

## BAB 2 LANDASAN TEORI

### 2.1 Tinjauan Teori

Dalam penelitian klasifikasi motif batik Cirebon ini, terdapat beberapa teori yang dipahami seperti algoritma CNN dan teori-teori lainnya.

#### 2.1.1 Convolutional Neuron Network

Convolutional Neural Network (CNN) adalah algoritma yang bekerja dengan mengekstraksi fitur melalui konvolusi dan melanjutkan ke tahap klasifikasi menggunakan jaringan saraf tiruan [10, 13]. CNN dikenal sebagai algoritma yang handal dan banyak diterapkan dalam aplikasi pengolahan citra dan identifikasi pola [8]. CNN memiliki prinsip kerja yang hampir menyerupai jaringan saraf tiruan pada umumnya. Namun, perbedaan utamanya terletak pada pemanfaatan kernel berdimensi dua atau lebih pada setiap unit dalam lapisan CNN yang digunakan untuk melakukan proses konvolusi. Kernel ini berfungsi untuk mengekstraksi fitur spasial yang memiliki pola serupa dengan struktur data input. Selain itu, CNN dirancang untuk menggunakan sejumlah parameter guna mereduksi jumlah variabel, sehingga proses pelatihan model menjadi lebih efisien dan terfokus [14, 15]. CNN terdiri atas empat lapisan utama, yaitu lapisan konvolusi (*Convolutional Layer*), lapisan pooling (*Pooling Layer*), lapisan aktivasi (*Activation Layer*), serta lapisan fully connected (*Fully Connected Layer*) [10, 16, 17].

##### 2.1.1.1 Convolutional Layer

Lapisan konvolusi (*Convolutional Layer*) berfungsi untuk melakukan proses konvolusi (*Convolution Operations*), operasi matematis yang melibatkan dua fungsi dengan argumen bernilai nyata. Pada tahap ini, input berupa citra akan diproses untuk menghasilkan output berupa feature map. Baik input maupun output dari operasi ini berbentuk fungsi dengan nilai nyata. Berikut adalah rumus penulisan untuk *Convolution operation* pada umumnya:

$$s(t) = (x * \omega) \quad (2.1)$$

Persamaan 2.1 memberikan hasil berupa feature map sebagai hasil konvolusi

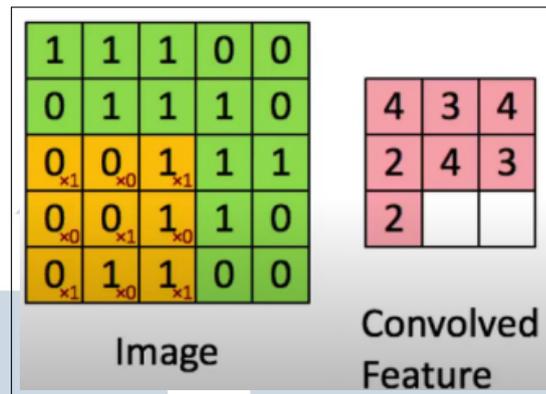
tunggal. Pada persamaan tersebut, argumen pertama yang digunakan adalah input citra yang dilambangkan dengan  $x$ , sedangkan argumen kedua adalah kernel atau filter yang dinotasikan sebagai  $\omega$ . Karena input berupa citra dua dimensi, maka indeks  $t$  dapat direpresentasikan sebagai piksel dan digantikan dengan dua argumen, yaitu  $i$  dan  $j$ , yang masing-masing menunjukkan posisi baris dan kolom pada citra. Oleh karena itu, operasi konvolusi dengan input berdimensi lebih dari satu dapat dituliskan sebagai berikut:

$$s(i, j) = (K * I)(i, j) = \sum_m \sum_n I(i - m, j - n) K(m, n) \quad (2.2)$$

Persamaan 2.2 merupakan perhitungan dasar dari operasi konvolusi, setiap piksel pada citra dinyatakan dengan indeks  $i$  dan  $j$ . Perhitungan ini bersifat komutatif, yang berarti kernel  $K$  dapat dibalik posisinya terhadap input  $I$ . Operasi konvolusi dapat dianalogikan sebagai proses perkalian matriks antara input citra dan kernel, hasil akhirnya diperoleh melalui perhitungan *dot product* pada masing-masing elemen yang bersesuaian. Selain itu, volume keluaran pada setiap lapisan dapat diatur dengan menggunakan hyperparameter. Hyperparameter ini berfungsi untuk menentukan jumlah neuron aktivasi pada satu keluaran, yang dirumuskan dalam persamaan berikut:

$$\frac{(W - F + 2P)}{S + 1} \quad (2.3)$$

Berdasarkan persamaan 2.3, ukuran spasial dari output volume dapat dihitung menggunakan beberapa hyperparameter, yaitu ukuran volume ( $W$ ), ukuran filter ( $F$ ), nilai stride ( $S$ ) yang digunakan, serta jumlah zero padding ( $P$ ) yang diterapkan. Stride adalah parameter yang menentukan seberapa jauh filter bergeser melewati citra input, sedangkan Zero Padding merupakan proses menambahkan nilai nol di sekitar tepi citra untuk mempertahankan atau menyesuaikan ukuran output. Dalam pengolahan citra, operasi konvolusi dilakukan dengan menerapkan kernel (ditandai dengan kotak berwarna kuning) ke seluruh bagian gambar pada setiap posisi yang memungkinkan atau setiap offset, sebagaimana diperlihatkan pada Gambar 2.1.



Gambar 2.1. Convolutional Operation.

Sumber: [17]

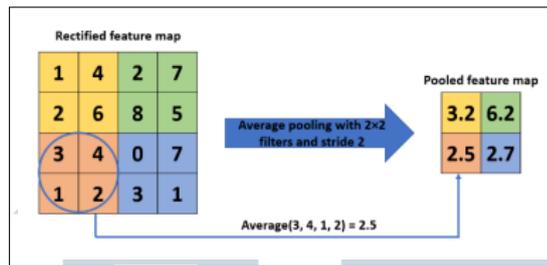
Kotak berwarna hijau secara keseluruhan merepresentasikan citra yang akan dikenai proses konvolusi. Kernel akan bergerak dari sudut kiri atas menuju ke sudut kanan bawah. Hasil dari proses konvolusi pada citra tersebut dapat dilihat pada gambar di sebelah kanan. Tujuan utama dari konvolusi pada data citra ini adalah untuk mengekstraksi fitur-fitur penting dari gambar yang menjadi input.

### 2.1.1.2 Pooling Layer

Pooling Layer merupakan lapisan yang memanfaatkan Feature Map sebagai input untuk kemudian diproses menggunakan berbagai operasi statistik yang didasarkan pada nilai piksel di sekitarnya. Pada arsitektur model CNN, lapisan pooling umumnya ditempatkan secara berkala setelah beberapa lapisan konvolusi. Kehadiran Pooling Layer di antara lapisan-lapisan konvolusi ini berfungsi untuk secara bertahap mengurangi dimensi output dari Feature Map. Pengurangan ini tidak hanya membantu menekan jumlah parameter yang digunakan dan beban komputasi jaringan, tetapi juga berperan dalam mengontrol risiko overfitting. Pooling layer terdiri dari average pooling, max pooling, dan soft pooling [18, 19].

- a) Average Pooling: Average pooling layer melakukan proses down-sampling dengan membagi input ke dalam beberapa area persegi panjang, lalu menghitung nilai rata-rata dari setiap area tersebut. Berikut adalah rumus [19] dan gambar average pooling :

$$f_{\text{avg}}(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^N |x_i| \quad (2.4)$$



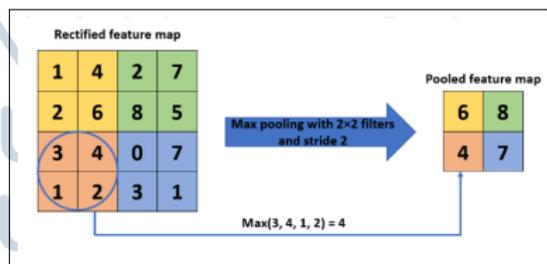
Gambar 2.2. Average Pooling.

Sumber: [18]

Grid di setiap warna yang terdiri dari kuning, hijau, merah biru akan dijumlah nilainya dan dibagi dengan jumlah grid di setiap warna tersebut. Sehingga hasilnya bisa dilihat bagian kanan Gambar 2.2. Kelemahan lainnya adalah berkurangnya informasi dalam hal kontras. Pada proses ini, seluruh nilai aktivasi dalam area kotak persegi panjang dipertimbangkan saat menghitung rata-rata. Jika kekuatan dari semua fungsi aktivasi rendah, maka nilai rata-rata yang dihasilkan juga akan rendah, sehingga menyebabkan penurunan kontras. Masalah ini akan semakin parah jika sebagian besar aktivasi pada area pooling bernilai nol. Dalam kondisi tersebut, karakteristik fitur hasil konvolusi akan berkurang secara signifikan [20].

- b) Max Pooling: Max pooling merupakan proses yang membagi output dari convolutional layer ke dalam beberapa wilayah kecil (grid) dan mengambil nilai maksimum dari setiap wilayah tersebut untuk menghasilkan output yang lebih ringkas. Berikut adalah rumus [19] dan gambar max pooling:

$$f_{\max}(\mathbf{x}) = \max_{i=1}^N \{x_i\} \quad (2.5)$$



Gambar 2.3. Max Pooling.

Sumber: [18]

Grid di setiap warna yang terdiri dari kuning, hijau, merah biru akan diambil nilai paling tinggi di setiap warna tersebut. Sehingga hasilnya bisa dilihat bagian kanan Gambar 2.3. Keterbatasan utama dari max pooling adalah hanya mempertimbangkan nilai maksimum dari setiap area pooling, sementara nilai-nilai lainnya diabaikan sepenuhnya. Jika sebagian besar nilai dalam area pooling memiliki magnitudo yang besar, maka karakteristik pembeda akan hilang setelah proses max pooling dilakukan. Kehilangan informasi ini dapat menyebabkan penurunan performa model yang signifikan dan menghasilkan keluaran yang kurang akurat [20].

- c) Soft Pooling: Soft pooling diklaim memiliki kemampuan generalisasi yang lebih baik dibandingkan dengan max pooling. Pada metode ini, dilakukan perhitungan rata-rata tertimbang dari nilai-nilai input di dalam area pooling. Berikut adalah rumus untuk soft pooling:

$$s_j = \left( \frac{1}{|R_j|} \sum_{i \in R_j} a_i^p \right)^{\frac{1}{p}} \quad (2.6)$$

Pada persamaan tersebut,  $s_j$  merupakan output dari operator pooling pada posisi  $j$ , sedangkan  $a_i$  adalah nilai fitur pada posisi  $i$  di dalam area pooling  $R_j$ . Nilai  $p$  bervariasi antara 1 hingga  $\infty$ . Ketika  $p = 1$ , operator  $LP$  berperilaku seperti average pooling, dan saat  $p = \infty$  maka operator ini setara dengan max pooling. Pada  $LP$  pooling, nilai  $p > 1$  dipandang sebagai kompromi antara average pooling dan max pooling.

### 2.1.1.3 Activation Layer

Dalam activation layer, terdiri dari beberapa activation function, yang terdiri dari Sigmoid, Tanh, ReLU, dan SoftMax [21, 22].

- a) Sigmoid: Fungsi aktivasi ini merupakan salah satu yang paling umum digunakan karena bersifat non-linear. Fungsi sigmoid akan mengubah nilai input ke dalam rentang (0,1). Berikut adalah persamaan dari aktivasi Sigmoid:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2.7)$$

b) Tanh: Fungsi Tanh bersifat kontinu dan dapat diturunkan, serta menghasilkan nilai dalam rentang (-1,1). Dibandingkan dengan fungsi sigmoid, fungsi Tanh memiliki gradien yang lebih curam. Fungsi ini lebih sering digunakan daripada sigmoid karena gradiennya tidak terbatas pada arah tertentu dan bersifat zero-centered, sehingga memudahkan proses optimasi. Berikut adalah persamaan dari aktivasi Tanh:

$$\tanh(z) := \frac{e^z - e^{-z}}{e^z + e^{-z}} \quad (2.8)$$

c) ReLU: Keunggulan utama dalam penggunaan fungsi ReLU adalah tidak semua neuron akan aktif secara bersamaan. Artinya, sebuah neuron hanya akan dinonaktifkan apabila hasil dari transformasi linear yang dilakukan bernilai nol. Fungsi ReLU dinilai lebih efisien dibandingkan fungsi aktivasi lainnya, karena tidak mengaktifkan seluruh neuron secara bersamaan, melainkan hanya sejumlah neuron tertentu yang diaktifkan pada waktu yang sama. Berikut adalah persamaan dari aktivasi ReLU:

$$f(x) = \max(0, x) \quad (2.9)$$

d) SoftMax: Fungsi Softmax dapat digunakan pada permasalahan klasifikasi multikelas. Fungsi ini akan menghasilkan probabilitas untuk setiap data terhadap masing-masing kelas, sehingga nilai output merepresentasikan seberapa besar kemungkinan data tersebut termasuk ke dalam setiap kelas yang tersedia. Ketika membangun sebuah jaringan atau model untuk klasifikasi multikelas, maka lapisan output dari jaringan tersebut akan memiliki jumlah neuron yang sama dengan jumlah kelas yang terdapat pada target. Berikut adalah persamaan untuk aktivasi SoftMax:

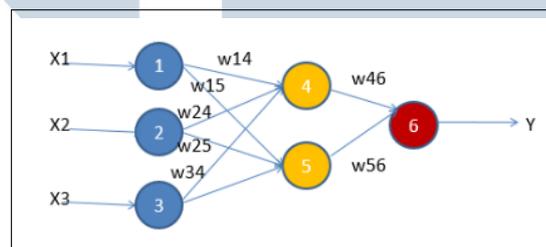
$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \quad \text{for } j = 1, \dots, K. \quad (2.10)$$

dengan  $z$  merupakan faktor input,  $(z)_j$  adalah komponen dari vektor input pada posisi ke- $j$ , dan  $K$  adalah total jumlah kelas yang digunakan dalam proses klasifikasi.

#### 2.1.1.4 Fully Connected Layer

*Fully Connected Layer* merupakan lapisan dalam jaringan ketika setiap neuron pada lapisan tersebut memiliki koneksi penuh terhadap seluruh neuron pada lapisan sebelumnya. Lapisan ini digunakan dalam arsitektur *Multi-Layer Perceptron* (MLP) dan berperan dalam mentransformasikan representasi data agar dapat diproses secara linear dalam tahap klasifikasi [23].

Perbedaan antara *Fully Connected Layer* dan *Convolutional Layer* terletak pada pola konektivitasnya. Pada *Convolutional Layer*, neuron hanya terhubung dengan sebagian kecil dari input (*local receptive field*), sedangkan pada *Fully Connected Layer*, koneksi bersifat menyeluruh sehingga setiap neuron menerima informasi dari seluruh elemen input pada lapisan sebelumnya. [23]. Meskipun demikian, kedua jenis lapisan ini tetap menggunakan operasi *dot product*, sehingga fungsi dasarnya tidak terlalu berbeda. Berikut adalah contoh proses *Fully Connected Layer*:



Gambar 2.4. Proses Fully Connected Layer.

Sumber: [8]

Berdasarkan Gambar 2.4, ditunjukkan arsitektur dari Fully Connected Layer atau Dense Layer pada jaringan neural network. Pada lapisan ini, setiap neuron dari layer sebelumnya terhubung secara penuh ke setiap neuron di layer berikutnya melalui bobot tertentu yang dilambangkan sebagai  $w$ .

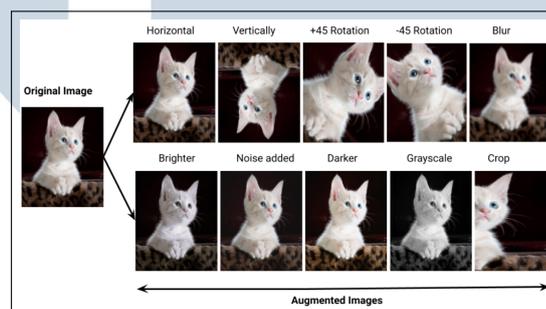
Terdapat tiga neuron input ( $X_1$ ,  $X_2$ ,  $X_3$ ) yang masing-masing terhubung ke dua neuron pada *hidden layer* (4 dan 5) melalui bobot  $w_{14}$ ,  $w_{15}$ ,  $w_{24}$ ,  $w_{25}$ ,  $w_{34}$ , dan  $w_{35}$ . Kemudian, kedua neuron pada *hidden layer* ini dihubungkan lagi ke satu neuron output (6) menggunakan bobot  $w_{46}$  dan  $w_{56}$ , yang akhirnya menghasilkan output  $Y$ .

*Fully Connected Layer* berfungsi untuk mengintegrasikan semua informasi dari layer sebelumnya dan memrosesnya untuk menghasilkan keputusan akhir. Lapisan ini melakukan perhitungan menggunakan operasi dot product antara input

dan bobot, yang dilanjutkan dengan fungsi aktivasi pada masing-masing neuron. Koneksi penuh ini memungkinkan jaringan untuk menangkap hubungan non-linear yang kompleks dalam data.

### 2.1.2 Data Augmentation

*Data augmentation* adalah teknik yang digunakan untuk menambah variasi pada gambar dalam dataset. Teknik ini dilakukan dengan berbagai cara seperti memutar gambar, memperbesar atau memperkecil dimensi gambar, melakukan zoom pada gambar, serta membalik gambar secara horizontal [24]. Proses ini bertujuan untuk meningkatkan keragaman data sehingga model dapat belajar lebih baik dan menjadi lebih robust terhadap variasi pada data nyata [25]. Berikut pada Gambar 2.5 adalah contoh dari *data augmentation*.



Gambar 2.5. Data Augmentation.

### 2.1.3 Evaluation

Model akan diuji menggunakan data latih dan data uji, kemudian dievaluasi dengan menggunakan beberapa parameter. Parameter evaluasi yang digunakan meliputi akurasi, presisi, recall, dan F1-score yang diperoleh dari perhitungan confusion matrix [26].

Tabel 2.1. Confusion Matrix

		Predict	
		Positive	Negative
Actual	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

1. *Precision* (P) merupakan perbandingan antara jumlah prediksi positif yang benar terhadap seluruh prediksi yang diklasifikasikan sebagai positif. Nilai presisi akan didapatkan dari persamaan berikut:

$$P = \frac{TP}{TP + FP} \cdot 100\% \quad (2.11)$$

2. *Recall* (R) merupakan perbandingan antara jumlah observasi positif yang berhasil diprediksi dengan benar terhadap total jumlah sampel positif yang relevan. Nilai recall akan didapatkan dari persamaan berikut:

$$R = \frac{TP}{TP + FN} \cdot 100\% \quad (2.12)$$

3. F1 score adalah rata-rata tertimbang dari Precision dan Recall serta merupakan metode yang digunakan untuk mengukur kinerja model. Nilai F1 score akan didapatkan dari persamaan berikut:

$$F1 \text{ score} = \frac{2PR}{P + R} \quad (2.13)$$

4. Akurasi (*Accuracy*) dihitung sebagai rasio antara jumlah prediksi yang benar terhadap keseluruhan jumlah data yang diuji. Nilai akurasi akan didapatkan dari hasil persamaan berikut:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \cdot 100\% \quad (2.14)$$

U M M N  
U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A