

BAB 2

LANDASAN TEORI

2.1 Perbedaan Daging Segar dan Daging Busuk

Daging segar umumnya memiliki tekstur yang kencang, warna yang cerah, serta aroma yang khas dan segar. Kondisi ini mencerminkan kualitas dan kesegaran daging yang optimal, yang penting untuk dikonsumsi. Tekstur kencang menunjukkan bahwa daging masih memiliki integritas struktural yang baik, sedangkan warna cerah, biasanya merah atau merah muda pada daging sapi, menandakan bahwa hemoglobin dan mioglobin dalam daging masih dalam kondisi yang baik. Aroma segar juga merupakan indikator bahwa daging belum mengalami dekomposisi atau pertumbuhan mikroorganisme yang berlebihan.

Sebaliknya, daging yang sudah busuk akan mengalami perubahan fisik dan kimia yang signifikan. Teksturnya akan menjadi lembek akibat dari proses degradasi protein dan lemak oleh enzim-enzim endogen serta mikroorganisme. Warna daging akan berubah menjadi gelap atau kecoklatan karena oksidasi pigmen daging, seperti mioglobin, yang menyebabkan pembentukan metmioglobin yang berwarna coklat. Selain itu, daging yang busuk akan mengeluarkan bau tidak sedap akibat produksi senyawa volatil oleh mikroorganisme seperti bakteri, jamur, dan ragi. Senyawa-senyawa ini termasuk amonia, hidrogen sulfida, dan amina biogenik, yang semuanya berkontribusi pada bau yang tidak menyenangkan.

Perubahan tersebut terjadi karena proses biokimia yang berlangsung setelah daging dipanen atau disembelih. Salah satu faktor utama yang memengaruhi perubahan ini adalah pertumbuhan mikroorganisme. Mikroorganisme ini dapat berasal dari lingkungan sekitar, peralatan yang digunakan selama proses penyembelihan, atau dari daging itu sendiri. Mereka akan tumbuh dan berkembang biak jika kondisi penyimpanan daging tidak optimal, seperti suhu yang terlalu tinggi atau kelembapan yang tidak terkontrol. Proses pertumbuhan mikroorganisme ini akan mempercepat dekomposisi daging dan menyebabkan penurunan kualitas yang signifikan (Kerry 2002) ^{source}.

2.1.1 Warna Daging

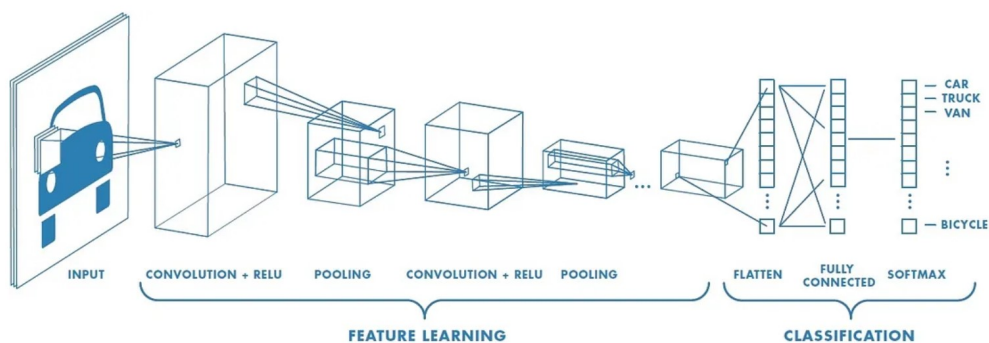
Daging segar memiliki tekstur yang kenyal dan lembut serta warna yang mengkilat, kemerahan, dan terang. Warna merah terang yang cerah pada

daging segar disebabkan oleh keberadaan oksimioglobin, yang merupakan bentuk mioglobin yang teroksidasi dan terjadi ketika daging masih dalam kondisi segar. Warna ini memberikan kesan kesegaran dan kualitas daging yang baik kepada konsumen [11].

Di sisi lain, daging yang sudah tidak segar atau mulai mengalami proses pembusukan dapat memiliki warna yang berbeda. Selain menjadi lebih gelap dan kecoklatan, daging yang mulai membusuk juga dapat menunjukkan warna ungu atau kebiruan. Warna ungu atau kebiruan ini disebabkan oleh adanya metmioglobin, yang terbentuk ketika mioglobin mengalami oksidasi lebih lanjut. Warna yang tidak terang dan kemerahan ini umumnya dianggap sebagai tanda bahwa daging telah mengalami degradasi dan mungkin tidak aman untuk dikonsumsi. Oleh karena itu, konsumen biasanya menghindari daging dengan warna ungu atau kebiruan karena dianggap sebagai tanda bahwa daging tersebut sudah tidak segar lagi [12].

2.2 CNN (Convolutional Neural Network)

Algoritma Convolutional Neural Network (CNN) berfungsi dalam mengenali pola visual dalam data, termasuk aplikasi pada pengolahan citra. CNN melakukan ekstraksi fitur secara otomatis dengan menggunakan serangkaian lapisan konvolusi dan pooling. Lapisan konvolusi bertugas untuk mendeteksi fitur-fitur seperti tepi, sudut, dan pola lainnya dalam gambar. Lapisan pooling mengurangi dimensi gambar sambil tetap mempertahankan informasi penting [13].



Gambar 2.1. Model CNN

Gambar 2.1 menggambarkan arsitektur dari Convolutional Neural Network (CNN) yang umum digunakan untuk klasifikasi gambar. Berikut adalah penjelasan detail dari setiap tahap yang ditunjukkan dalam diagram alir:

1. Input

- **Gambar Input:** Proses dimulai dengan sebuah gambar input, yang dalam contoh ini adalah gambar sebuah mobil. Gambar input dimasukkan ke dalam jaringan untuk dianalisis.

2. Pembelajaran Fitur

- **Konvolusi + ReLU:**
 - **Layer Konvolusi:** Layer ini menerapkan filter konvolusi ke gambar input untuk mengekstraksi berbagai fitur, seperti tepi, tekstur, dan pola. Filter-filter tersebut meluncur di atas gambar input dan membuat peta fitur.
 - **Aktivasi ReLU:** Setelah konvolusi, fungsi aktivasi ReLU (Rectified Linear Unit) diterapkan untuk memperkenalkan non-linearitas ke dalam model, membantu jaringan mempelajari pola-pola kompleks.
- **Pooling:**
 - **Layer Pooling:** Layer ini mengurangi dimensi spasial dari peta fitur dari layer konvolusi. Jenis pooling yang paling umum adalah max pooling, yang memilih nilai maksimum dari suatu wilayah pada peta fitur. Pooling membantu dalam mengurangi beban komputasi dan juga membuat jaringan invarian terhadap translasi kecil dari input.

Langkah-langkah di atas (Konvolusi + ReLU dan Pooling) sering kali diulang beberapa kali untuk membuat jaringan yang lebih dalam yang dapat mempelajari fitur-fitur yang lebih kompleks. Dalam diagram, ini diilustrasikan dengan pengulangan layer Konvolusi + ReLU dan Pooling.

Klasifikasi

- **Flatten:**
 - Setelah beberapa layer konvolusi dan pooling, peta fitur diubah menjadi vektor satu dimensi. Vektor ini kemudian dimasukkan ke dalam layer fully connected.
- **Fully Connected Layer:**

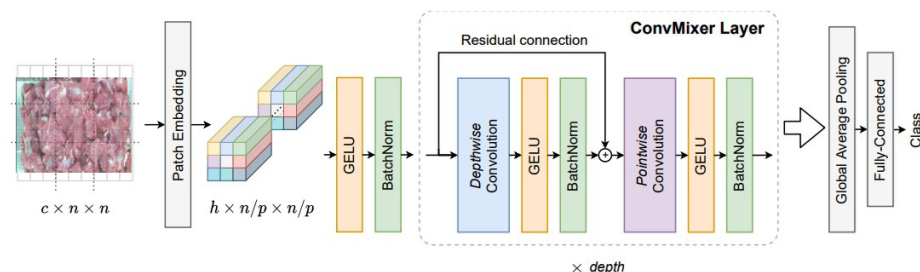
- Layer-layer ini mirip dengan layer-layer dalam jaringan saraf tradisional. Mereka mengambil vektor yang telah di-flatten dan melewatkan vektor tersebut melalui beberapa neuron. Setiap neuron dalam layer fully connected terhubung ke setiap neuron dalam layer sebelumnya, memungkinkan jaringan untuk mempelajari fitur-fitur tingkat tinggi dan melakukan klasifikasi akhir.

- **Softmax:**

- Layer terakhir dalam jaringan adalah layer softmax, yang menghasilkan distribusi probabilitas atas kelas-kelas yang mungkin. Dalam contoh ini, kelas-kelasnya adalah mobil, truk, van, sepeda, dll. Fungsi softmax memastikan bahwa nilai-nilai output berjumlah 1, mewakili probabilitas gambar input milik setiap kelas.

2.3 Model CNN ConvMixer

Dalam konteks ini, model yang akan digunakan untuk analisis adalah ConvMixer, sebuah arsitektur jaringan saraf yang relatif baru dan menjanjikan. ConvMixer membedakan dirinya dengan menggantikan lapisan konvolusi tradisional dengan operasi pencampuran linear, menghasilkan pendekatan yang lebih efisien secara komputasional. Dibandingkan dengan model Convolutional Neural Network (CNN) konvensional, ConvMixer menggunakan lapisan pencampuran saluran dan lapisan pencampuran spasial untuk memproses fitur, dengan fokus pada pemrosesan urutan patch dari gambar input. [14]



Gambar 2.2. Gambaran Arsitektur ConvMixer

A Arsitektur

ConvMixer terdiri dari serangkaian blok mixer. Setiap blok mixer terdiri dari dua komponen utama: lapisan pencampuran saluran (channel-mixing) dan lapisan pencampuran spasial (spatial-mixing). Berikut merupakan penjelasannya:

B Channel-mixing layer

Channel-mixing layer bertanggung jawab untuk mencampur fitur di seluruh saluran. Lapisan ini beroperasi pada peta fitur yang dihasilkan oleh lapisan sebelumnya. Secara khusus, channel-mixing layer menerapkan transformasi linear ke setiap lokasi spasial secara independen. Dengan cara ini, lapisan ini dapat menggabungkan informasi dari berbagai saluran tanpa mengubah posisi spasialnya. Proses ini memungkinkan jaringan untuk memanfaatkan informasi antar-saluran secara lebih efektif, yang penting untuk meningkatkan representasi fitur dalam peta fitur [15].

C Spatial-mixing layer

Spatial-mixing layer mencampur informasi di seluruh lokasi spasial dalam setiap saluran. Ini mengubah bentuk peta fitur menjadi matriks 2D dan menerapkan transformasi linear ke setiap saluran secara independen [16].

D Patch-based processing

ConvMixer memproses gambar input dalam bentuk patch, mirip dengan cara model transformer memproses urutan. Setiap patch dari gambar input dipampatkan menjadi vektor, dan seluruh gambar input direpresentasikan sebagai urutan patch. Hal ini memungkinkan ConvMixer untuk menangkap ketergantungan jarak jauh dengan efisien.

2.4 Embedding Patch

Embedding patch dengan ukuran patch p dan dimensi embedding h dapat diimplementasikan sebagai konvolusi dengan c_{in} saluran input, h saluran output, ukuran kernel p , dan langkah p . Proses ini dapat diformulasikan sebagai berikut:

$$z_0 = \text{BN}(\sigma(\text{Conv}_{\text{in} \rightarrow h}(X, \text{stride} = p, \text{kernel_size} = p))) \quad (2.1)$$

Di sini, X adalah gambar input, BN adalah batch normalization, dan σ adalah fungsi aktivasi (misalnya, ReLU).

2.5 Blok ConvMixer

Blok ConvMixer itu sendiri terdiri dari konvolusi depthwise (yaitu, konvolusi berkelompok dengan jumlah kelompok yang sama dengan jumlah saluran, h) diikuti oleh konvolusi pointwise (yaitu, ukuran kernel 1×1). ConvMixers bekerja paling baik dengan ukuran kernel yang tidak lazim besar untuk konvolusi depthwise. Setiap konvolusi diikuti oleh aktivasi dan BatchNorm pasca-aktivasi [14].

Proses ini dapat dirumuskan sebagai berikut:

$$z_{0l} = \text{BN}(\sigma(\text{ConvDepthwise}(z_{l-1}))) + z_{l-1} \quad (2.2)$$

$$z_{l+1} = \text{BN}(\sigma(\text{ConvPointwise}(z_{0l}))) \quad (2.3)$$

Di sini, ConvDepthwise adalah konvolusi depthwise, dan ConvPointwise adalah konvolusi pointwise dengan ukuran kernel 1×1 . Kombinasi ini memungkinkan ConvMixer untuk menangkap fitur dari gambar input dengan efisien dan menangkap ketergantungan jarak jauh dalam data gambar.

A Positional Coding

Untuk memasukkan informasi posisional ke dalam jaringan, ConvMixer menggunakan pengkodean posisional yang dapat dipelajari. Pengkodean ini ditambahkan ke embedding patch sebelum diberikan kepada jaringan. Dengan demikian, setiap embedding patch tidak hanya mengandung informasi tentang konten visual dari bagian gambar yang sesuai, tetapi juga mencakup informasi tentang posisi relatifnya dalam gambar. Dengan menggunakan pengkodean posisional yang dapat dipelajari, ConvMixer dapat secara efektif memperoleh representasi yang lebih kaya dan lebih kontekstual dari gambar, yang dapat meningkatkan kinerja jaringan dalam berbagai tugas pengolahan gambar. Pendekatan ini memungkinkan ConvMixer untuk mengatasi tantangan dalam

memahami konteks spasial dari gambar dengan lebih baik, yang sering kali kritis dalam pemrosesan gambar secara efektif [15].

B Pelatihan dan optimisasi

ConvMixer dilatih menggunakan backpropagation standar dan teknik optimisasi menggunakan torch.optim.Adam dan fungsi kerugian 'nn.CrossEntropyLoss'. Selama pelatihan, model belajar untuk menyesuaikan parameter-parameternya untuk meminimalkan fungsi kerugian yang diberikan, biasanya suatu ukuran kesalahan prediksi pada dataset pelatihan. Fungsi get accuracy didefinisikan untuk menghitung akurasi model pada set data pelatihan dan uji. Akurasi model dihitung pada kedua dataset setelah pelatihan.[17]

2.5.1 Confusion Matrix

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Gambar 2.3. Gambaran Confusion Matrix

Confusion matrix adalah alat evaluasi kritis dalam analisis klasifikasi yang memberikan pemahaman mendalam tentang kinerja model. Confusion matrix menggambarkan performa model dengan membandingkan hasil klasifikasi aktual dengan prediksi model untuk setiap kelas target. Dari confusion matrix, berbagai metrik evaluasi seperti akurasi, presisi, recall, dan F1-score dapat dihitung, yang masing-masing memberikan wawasan yang berbeda tentang kinerja model. Telaah literatur tentang confusion matrix memberikan pemahaman yang kokoh

tentang bagaimana menerapkan dan menginterpretasikan metrik evaluasi ini untuk mengukur kinerja model CNN dalam klasifikasi gambar daging segar dan busuk. Dengan pemahaman yang mendalam tentang confusion matrix, peneliti dapat membuat keputusan yang lebih tepat dalam memperbaiki dan mengoptimalkan model mereka [18].

- **True Positive (TP):** Situasi di mana model memprediksi kelas positif dan prediksinya benar. Sebagai contoh, jika model bertugas mendeteksi penyakit dalam gambar medis, TP terjadi ketika model memprediksi adanya penyakit dan memang benar ada penyakit dalam gambar tersebut.
- **True Negative (TN):** Situasi di mana model memprediksi kelas negatif dan prediksinya benar. Contohnya, jika model memprediksi bahwa tidak ada penyakit dalam gambar medis dan memang benar tidak ada penyakit dalam gambar tersebut, maka itu adalah TN.
- **False Positive (FP):** Situasi di mana model memprediksi kelas positif tetapi prediksinya salah. Sebagai contoh, jika model memprediksi adanya penyakit dalam gambar medis padahal sebenarnya tidak ada penyakit, ini disebut sebagai FP. False positive juga dikenal sebagai tipe kesalahan tipe I.
- **False Negative (FN):** Situasi di mana model memprediksi kelas negatif tetapi prediksinya salah. Contohnya, jika model memprediksi bahwa tidak ada penyakit dalam gambar medis padahal sebenarnya ada penyakit, maka itu adalah FN. False negative juga dikenal sebagai tipe kesalahan tipe II.

A Accuracy

- **Accuracy** merupakan rasio dari prediksi yang benar (positif dan negatif) terhadap keseluruhan data. Rumusnya adalah:

$$\text{Akurasi} = \frac{TP + TN}{TP + FP + FN + TN}$$

B Precision

- **Precision** merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif. Precision mengukur akurasi dari

prediksi positif yang dibuat oleh model. Rumusnya adalah:

$$\text{Precision} = \frac{TP}{TP + FP}$$

C Recall

- **Recall** merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif. Recall mengukur kemampuan model untuk menemukan semua sampel positif yang ada. Rumusnya adalah:

$$\text{Recall} = \frac{TP}{TP + FN}$$

D F1-Score

- **F1-Score** merupakan rata-rata harmonis dari Precision dan Recall, yang memberikan ukuran seimbang dari kedua metrik tersebut. F1-Score sangat berguna ketika kita membutuhkan keseimbangan antara Precision dan Recall. Rumusnya adalah:

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

