

BAB 2

LANDASAN TEORI

2.1 Tinjauan Teori

Dalam upaya merancang sistem klasifikasi kesegaran daging sapi berbasis CNN dengan implementasi algoritma ekstraksi fitur DCT dan GLCM, dilakukan kajian literatur yang relevan. Pengkajian ini bertujuan untuk memahami perkembangan terkini, sekaligus menyediakan landasan teoritis yang kokoh untuk pengembangan lebih lanjut. Selain itu, penelitian ini akan menyoroti metode-metode inovatif yang telah digunakan untuk meningkatkan akurasi dan efisiensi sistem klasifikasi. Kajian ini juga akan mempertimbangkan potensi integrasi algoritma baru yang dapat memperkaya hasil analisis.

2.1.1 Kesegaran Daging Sapi

Daging adalah bagian otot rangka dari karkas sapi yang biasa dan aman untuk dikonsumsi manusia [12]. Daging dapat berupa daging segar, daging segar yang disimpan dalam kondisi dingin, atau daging yang dibekukan. Daging segar adalah daging yang belum mengalami proses pengolahan dan tidak memiliki tambahan bahan apa pun. Kualitas kesegaran daging sapi dapat ditentukan melalui beberapa indikator, seperti aroma yang khas yang membedakannya dari daging yang tidak segar, tekstur yang kenyal dan padat sehingga tidak mengalami perubahan ketika ditekan dengan tangan, serta tidak menimbulkan bekas yang bertahan dan kembali ke bentuk semula. Daging yang segar juga tidak berlendir dan tidak lengket saat disentuh, namun tetap mempertahankan kelembaban yang tepat [13]. Selain dari faktor aroma dan tekstur, tingkat kesegaran daging juga dapat diamati secara visual melalui warna daging yang merah cerah, serta adanya marbling yang cukup banyak dan warna lemak yang berwarna kekuningan [14]. Marbling adalah pola garis-garis halus dan bintik-bintik lemak putih yang terlihat pada potongan daging, memberikan corak warna dan tekstur khas. Pada daging yang tidak segar, oksidasi dan degradasi lemak dapat menyebabkan perubahan warna dan tekstur marbling, yang membuat marbling tampak lebih kusam dan bahkan bisa mengubah warna daging di sekitarnya [15].

2.1.2 Discrete Cosine Transform (DCT)

Discrete Cosine Transform (DCT) merupakan metode instrumental yang menonjol dalam ekstraksi fitur karena kemampuannya dalam mengubah gambar dari domain spasial ke domain frekuensi [9]. DCT sering digunakan karena keunggulannya dalam mengemas energi dan *decorrelation*, sehingga dianggap sebagai transformasi optimal yang mempertahankan koefisien transformasi dan mengurangi ketergantungan statistik antara koefisien tersebut secara menyeluruh [16]. Ekstraksi fitur DCT dengan mengambil hanya komponen *low-frequency* merupakan metode yang paling optimal, dikarenakan pada level *mid-frequency* dan *high-frequency* banyak terdapat noise [17]. Dasar teorema DCT yang digunakan adalah formula DCT sebagai berikut [18]:

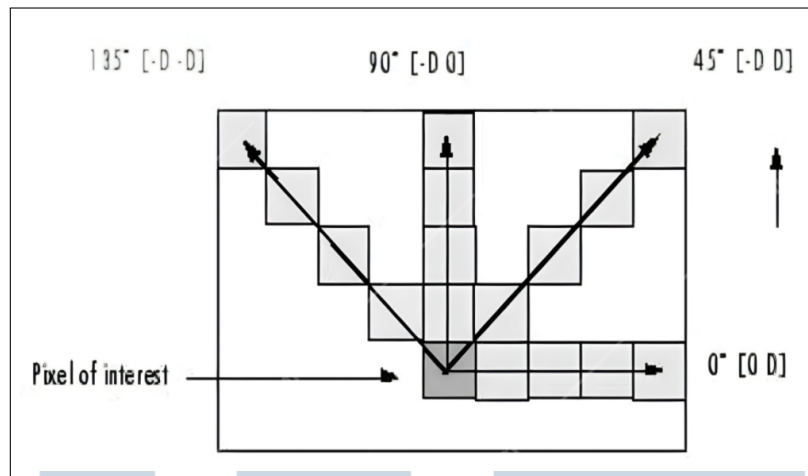
$$C(u) = \alpha(u) \sum_{x=0}^{N-1} f(x) \cos \left[\frac{\pi(2x+1)u}{2N} \right], \quad u = 0, 1, \dots, N-1 \quad (2.1)$$

Sedangkan formula DCT untuk 2D adalah sebagai berikut [18]:

$$C(u, v) = \alpha(u)\alpha(v) \sum_{x=0}^{N-1} \cos \left[\frac{\pi(2x+1)u}{2N} \right] \left\{ \sum_{y=0}^{N-1} f(x, y) \cos \left[\frac{\pi(2x+1)v}{2N} \right] \right\} \quad (2.2)$$

2.1.3 Gray Level Co-occurrence Matrix (GLCM)

Metode fitur ekstraksi *Gray level co-occurrence matrix* (GLCM) secara luas digunakan untuk menyelesaikan permasalahan pada klasifikasi gambar. GLCM mencerminkan informasi statistik urutan kedua dari tingkat keabuan antara piksel-piksel yang berdekatan dalam gambar [19]. GLCM menghitung seberapa sering piksel-piksel dengan nilai dan hubungan spasial yang spesifik muncul pada gambar. Beberapa GLCM juga bisa diterapkan untuk sebuah gambar pada arah dan jarak yang berbeda [9]. Terdapat empat derajat yang umumnya digunakan pada GLCM yaitu 0° , 45° , 90° , dan 135° [20]. Gambar 2.1 merepresentasikan sudut yang digunakan



Gambar 2.1. Sudut GLCM

Sumber: [20]

Ekstraksi fitur GLCM digunakan untuk memperoleh nilai fitur tekstur seperti *contrast*, *correlations*, *energy*, dan *homogeneity* [20].

1. *ASM (Angular Second Moments)* yang menggambarkan keseragaman atau *homogenitas* tekstur dalam citra.

$$ASM = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} P(i, j)^2 \quad (2.3)$$

2. *Contrast (CON)* merupakan fitur yang merujuk pada jumlah perbedaan warna atau skala abu-abu yang muncul pada gambar.

$$CON = \sum_i \sum_j |(i, j)|^2 p(i, j) \quad (2.4)$$

3. *Correlations (COR)* merupakan fitur yang menunjukkan ketergantungan linier dari derajat gambar abu-abu, yang dapat mengindikasikan keberadaan struktur linier dalam gambar.

$$COR = \sum_{i,j} \frac{(i - \mu_i)(j - \mu_j)p(i, j)}{\sigma_i \sigma_j} \quad (2.5)$$

μ_i merupakan nilai rata-rata dari elemen baris matriks $p(i, j)$.

$$\mu_i = \sum_i \sum_j ip(i, j) \quad (2.6)$$

μ_j merupakan nilai rata-rata dari elemen kolom matriks $p(i,j)$.

$$\mu_j = \sum_i \sum_j j p(i, j) \quad (2.7)$$

σ_i merupakan standar deviasi elemen baris matriks $p(i,j)$.

$$\sigma_i = \sqrt{\sum_{i,j} (i - \mu_i)^2 p(i, j)} \quad (2.8)$$

σ_j merupakan standar deviasi elemen kolom matriks $p(i,j)$.

$$\sigma_j = \sqrt{\sum_{i,j} (j - \mu_j)^2 p(i, j)} \quad (2.9)$$

4. *Dissimilarity* merupakan fitur yang mengukur perbedaan pasangan nilai piksel dalam citra.

$$\text{Dissimilarity} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |i - j| \cdot P(i, j) \quad (2.10)$$

5. *Energy* (ENG) merupakan fitur yang mengindikasikan ukuran sifat keseragaman gambar. Jika nilai piksel mirip satu dengan yang lain nilai fitur *energy* akan tinggi.

$$\text{ENG} = \sum_{i,j} p(i, j)^2 \quad (2.11)$$

6. *Homogeneity* (HOM) merupakan fitur yang mengukur homogenitas gambar. Jika semua piksel memiliki nilai yang seragam, nilai *homogeneity* akan tinggi.

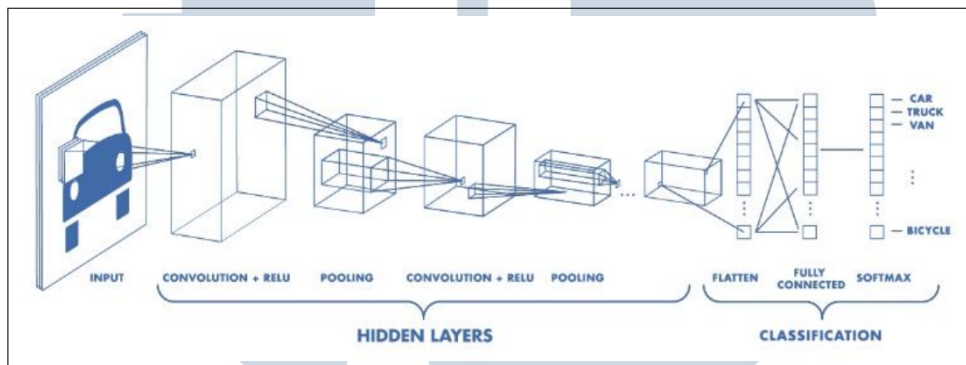
$$\text{HOM} = \sum_{i,j} \frac{p(i, j)}{1 + |i - j|} \quad (2.12)$$

7. *Entropy* merupakan fitur yang mengukur tingkat keacakan atau ketidakpastian distribusi nilai piksel dalam citra.

$$\text{Entropy} = - \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} P(i, j) \cdot \log(P(i, j)) \quad (2.13)$$

2.1.4 Convolutional Neural Network (CNN)

Convolutional neural network (CNN) merupakan cabang Deep Learning yang paling banyak digunakan saat ini. Keuntungan utama yang ditawarkan oleh CNN dibandingkan dengan pendahulunya adalah otomatisasi dalam pendeteksian fitur tanpa perlu pengawasan manusia [4]. CNN memiliki beberapa lapisan seperti *convolution layer*, *pooling layer*, dan *fully connected layer*. Ilustrasi dari lapisan-lapisan tersebut dapat dilihat pada Gambar 2.2.

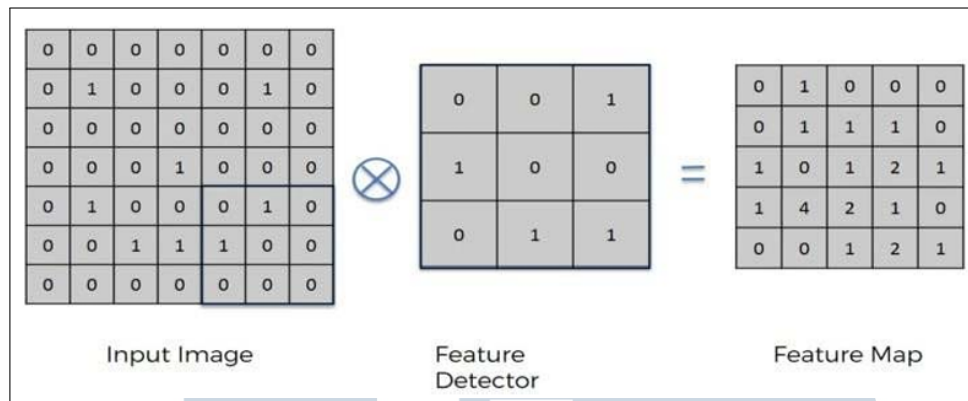


Gambar 2.2. Arsitektur CNN

Sumber: [4]

A Convolution Layer

Convolution layer merupakan lapisan terpenting [21]. *Convolution layer* terdiri dari kumpulan filter konvolusi atau yang disebut juga kernel. Proses konvolusi ini melibatkan pemrosesan gambar input, diekspresikan sebagai matriks N-dimensi, dengan filter-filter untuk menghasilkan *output feature map* [4]. Keuntungan utama dari *output feature map* yaitu kemampuannya untuk menyimpan semua fitur pembeda pada gambar input sambil mereduksi jumlah data yang diproses, sehingga memungkinkan pengenalan pola yang lebih efisien. Matriks yang mengalami konvolusi ini berfungsi sebagai *feature detector*, yang terdiri dari pasangan nilai yang dapat diolah oleh mesin untuk mengidentifikasi pola dalam gambar. Gambar yang berbeda dihasilkan menggunakan nilai *feature detector* yang berbeda [21].



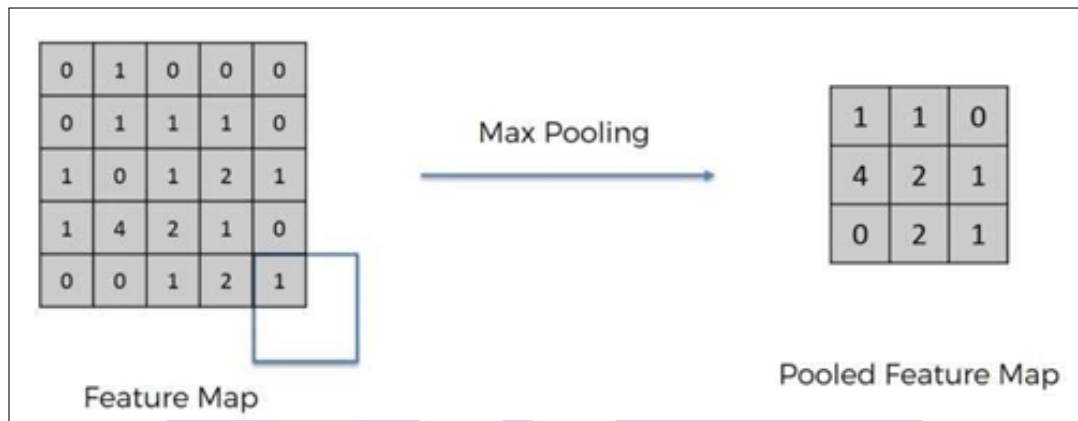
Gambar 2.3. Konvolusi untuk Menghasilkan Feature Map

Sumber: [21]

Gambar 2.3 menjelaskan cara kerja konvolusi. Langkah ini melibatkan konvolusi matriks berisi data gambar dan *feature detector* yang menghasilkan *feature map*. Proses ini menyimpan data dengan nilai 1 atau lebih pada posisi yang sama antara data dan *feature map* yang identik, sedangkan nilai lainnya dihapuskan. Konvolusi adalah operasi dasar dalam CNN yang membantu mendeteksi fitur penting dari gambar. Hal ini dilakukan dengan menggeser *feature detector* di seluruh gambar. Hasilnya adalah ekstraksi fitur yang dapat digunakan untuk tahap selanjutnya dalam jaringan saraf.

B Pooling Layer

Fungsi utama dari *pooling layer* adalah mengambil sampel dari *feature map*. Atau bisa disebut juga mengecilkan *feature map* berukuran besar ke ukuran yang lebih kecil [4]. *Pooling* memungkinkan CNN untuk mengabungkan dimensi yang berbeda dari gambar sehingga berhasil mengenali objek yang diberikan bahkan jika bentuknya miring atau ditampilkan dari sudut yang berbeda. Ada berbagai jenis *pooling* seperti *maxpooling*, *average pooling*, *stochastic pooling*, dan *spatial pyramid pooling*. Namun yang paling populer untuk digunakan adalah *maxpooling* [21].



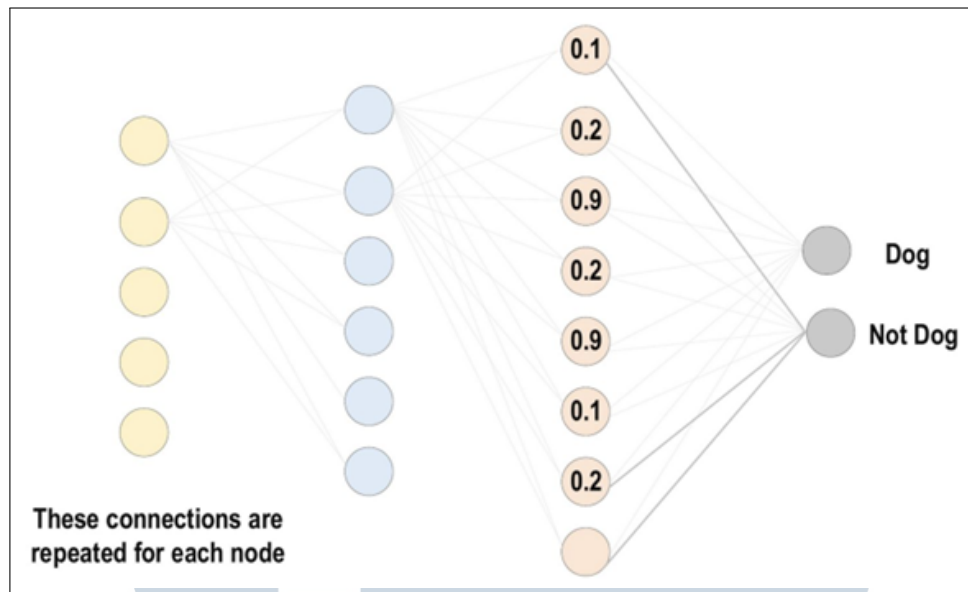
Gambar 2.4. Konvolusi untuk Menghasilkan Feature Map

Sumber: [21]

Gambar 2.4 merupakan visualisasi dari proses *max pooling*. Proses tersebut mengambil nilai tertinggi dari setiap submatriks yang berukuran lebih kecil dalam gambar asli dan membentuk matriks terpisah. Sehingga fitur-fitur terpenting bisa dipertahankan dan dimensionalitas data bisa dikurangi. Proses tersebut memastikan untuk membuat fitur yang dapat dipelajari tetap terbatas jumlahnya sambil menyimpan fitur kunci dari gambar manapun. *Max pooling* umumnya menggunakan filter 2x2, yang secara sistematis menelusuri seluruh gambar untuk mengekstrak nilai maksimum dalam setiap wilayah kecil tersebut.

C Fully Connected Layer

Lapisan ini umumnya terletak pada lapisan paling akhir CNN. Alasan mengapa lapisan ini disebut *fully connected layer* adalah karena dalam lapisan ini, setiap neuron terkoneksi dengan lapisan sebelumnya. Lapisan ini digunakan sebagai proses pengklasifikasi CNN. Proses ini mengikuti *multiple-layer perceptron neural network* konvensional. Inputnya berupa vektor yang dibuat dari *feature map* setelah dilakukan *flatten* [21]. Dari proses tersebutlah keluar hasil akhir CNN, seperti yang diilustrasikan pada gambar 2.5.



Gambar 2.5. Fully Connected Layer

Sumber: [4]

Pelatihan model CNN dimulai dengan membagi dataset gambar input menjadi tiga bagian dengan proporsi yang telah ditentukan. Tiga bagian tersebut yaitu data latih, data uji, dan data validasi. Selanjutnya dalam rangka mencapai kinerja yang optimal, model CNN akan melalui proses optimasi parameter menggunakan algoritma *machine learning*. Proses tersebut bertujuan untuk memperbarui bobot dan parameter model secara berulang dengan memanfaatkan informasi dari data latih. Langkah tersebut diulang secara iteratif hingga mendapat akurasi model yang diinginkan.

Setelah model CNN selesai dilatih, langkah selanjutnya yaitu mengevaluasi kinerjanya menggunakan data uji yang sebelumnya telah dipisahkan. Evaluasi dilakukan dengan menghitung *akurasi*, *presisi*, *recall*, dan *F1-Score*. Dari hasil evaluasi tersebut, diperoleh pemahaman yang jelas mengenai seberapa baik model CNN dapat mengklasifikasikan tingkat kesegaran daging sapi berdasarkan citra yang disediakan. Evaluasi yang dilakukan tersebut akan memberikan pandangan yang lebih menyeluruh tentang keandalan dan generalisasi model CNN dalam berbagai skenario dunia nyata.

2.1.5 Confusion Matrix

Confusion matrix merupakan matriks berukuran $N \times N$ yang digunakan sebagai alat untuk analisis prediksi pada *machine learning*. N merupakan jumlah

kelas yang menjadi target prediksi. *Confusion matrix* berisi tabel ringkasan dari jumlah prediksi mesin klasifikasi yang benar dan salah. Gambar 2.6 merupakan *confusion matrix* 2x2 [22].

		Kelas Sebenarnya	
		Positif	Negatif
Kelas Prediksi	Positif	TP	FP
	Negatif	FN	TN

Gambar 2.6. *Confusion matrix*

Berikut penjelasan dari setiap kolom *confusion matrix*:

- TP: *True Positive* yaitu nilai positif diprediksi model sebagai nilai positif.
- FP: *False Positive* yaitu prediksi model adalah nilai positif, dan hasil prediksinya salah.
- TN: *True Negative* yaitu prediksi model adalah nilai negatif, dan hasil prediksinya salah.
- FN: *True Positive* yaitu nilai negatif diprediksi model sebagai nilai negatif.

Hasil dari *confusion matrix* digunakan untuk menghitung metrik-metrik dalam *classification report*. Berikut merupakan matriks yang dihasilkan oleh *classification report*:

1. *Accuracy* merupakan ukuran seberapa banyak prediksi benar yang dilakukan oleh model.

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.14)$$

2. *Precision* merupakan ukuran seberapa banyak nilai yang diprediksi benar ternyata adalah positif.

$$\text{Precision} = \frac{TP}{TP + TN} \quad (2.15)$$

3. *Recall* merupakan ukuran seberapa besar dari kasus positif yang sebenarnya dapat diprediksi dengan benar oleh model.

$$\text{Recall} = \frac{TP}{TP + FN} \quad (2.16)$$

4. *F1-score* merupakan rata-rata dari *precision* dan *recall* yang memberikan gambaran mengenai kedua metrik tersebut.

$$\text{F1-Score} = \frac{1}{\frac{1}{\text{recall}} + \frac{1}{\text{precision}}} \quad (2.17)$$

