan apa yang ada pada motif-motif batik tersebut. Perkembangan teknologi informasi dapat dimanfaatkan dalam klasifikasi jenis-jenis motif batik. Salah satu teknologi yang sedang berkembang adalah *image classification*. Definisi dari *image classification* adalah proses pengenalan dan pengelompokan piksel-piksel pada citra berdasarkan karakteristik, pola, dan kriteria tertentu. Klasifikasi citra motif batik dapat dilakukan dengan beberapa algoritma *machine learning* ataupun *deep learning*. Contoh beberapa algoritma *machine learning* adalah *Linear Regression*, *Decision Tree*, *Support Vector Machine* (SVM), *Random Forest* dan lainnya. Sedangkan untuk algoritma *deep learning* yang cukup banyak digunakan untuk klasifikasi citra adalah *Convolutional Neural Network* (CNN).

Beberapa penelitian sebelumnya sudah dilakukan untuk mendapatkan algoritma terbaik untuk klasifikasi seperti klasifikasi batik dengan *K-Nearest Neighbor* [1], perbandingan *support vector machine* (SVM) dan *Backpropagation Neural Network* (BPNN) [2], *Decision Tree* [3], CNN dengan arsitektur VGG16 [4]. Selain itu penelitian mengenai klasifikasi batik mencakup proses pengolahan dataset, *preprocessing*. Berikut ini beberapa topik penelitian tersebut, pengaruh augmentasi dataset pada algoritma CNN dengan arsitektur ResNet18 [5], pengaruh augmentasi dataset dengan algoritma CNN dan KNN [6], pendeteksian bias pada arsitektur CNN *pretrained* [7]. Pada algoritma *machine learning* proses ekstraksi fitur sangat mempengaruhi performa model, berikut ini beberapa penelitian mengenai ekstraksi fitur, SVM dengan MTCO, GLCM, dan MTH [8], perbandingan *self-organizing map* (SMO) dengan dataset RGB dan HSV dengan ekstraksi fitur GLCM [9]. Selain pemilihan algoritma, dataset yang digunakan juga menjadi faktor penting untuk membangun model yang dapat melakukan klasifikasi dengan akurat. Salah satu dataset batik yang bisa diakses secara publik adalah batik nitik 960 [10]. Berdasarkan beberapa penelitian tersebut, masih sedikit penelitian mengenai dataset batik nitik 960. Oleh karena itu, penelitian ini akan berfokus pada membangun model dengan dataset batik nitik dengan algoritma *machine learning* dan *deep learning*.

Pada penelitian ini akan membandingkan performa model dengan SVM dan CNN. Kedua algoritma tersebut adalah representasi dari *machine learning* dan *deep learning*. Pada SVM, akan menggunakan ekstraksi fitur *color moment* dan *Multi Texton Co Occurrence Descriptor* (MTCO). Sedangkan CNN menggunakan arsitektur *pretrained* VGG16. Proses evaluasi model akan menggunakan *confusion matrix* untuk mengukur seberapa bagus model yang sudah dibangun.

2. METODE PENELITIAN

Metode penelitian dimulai dari dataset batik nitik, kemudian data *splitting*, *preprocessing* data, pembangunan model SVM dan CNN, evaluasi model, sampai dengan testing model. Diagram alir dari perancangan sistem ditunjukkan pada gambar 1.



Gambar 1 Rancangan sistem secara umum

2.1 Dataset Batik Nitik

Dataset yang digunakan pada penelitian ini adalah batik nitik 960. Dataset tersebut disediakan oleh Paguyuban Pecinta Batik Indonesia (PPBI) Sekar Jagad Yogyakarta dan proses ekstraksi dilakukan oleh galeri APIPS [10]. Dataset ini terdiri dari 60 pola atau kelas yang masing-masing kelas terdiri dari 4 citra dengan dimensi 512x512 piksel. Keempat citra tersebut diputar dengan sudut 90, 180, dan 270 derajat. Sehingga jumlah citra tiap kelas adalah 16 citra, dengan total seluruh data adalah 960 citra. Gambar 2 menunjukkan beberapa pola batik nitik 960, setiap motif batik terdapat penjelasan latar belakang dan makna filosofis. Dataset ini bersifat umum dan tersedia untuk diakses oleh masyarakat luas.



Gambar 2 Batik nitik

Dataset ini juga memiliki kualitas gambar yang minim noise sehingga pada proses preprocessing tidak terlalu banyak dilakukan manipulasi untuk meningkatkan kualitas dataset.

2.2 Data splitting

Jumlah Citra pada dataset ini berjumlah 960 citra yang dibagi kedalam 60 kelas yang sudah dilabeli. Dalam membangun model SVM dan CNN citra-citra tersebut akan dibagi menjadi dua dataset.

Dataset pertama akan dibagi menjadi dua kategori kategori yaitu data latih dan data uji. Data latih digunakan untuk melatih model supaya dapat mengenali pola dan data uji digunakan untuk mengevaluasi seberapa baik model yang sudah dibangun, data ini adalah data yang belum pernah digunakan untuk melatih model. Pembagian antara data latih dan data uji adalah 80:20. Sehingga pada setiap kelas terdapat 12 data latih dan 4 data uji.

Dataset kedua akan dibagi menjadi 3 kategori yaitu data training, data validasi, dan data testing dimana rasio pembagian adalah 50:25:25. Untuk data testing sebesar 25% dari keseluruhan data akan diambil satu data yang tidak di augmentasi untuk menjaga model tidak membaca data yang sama saat pelatihan. Sehingga jumlah citra per kelas ada 8 data latih, 4 data validasi, dan 1 data uji.

2.3 Preprocessing Data

Setelah dataset dibagi menjadi dua kategori, citra-citra dimasing-masing kategori akan dilakukan proses perubahan ukuran citra. Ukuran citra sebelum proses preprocessing berukuran 512x512 piksel yang akan diubah menjadi 224x224 piksel. Perubahan ukuran citra bertujuan untuk konsistensi ukuran pada semua citra, mengurangi beban komputasi dengan mereduksi jumlah piksel pada setiap citra tanpa menghilangkan banyak informasi penting, dan menyesuaikan arsitektur VGG16 pretrained yang memiliki ukuran input 224x224 piksel.

Tahap selanjutnya adalah normalisasi nilai fitur. Normalisasi bertujuan untuk mengubah skala nilai fitur dalam rentang tertentu. Normalisasi dilakukan supaya meningkatkan stabilitas model, mempercepat konvergensi saat training, dan menjaga konsistensi nilai fitur sehingga model tidak terjebak dalam skala yang berbeda-beda antar fitur. Pada SVM metode normalisasi

yang dipakai adalah z score yaitu mengubah nilai mean menjadi 0 dan standar deviasi menjadi 1. Metode ini cocok untuk algoritma machine learning karena algoritma tersebut sensitif terhadap skala fitur. Dengan menggunakan z-score, semua fitur memiliki rata-rata nol dan skala seragam, sehingga model lebih stabil. Sedangkan VGG16 menggunakan metode min-max normalization yang akan mengubah nilai dalam rentang 0 sampai 1. Min-max normalization mencegah bobot menjadi terlalu besar atau kecil, sehingga optimalisasi lebih stabil.

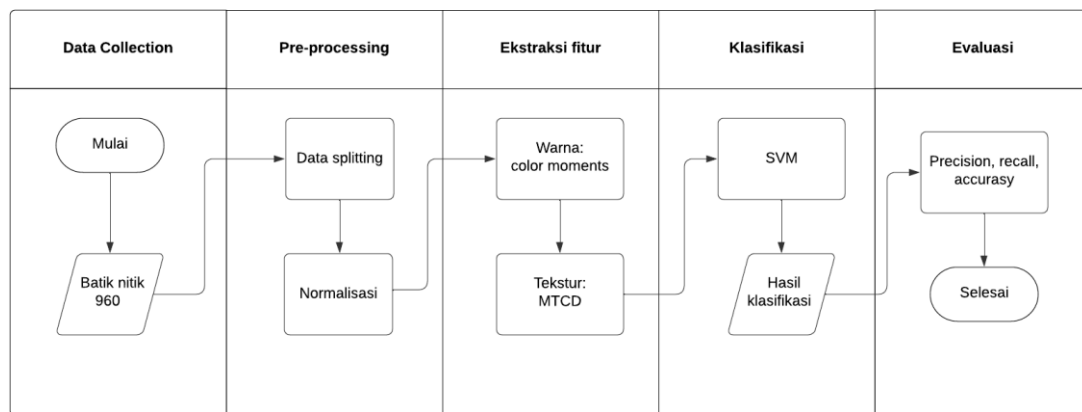
2.4 SVM

SVM merupakan salah satu algoritma klasifikasi dan regresi. Ide utama SVM adalah memisahkan beberapa kelas pada data pelatihan menggunakan bidang pemisah (surface) dengan memaksimalkan margin antar kelas [11]. Fungsi yang digunakan sebagai pemisah antar kelas disebut *hyperplane*. Sementara, data terdekat dari masing-masing kelas dengan hyperplane disebut *support vector*. Persamaan 1 berikut adalah bentuk matematis dari hyperplane.

$$\omega \cdot x + b = 0 \quad (1)$$

Di mana:

- ω : vektor bobot
- x : vektor fitur data
- b : bias



Gambar 3 Diagram alir SVM

Gambar 3 merupakan diagram rancangan sistem pada untuk membangun SVM. Ekstraksi fitur akan menggunakan dua metode yaitu color moment yang mengekstrak fitur warna dan Multi Texton Co-occurrence Descriptor (MTCD) yang mengekstrak fitur tekstur. Ekstraksi fitur merupakan proses mengolah data mentah menjadi data yang lebih informatif dan juga sederhana sehingga bisa digunakan dalam membangun model. Fitur-fitur yang ada di citra digunakan untuk mengidentifikasi pola, membedakan warna, dan mengklasifikasi objek. Ekstraksi fitur juga dapat mempercepat proses training karena terdapat proses pengurangan dimensi dan pengurangan kompleksitas data dengan cara hanya menyimpan informasi yang dianggap penting. Selain itu, ekstraksi fitur dapat meningkatkan akurasi dan kestabilan model jika fitur-fitur hasil ekstraksi merepresentasikan pola yang signifikan pada data.

Ekstraksi fitur pertama yaitu ekstraksi fitur warna dengan metode color moment. Color moment menggunakan tiga moment yaitu mean yang menggambarkan warna paling dominan pada citra, standar deviasi mengukur persebaran variasi warna, dan skewness menunjukkan asimetri distribusi warna. Dengan metode ini dapat menangkap karakteristik unik warna pada citra.

Momen 1 – Mean:

$$E_i = \sum_{j=1}^N \frac{1}{N} p_{ij} \quad (2)$$

Mean berarti nilai rata-rata warna pada citra. Pada persamaan 2 E_i adalah nilai mean untuk ruang warna ke- i , N untuk jumlah total piksel citra, dan p_{ij} adalah nilai piksel ke- j untuk ruang warna ke- i .

Momen 2 – Standar deviasi

$$\sigma_i = \sqrt{\left(\frac{1}{N} \sum_{j=1}^N (p_{ij} - E_i)^2 \right)} \quad (3)$$

Di mana E_i adalah mean dari ruang warna ke- i pada persamaan 3. Standar deviasi (σ_i) mengukur variasi warna pada citra. Semakin tinggi nilainya maka variasi warna pada citra semakin tinggi dan semakin rendah maka warna pada citra cenderung seragam.

Momen 3 – Skewness:

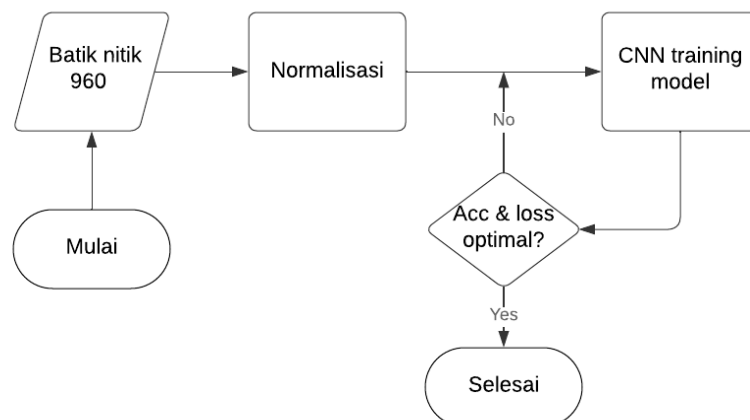
$$s_i = \sqrt[3]{\left(\frac{1}{N} \sum_{j=1}^N (p_{ij} - E_i)^3 \right)} \quad (4)$$

Pada persamaan 4, skewness (s_i) menunjukkan nilai asimetris distribusi warna pada ruang warna.

Ekstraksi fitur kedua adalah MTCO (Multi Texton Co-occurrence Descriptor) yang mengekstraksi fitur tekstur. Ekstraksi fitur ini akan menangkap ciri tekstur sebagai ciri penting untuk proses selanjutnya. MTCO menangkap hubungan antar tekston (pola garis, tepi atau titik yang berulang) suatu citra.

Selanjutnya data hasil ekstraksi fitur akan diolah oleh SVM. Algoritma SVM akan mengelompokkan data berdasarkan hyperplane optimal yang memisahkan data antar kelas. Pada penelitian ini akan digunakan tiga kernel yaitu linear, polinomial, dan RBF (Radial Basis Function). Dari tiga kernel tersebut akan dibangun beberapa model dengan memvariasikan nilai C untuk mendapatkan model dengan performa terbaik.

2.5 VGG16



Gambar 4 Diagram alir perancangan sistem CNN

Model kedua yang akan dibangun adalah CNN dengan arsitektur VGG16. Berdasarkan gambar 4, setelah dataset dinormalisasi tahap berikutnya adalah dataset akan dilatih dengan

arsitektur VGG16 yang menggunakan nilai bobot (weight) dari model yang sudah dilatih (Pre-trained). Model akan dilatih berulang ulang hingga menemukan performa terbaik berdasarkan nilai loss function yang minimum. Proses ini akan mengubah nilai bobot melalui backpropagation dan optimasi sehingga model mampu mengenali pola citra dengan baik.

Layer (type)	Output Shape	Param #
vgg16 (Functional)	(None, 7, 7, 512)	14,714,688
↳ input_layer_3 (InputLayer)	(None, 224, 224, 3)	0
↳ block1_conv1 (Conv2D)	(None, 224, 224, 64)	1,792
↳ block1_conv2 (Conv2D)	(None, 224, 224, 64)	36,928
↳ block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
↳ block2_conv1 (Conv2D)	(None, 112, 112, 128)	73,856
↳ block2_conv2 (Conv2D)	(None, 112, 112, 128)	147,584
↳ block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
↳ block3_conv1 (Conv2D)	(None, 56, 56, 256)	295,168
↳ block3_conv2 (Conv2D)	(None, 56, 56, 256)	590,880
↳ block3_conv3 (Conv2D)	(None, 56, 56, 256)	590,880
↳ block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
↳ block4_conv1 (Conv2D)	(None, 28, 28, 512)	1,180,160
↳ block4_conv2 (Conv2D)	(None, 28, 28, 512)	2,359,808
↳ block4_conv3 (Conv2D)	(None, 28, 28, 512)	2,359,808
↳ block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
↳ block5_conv1 (Conv2D)	(None, 14, 14, 512)	2,359,808
↳ block5_conv2 (Conv2D)	(None, 14, 14, 512)	2,359,808
↳ block5_conv3 (Conv2D)	(None, 14, 14, 512)	2,359,808
↳ block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten_2 (Flatten)	(None, 25088)	0
dense_5 (Dense)	(None, 4096)	102,764,544
dropout_4 (Dropout)	(None, 4096)	0
dense_6 (Dense)	(None, 4096)	16,781,312
dropout_5 (Dropout)	(None, 4096)	0
dense_7 (Dense)	(None, 60)	245,820

Total params: 134,506,364 (513.10 MB)
 Trainable params: 119,791,676 (456.97 MB)
 Non-trainable params: 14,714,688 (56.13 MB)

Gambar 5 Arsitektur VGG16

Arsitektur VGG16 memiliki 16 layer yang terdiri dari 13 convolutional layer dan 3 fully connected (dense). Tabel 5 menampilkan arsitektur VGG16. VGG16 akan menerima data masukan dengan dimensi (224, 224, 3). Pada lapisan convolutional akan dilakukan filtering dengan menggunakan kernel berukuran 3x3. Setelah beberapa lapisan convolutional, terdapat layer max pooling dengan ukuran kernel 2x2 yang bertujuan untuk mereduksi fitur pada citra. Lapisan terakhir terdapat 3 lapisan fully connected yang kemudian akan melewati lapisan softmax untuk mendapatkan keluaran dari model.

2.6 Rancangan Pengujian Sistem

Setelah model dibangun, perlu dilakukan pengujian performa model klasifikasi. Pengujian tersebut dapat dilakukan dengan menggunakan confusion matrix. Pada dasarnya confusion matrix akan membandingkan hasil klasifikasi model dengan hasil klasifikasi sebenarnya. Dari confusion matrix dapat menghitung berbagai performance matrix untuk mengukur kinerja model, diantaranya akurasi, presisi, recall, F1 score.

Akurasi

Akurasi menggambarkan seberapa akurat model dalam mengklasifikasi data dengan benar. Maka, akurasi merupakan rasio prediksi yang benar dengan total keseluruhan data. Akurasi dirumuskan dengan persamaan 4

$$akurasi = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

Keterangan:

TP : True Positive

TN : True Negative

FP : False Positive

FN : False Negative

Presisi

Presisi merupakan rasio dari prediksi benar positif dengan semua hasil yang diprediksi positif. Dari semua prediksi yang benar, seberapa banyak data yang benar-benar positif. Persamaan 5 adalah persamaan untuk memperoleh nilai presisi.

$$presisi = \frac{TP}{TP + FP} \quad (5)$$

Recall

Recall adalah rasio dari prediksi benar dengan seluruh data yang positif. Recall menggambarkan kemampuan model dalam menemukan kembali sebuah informasi. Berikut adalah persamaan 6 untuk mencari nilai recall.

$$recall = \frac{TP}{TP + FN} \quad (6)$$

F1 Score

F1 score bertujuan untuk mengukur keseimbangan antara presisi dan recall. Persamaan 7 adalah persamaan F1 score.

$$F1 - score = 2 \frac{presisi \cdot recall}{presisi + recall} \quad (7)$$

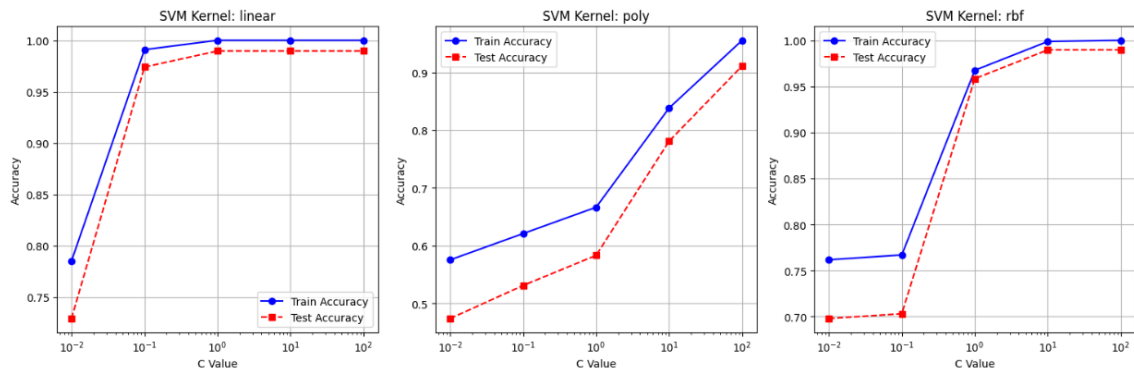
Evaluasi model juga akan melibatkan proses tuning parameter. Hal ini untuk melihat pengaruh perubahan parameter terhadap akurasi model yang dibangun. Model SVM akan mengkombinasikan jenis kernel dan nilai C. Kernel yang dipakai percobaan adalah kernel linear, polinomial, dan RBF. Sedangkan nilainya adalah 0,01; 0,1; 1, 10, 100. Sementara model VGG16 akan dibangun dengan kombinasi *learning rate*, *batch size*, dan *epoch*. Nilai learning yang diuji adalah 0,1; 0,001; 0,00001. Untuk batch size terdiri dari dua nilai yaitu 32 dan 64. Nilai epoch juga terdiri dari dua nilai yaitu 50 dan 100.

3. HASIL DAN PEMBAHASAN

Pada bagian ini akan memaparkan hasil pengujian dari model SVM dan VGG16. Model dibangun berdasarkan dataset batik nitik yang kemudian dibedakan menjadi dua dataset. Dataset pertama (80:20), dimana 80% data training dan 20% testing. Dataset kedua (50:25:25), dimana 50% data training, 25% data validation, dan 1 citra dari 25% lagi yang belum augmentasi sebagai data testing.

3.1 SVM (Dataset 80:20)

Pada penelitian ini, model support vector machine (SVM) dibangun dengan mengkombinasikan beberapa kernel dan juga nilai C. Ada tiga kernel yang digunakan yaitu linear, polinomial, dan RBF. Sementara nilai C yang digunakan adalah 0,01; 0,1; 1, 10, 100. Hasil klasifikasi tersebut menunjukkan jenis kernel dan nilai C berdampak signifikan terhadap performa model.



Gambar 6 a. akurasi kernel linear terhadap nilai C. b. akurasi kernel polinomial terhadap nilai C. c. akurasi kernel RBF terhadap nilai C

Pada gambar 6, ketiga kernel akurasi model konsisten mengalami kenaikan seiring dengan penambahan nilai C. Hal ini mengindikasikan model memerlukan margin dan toleransi yang lebih ketat terhadap kesalahan klasifikasi. Akurasi tertinggi didapatkan oleh kernel linear dan RBF dimana kedua kernel tersebut mendapatkan akurasi tertinggi pada testing sebesar 98%. Sementara kernel polinomial akurasinya hanya sebesar 0.91%. Akurasi tertinggi pada kernel linear pada saat $C = 1$ sampai $C = 100$. Sedangkan pada kernel RBF pada saat nilai $C = 10$ sampai $C = 100$. Akurasi model SVM pada masing-masing kernel terhadap nilai C terdapat pada tabel 1.

Tabel 1 Perbandingan akurasi model SVM dataset 1

Kernel	C	Train Accuracy	Test Accuracy
Linear	0,01	0,712	0,661
Linear	0,10	0,868	0,838
Linear	1,00	0,995	0,969
Linear	10,0	1,000	0,990
Linear	100,00	1,000	0,990
Polinomial	0,01	0,588	0,557
Polinomial	0,10	0,642	0,630
Polinomial	1,00	0,755	0,745
Polinomial	10,0	0,933	0,885
Polinomial	100,0	0,987	0,964
RBF	0,01	0,728	0,646
RBF	0,10	0,739	0,667
RBF	1,00	0,926	0,922
RBF	10,0	0,999	0,990
RBF	100,00	1,000	0,990

3.2 VGG16 (Dataset 80:20)

Model kedua yang dibangun pada penelitian ini menggunakan CNN dengan arsitektur VGG16. Terdapat beberapa kombinasi nilai *learning rate*, *epoch*, dan *batch size*. Pada penelitian ini, peneliti menggunakan tiga nilai learning rate yaitu 0,1; 0,001; dan 0,00001 yang kemudian dikombinasikan dengan nilai epoch dan batch size. Nilai epoch yang dipakai pada pengujian yaitu 50 dan 100, sedangkan untuk batch size nilainya 32 dan 64.

Tabel 2 Perbandingan akurasi VGG16 dataset 1

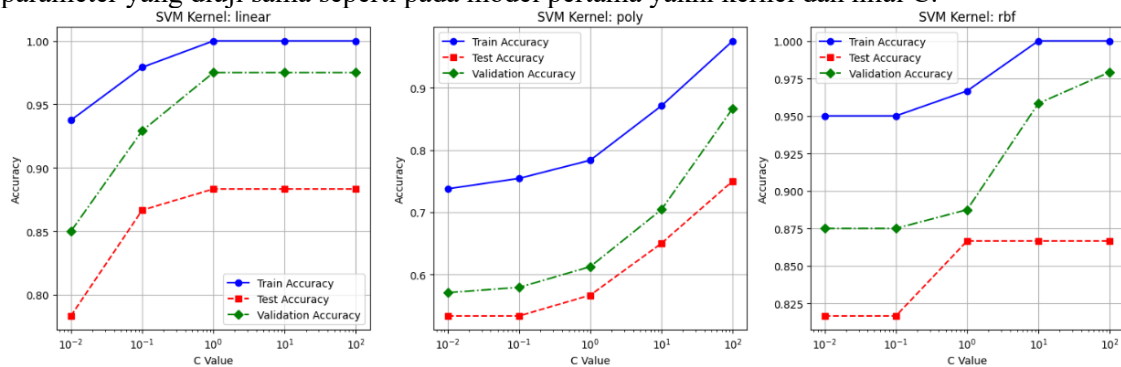
Learning Rate	Epoch	Batch Size	Train Accuracy	Test Accuracy
0,1	50	32	0,01	0,02
0,001	50	32	0,88	1,00
0,00001	50	32	1,00	1,00
0,1	50	64	0,02	0,03

0,001	50	64	0,91	1,00
0,00001	50	64	1,00	1,00
0,1	100	32	0,02	0,03
0,001	100	32	0,90	1,00
0,00001	100	32	1,00	1,00
0,1	100	64	0,04	0,08
0,001	100	64	0,93	1,00
0,00001	100	64	1,00	1,00

Dari tabel 2, dapat dilihat jika model VGG16 mendapatkan akurasi tertinggi pada saat learning rate 0,001 dan 0,00001 dengan akurasi pada data testing sebesar 100%. Sedangkan untuk learning 0,1 model gagal dalam mengenali pola sehingga akurasi yang didapatkan hanya 2%. Pada model ini menggunakan learning rate 0,001 epoch 50 dan batch size 32 sudah mampu mendapatkan akurasi maksimal. Sehingga bisa menghemat penggunaan data komputasi.

3.3 SVM (Dataset 50:25:25)

Model SVM kedua ini dilatih dengan konfigurasi dataset 50% untuk data latih, 25% untuk validasi, dan satu citra dari 25% yang belum di augmentasi untuk data uji. Kombinasi parameter yang diuji sama seperti pada model pertama yakni kernel dan nilai C.



Gambar 7 akurasi kernel linear terhadap nilai C. b. akurasi kernel polinomial terhadap nilai C. c. akurasi kernel RBF terhadap nilai C

Pada Gambar 7, akurasi model SVM masih konsisten mengalami kenaikan seiring dengan bertambahnya nilai C. Kernel linear mendapatkan nilai akurasi tertinggi dengan akurasi pada data testing sebesar 88%. Sedangkan kernel RBF dan polinomial mendapatkan nilai 86% dan 75%. Terdapat penurunan akurasi pada model jika dibandingkan dengan percobaan pertama, hal ini dikarenakan berkurangnya jumlah data latih. Model pada SVM ini juga cenderung mengalami overfitting pada semua kernel. Tabel 3 menunjukkan hasil akurasi setiap kernel dan nilai C

Tabel 3 Perbandingan akurasi model SVM dataset 2

Kernel	C	Train Accuracy	Validation Accuracy	Test Accuracy
Linear	0,01	0,93	0,85	0,78
Linear	0,10	0,97	0,92	0,86
Linear	1,00	1,00	0,97	0,88
Linear	10,0	1,00	0,97	0,88
Linear	100,0	1,00	0,97	0,88
Polinomial	0,01	0,73	0,57	0,53
Polinomial	0,10	0,75	0,58	0,53
Polinomial	1,00	0,78	0,61	0,56
Polinomial	10,0	0,87	0,70	0,65

Polinomial	100,0	0,97	0,86	0,75
RBF	0,01	0,95	0,87	0,82
RBF	0,10	0,95	0,87	0,82
RBF	1,00	0,96	0,88	0,86
RBF	10.0	1,00	0,95	0,86
RBF	100	1,00	0,98	0,86

3.4 VGG16 (Dataset 50:25:25)

Pada model VGG16 kali ini akan bangun dengan menggunakan konfigurasi dataset yang kedua, yaitu 50% data latih, 25% data validasi, dan satu citra dari 25% yang belum di augmentasi untuk data uji. Kali ini pelatihan model tidak menggunakan seluruh kombinasi learning rate, epoch, dan batch size. Parameter yang akan dipakai adalah parameter yang mendapatkan hasil paling tinggi pada setiap learning rate percobaan sebelumnya. Hal ini bertujuan untuk menghemat waktu dan sumber daya komputasi. Berikut ini adalah hasil dari pengujian model dengan arsitektur VGG16 yang ditujukan pada tabel 4.

Tabel 4 Perbandingan akurasi VGG16 dataset 2

Learning Rate	Epoch	Batch Size	Train Accuracy	Validation Accuracy	Test Accuracy
0,1	100	64	0,04	0,05	0,03
0,001	100	64	0,99	1,00	1,00
0,00001	100	64	1,00	1,00	1,00

Model VGG16 masih dapat mengklasifikasikan batik nitik dengan sangat baik meskipun jumlah data latih mengalami pengurangan. Model mampu mencapai akurasi maksimal dengan learning rate 0,001 dan 0,00001 dengan epoch 100 dan batch size 64. Pada learning rate tersebut model mencapai akurasi 100% pada data testing. Sedangkan pada saat learning rate 0,1 model VGG16 gagal belajar sehingga akurasi yang didapatkan hanya 3%.

4. KESIMPULAN

Akurasi model di dataset satu, model terbaik didapatkan oleh kernel linear dan RBF dengan nilai $C = 100$. Kedua model tersebut mampu mendapatkan akurasi 100% pada data uji. Untuk dataset dua, semua model mengalami penurunan akurasi, ini dikarenakan pengurangan jumlah data latih. Model terbaik pada dataset dua menggunakan kernel linear dengan akurasi pada data uji sebesar 88%.

Sedangkan, pada model CNN, baik di dataset 1 dan dataset 2, performa model sangat baik pada saat learning rate = 0,001 dan 0,00001 yang mendapatkan akurasi 100% dan loss function yang rendah pada data testing. Dari hasil tersebut dengan learning rate = 0,001 sudah cukup untuk mengklasifikasi batik nitik. Dengan menggunakan learning rate yang kecil maka akan menghemat waktu dan sumber daya komputer.

5. SARAN

Jumlah dataset pada batik nitik 960 masih sedikit untuk menilai apakah model benar-benar belajar dengan data yang ada atau model menghafal pola dari dataset. Dengan memperbanyak dataset maka akan lebih banyak data yang digunakan untuk pelatihan dan pengujian model. Sehingga kemungkinan model mengalami overfitting semakin kecil.

DAFTAR PUSTAKA

- [1] C. Irawan, A. Winarno, H. Kusumodestoni, A. Sucipto, T. Tamrin, and M. Doheir, "A Combination of Statistical Extraction and Texture Features Based on KNN for Batik Classification," *Proc. - 2021 Int. Semin. Appl. Technol. Inf. Commun. IT Oppor. Creat. Digit. Innov. Commun. within Glob. Pandemic, iSemantic 2021*, pp. 113–117, 2021, doi:

- 10.1109/iSemantic52711.2021.9573214.
- [2] W. Herulambang, M. N. Hamidah, and F. Setyatama, "Comparison of SVM and BPNN Methods in the Classification of Batik Patterns Based on Color Histograms and Invariant Moments," *Proceeding - ICoSTA 2020 2020 Int. Conf. Smart Technol. Appl. Empower. Ind. IoT by Implement. Green Technol. Sustain. Dev.*, pp. 31–34, 2020, doi: 10.1109/ICoSTA48221.2020.1570615583.
- [3] A. Tejawati *et al.*, "Batik Pattern Classification Using Decision Tree Based on Color-Texture Features," *2023 6th Int. Conf. Vocat. Educ. Electr. Eng. Integr. Scalable Digit. Connect. Intell. Syst. Green Technol. Educ. Sustain. Community Dev. ICVEE 2023 - Proceeding*, pp. 347–352, 2023, doi: 10.1109/ICVEE59738.2023.10348243.
- [4] N. W. Parwati Septiani *et al.*, "Convolutional Neural Network (CNN) Algorithm for Geometrical Batik Sade' Motifs," *ICCoSITE 2023 - Int. Conf. Comput. Sci. Inf. Technol. Eng. Digit. Transform. Strateg. Facing VUCA TUNA Era*, pp. 597–602, 2023, doi: 10.1109/ICCoSITE57641.2023.10127829.
- [5] D. G. T. Meranggi, N. Yudistira, and Y. A. Sari, "Batik Classification Using Convolutional Neural Network with Data Improvements," *Int. J. Informatics Vis.*, vol. 6, no. 1, pp. 6–11, 2022, doi: 10.30630/joiv.6.1.716.
- [6] D. Trimakno and Kusriani, "Impact of Augmentation on Batik Classification using Convolution Neural Network and K-Nearest Neighbor," *ICOIACT 2021 - 4th Int. Conf. Inf. Commun. Technol. Role AI Heal. Soc. Revolut. Turbul. Era*, pp. 285–289, 2021, doi: 10.1109/ICOIACT53268.2021.9564000.
- [7] A. C. I. Ardison, M. J. R. Hutagalung, R. Chernando, and T. W. Cenggoro, "Observing Pre-Trained Convolutional Neural Network (CNN) Layers as Feature Extractor for Detecting Bias in Image Classification Data," *CommIT J.*, vol. 16, no. 2, pp. 149–158, 2022, doi: 10.21512/commit.v16i2.8144.
- [8] A. E. Minarno, Y. Azhar, F. D. Setiawan Sumadi, and Y. Munarko, "A Robust Batik Image Classification using Multi Texton Co-Occurrence Descriptor and Support Vector Machine," *2020 3rd Int. Conf. Intell. Auton. Syst. ICoIAS 2020*, pp. 51–55, 2020, doi: 10.1109/ICoIAS49312.2020.9081833.
- [9] A. D. Wibawa, E. Arif Wicaksono, S. D. Suryani, and R. Rumadi, "Javanese Batik Image Classification using Self-Organizing Map," *ICCoSITE 2023 - Int. Conf. Comput. Sci. Inf. Technol. Eng. Digit. Transform. Strateg. Facing VUCA TUNA Era*, pp. 472–477, 2023, doi: 10.1109/ICCoSITE57641.2023.10127783.
- [10] A. E. Minarno, I. Soesanti, and H. A. Nugroho, "Batik Nitik 960 Dataset for Classification, Retrieval, and Generator," *Data*, vol. 8, no. 4, pp. 1–10, 2023, doi: 10.3390/data8040063.
- [11] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, and A. Lopez, "A comprehensive survey on support vector machine classification: Applications, challenges and trends," *Neurocomputing*, vol. 408, pp. 189–215, 2020, doi: 10.1016/j.neucom.2019.10.118.