

DETEKSI TINGKAT KEMATANGAN BUAH PEPAYA MENGGUNAKAN MODEL CONVOLUTIONAL NEURAL NETWORK

Naufal Putra Sutrisna*¹, Rafifa Addin Sahirah², Khansa Salsabila Sangdiva Laksono³, Raditya Atmaja Satria Permadhi⁴, Nadhira Nurannisa⁵, Saqina Salsabila Larasati⁶, Wahyu Widyaning Asmani⁷, Novanto Yudistira⁸

^{1,2,3,4,5,6,7,8} Universitas Brawijaya, Malang

Email: ¹naufalptrs@student.ub.ac.id, ²rafifasahirah6@student.ub.ac.id, ³khansalaksono@student.ub.ac.id,
⁴aditdhito@student.ub.ac.id, ⁵nndhiraa@student.ub.ac.id, ⁶saqinasalsabila@student.ub.ac.id,
⁷wahyuwiedya@student.ub.ac.id, ⁸yudistira@ub.ac.id

*Penulis Korespondensi

(Naskah masuk: 23 Noveber 2023, diterima untuk diterbitkan: 14 Mei 2024)

Abstrak

Pepaya adalah buah bergizi tinggi dengan banyak manfaat. Per 100 gram pepaya mengandung 3,65 milligram vitamin A dan 78 milligram vitamin C. Apapun musimnya, buah pepaya dapat berbuah terus menerus, yang menjadi salah satu keunggulannya. Kematangan buah pepaya dapat ditentukan oleh tekstur dan warna kulit buahnya. Deteksi tingkat kematangan buah pepaya memiliki signifikansi penting dalam industri pertanian dan pengelolaan persediaan makanan. Metode konvensional seringkali mengalami keterbatasan dalam akurasi dan efisiensi dalam pengidentifikasian kematangan buah pepaya. Dalam penelitian ini, diperkenalkan pendekatan inovatif dengan menggunakan *Convolutional Neural Network* (CNN) untuk mengatasi permasalahan tersebut. Dataset yang digunakan merupakan dataset gambar buah pepaya dalam berbagai tingkat kematangan. Dataset kemudian dilatih menggunakan model CNN untuk mengklasifikasikan buah pepaya berdasarkan warna kulit dan fitur visual lainnya. Hasil eksperimen menunjukkan bahwa model CNN yang digunakan mampu mengidentifikasi tingkat kematangan buah pepaya dengan tingkat akurasi yang tinggi, yaitu sebesar 96,63% sehingga dapat menghasilkan solusi yang andal dan efisien. Hal ini memberikan dampak yang signifikan dalam industri pertanian dan pengelolaan persediaan makanan.

Kata kunci: deteksi kematangan, pepaya, convolutional neural network

DETECTION OF PAPAYA FRUIT RIPENESS USING CONVOLUTIONAL NEURAL NETWORK MODEL

Abstract

Papaya is a fruit with many benefits and high nutritional value. 3,65 mg of vitamin A and 78 mg of vitamin C are found in every 100 grams of papaya. One of the advantages of papaya is its ability to bear fruit continuously, regardless of the season. The ripeness of papaya can be determined by the texture and color of its skin. The detection of papaya fruit ripeness levels holds significant importance in the agricultural industry and food inventory management. Conventional methods often face limitations in accuracy and efficiency when identifying the ripeness of papaya. In this research, an innovative approach is introduced using *Convolutional Neural Network* (CNN) to address this issue. The dataset used consists of images of papaya fruit at various ripeness levels. The dataset is then trained using the CNN model to classify papaya fruit based on skin color and other visual features. Experimental results show that the CNN model used can identify the ripeness level of papaya fruit with high accuracy, specifically 96,63%, thus providing a reliable and efficient solution. This has significant implications in the agricultural industry and food inventory management.

Keywords: ripeness detection, papaya, convolutional neural network

1. PENDAHULUAN

Pepaya adalah buah yang kaya akan manfaat. Terdapat 3,65 milligram vitamin A dan 78 milligram vitamin C dalam setiap 100 gram pepaya. Pepaya

dapat digunakan untuk membuat manisan, puding, dan jus. Seluruh bagian buah pepaya sangat bermanfaat bagi kehidupan manusia. Bahkan sebotol buah pepaya bisa dijadikan makanan atau pakan ternak. Pepaya juga banyak mengandung pektin

sehingga bisa dibuat menjadi selai dengan menambahkan gula pasir dan asam sitrat. Hasilnya selai yang enak, halus dan mengkilat (Wardani, Wijaya, & Bimantoro, 2022).

Salah satu keunggulan pepaya adalah kemampuannya untuk terus berbuah tanpa memperhatikan perubahan musim. Produksi pepaya terus meningkat dari tahun ke tahun, menurut data dari sistem informasi Badan Pusat Statistik Pertanian dan Kementerian Pertanian Republik Indonesia. Pada tahun 2018, produksi pepaya Indonesia sebanyak 887.580 ton, kemudian meningkat menjadi 986.991 ton pada tahun 2019 dan menjadi 1.016.388 ton pada tahun 2020. Produksi buah pepaya terbesar di Indonesia pada tahun 2020 adalah provinsi Jawa Timur (Agustina & Sukron, 2022).

Kematangan buah pepaya dapat terlihat dari perbedaan tekstur dan warna kulit buahnya. Dalam pengolahan citra, ada beberapa model analisis warna, salah satunya adalah *Hue, Saturation, Value* (HSV). Format gambar berwarna HSV adalah format yang paling umum digunakan dalam pemrosesan gambar. *Hue* menggambarkan warna asli, seperti merah, ungu, dan kuning, antara lain. *Saturation*, juga disebut kroma, adalah istilah yang mengacu pada intensitas atau kemurnian suatu warna. *Value* (Nilai) menunjukkan tingkat pencahayaan, tetapi dapat bervariasi antara 0% dan 100%. Misal nilainya diatur menjadi 0 (nol), maka warnanya menjadi hitam. Semakin tinggi nilainya, warnanya semakin terang, menciptakan corak baru (Masruroh, Sorikhi, & Syaqui, 2023).

Sebelumnya telah dilakukan penelitian untuk mengukur tingkat kematangan buah pepaya, yaitu Deteksi Kematangan Buah Pepaya Menggunakan Algoritma *You Only Look Once* (YOLO) oleh android. Penelitian tersebut menghasilkan akurasi sebesar 93%. Penulis mengatakan bahwa pada penelitian tersebut terdapat kendala yang mempengaruhi deteksi seperti pencahayaan yang terlalu gelap atau yang terlalu terang, melakukan deteksi dengan sudut kamera ke objek terlalu miring dan lain sebagainya (Agustina & Sukron, 2022). Kemudian, terdapat penelitian Implementasi Sistem Pendeteksi Tingkat Kematangan Buah Pepaya Menggunakan Metode RGB. Penelitian ini menggunakan metode deskriptif yang dimana permasalahan dibahas dengan menguraikan, membandingkan, hingga menarik kesimpulan. Penelitian ini menghasilkan akurasi sebesar 50%. Penulis juga menyarankan dalam proses pengambilan gambar diharapkan untuk memperhatikan posisi, pencahayaan, serta menambahkan lebih banyak data uji (Widyasari, Rosiani, & Pramudhita, 2021). Terakhir, terdapat penelitian Klasifikasi Tingkat Kematangan Buah Pepaya California Dalam Ruang Warna Hsv (Hue Saturation Value) Dengan Algoritma K-Nearest Neighbors. Hasil percobaan dan pengujian dari penelitian ini terhadap dataset 65 citra buah pepaya california yang diklasifikasikan

berdasarkan kematangan dengan algoritma K-NN menghasilkan akurasi sebesar 86,6667% dengan kriteria klasifikasi baik menggunakan matriks konfusi (Masruroh, Sorikhi, & Syaqui, 2023).

Selain penelitian yang bertujuan untuk mengukur tingkat kematangan buah pepaya, terdapat juga penelitian menggunakan metode pembelajaran *Deep Learning Convolutional Neural Network* (CNN) untuk mengembangkan sistem pengenalan kematangan buah jeruk. Keputusan untuk menggunakan CNN dalam penelitian ini didasarkan pada fakta bahwa CNN mudah digunakan, memiliki hasil yang kompetitif, dan memiliki parameter yang lebih sedikit untuk diperhatikan (Wan Ng, 2018). Penelitian ini menghasilkan daftar nama model jeruk yang ditemukan melalui pengenalan dan klasifikasi tingkat kematangan buah jeruk (Christian & Al Idrus, 2023).

Selain diterapkan pada buah jeruk, pada penelitian lain metode CNN digunakan untuk mengklasifikasikan tingkat kematangan buah mangga badami. Penelitian ini berusaha untuk mengetahui kualitas mangga badami dengan memilih atau memilah buah mentah. Buah mentah yang dibuat secara konvensional mungkin tidak benar-benar matang atau sudah melewati batas kematangan, yang berarti mereka akan membusuk atau tidak layak dimakan. Sebelum percobaan dimulai, penelitian tersebut melakukan beberapa langkah. Langkah penelitian tersebut meliputi pengumpulan data mangga badami, pra-proses gambar atau citra, dan klasifikasi, dan evaluasi hasil penelitian. Hasil penelitian menunjukkan bahwa penggunaan model CNN cukup baik untuk mengklasifikasikan kematangan buah mangga badami. Pengujian terhadap data pengujian (*testing*) menunjukkan nilai akurasi sebesar ~97,2% dan nilai *loss* sebesar ~0,5%, sedangkan pengujian terhadap data pelatihan (*training*) menghasilkan nilai akurasi sebesar ~94,6% dan nilai *loss* sebesar ~0,1%. (Damayanti, Arkadia, & Prasvita, 2021).

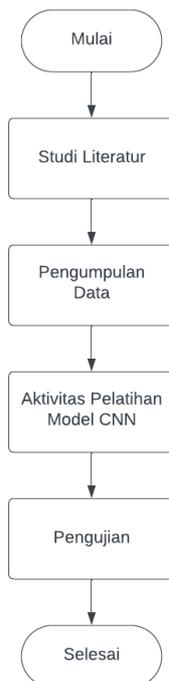
Berdasarkan paparan di atas, menjadi latar belakang penulis melakukan penelitian tingkat kematangan buah pepaya dengan menggunakan model *Convolutional Neural Network* (CNN). Dibandingkan dengan penelitian sebelumnya dengan objek yang sama yakni buah pepaya yang menggunakan pendekatan lain seperti algoritma K-NN, YOLO, dan metode RGB, penelitian yang dilakukan oleh penulis berfokus pada penerapan model CNN yang merupakan pendekatan yang lebih terkini dalam banyak tugas pengenalan gambar. Diharapkan bahwa penelitian ini akan menghasilkan klasifikasi buah pepaya dengan tingkat akurasi yang lebih tinggi dan dapat mendeteksi tingkat kematangan buah pepaya sesuai dengan perkiraan yang dibuat berdasarkan klasifikasi kematangan buah pepaya.

2 METODE PENELITIAN

2.1 Alat dan Bahan

Penerapan metode klasifikasi dan evaluasi akan dilakukan dengan menggunakan platform Google Colab dengan bahasa pemrograman Python. Hardware yang digunakan dalam melakukan penelitian adalah Macbook Air 2020 dengan RAM 8 GB. *Framework* yang digunakan pada penelitian ini adalah Tensorflow. Tensorflow menyediakan seperangkat alat yang memungkinkan pengguna untuk dengan mudah membuat dan melatih model *deep learning* yang kompleks. *Library* yang digunakan yakni Keras, Sklearn, Matplotlib, Numpy, dan Seaborn. Keras adalah *library deep learning* yang ditulis dalam bahasa Python. Untuk mempercepat penelitian, Keras dapat dijalankan dengan menggunakan Tensorflow yang memang dirancang khusus untuk *Deep Learning* (Alfikri, et al., 2022). Sklearn adalah *library* yang menyediakan beragam algoritma *machine learning* yang dapat digunakan untuk tugas seperti klasifikasi, regresi, dan *clustering*. Matplotlib adalah sebuah *library* Python yang digunakan untuk membuat berbagai jenis visualisasi data yang memiliki kemampuan untuk membuat berbagai jenis plot seperti plot garis, *scatter plot*, *bar plot*, *histogram*. Numpy adalah *library* Python yang digunakan untuk melakukan operasi numerik dan matematika pada *array* dan matriks (Gunawan, 2023). Seaborn adalah pustaka visualisasi data Python yang dibuat berdasarkan matplotlib (Waskom, 2021).

2.2 Proses Penelitian



Gambar 1. Diagram alir rancangan penelitian

Gambar 1 merupakan alur rancangan penelitian deteksi tingkat kematangan buah pepaya.

Penelitian tersebut mencakup Studi literatur, Pengumpulan data, Aktivitas pelatihan model CNN, dan Pengujian.

2.2.1 Studi Literatur

Studi literatur dilakukan dengan tujuan untuk mengumpulkan ide teori yang berkaitan dengan subjek penelitian. Studi literatur merupakan komponen penting dalam tahap perencanaan penelitian yang berfokus pada ekstraksi menggunakan metode *deep learning* pada tahapan *feature learning* dan deteksi citra menggunakan metode CNN. Dengan melakukan studi literatur, peneliti dapat memahami landasan teoritis yang kuat yang mendukung pengembangan metode tersebut.

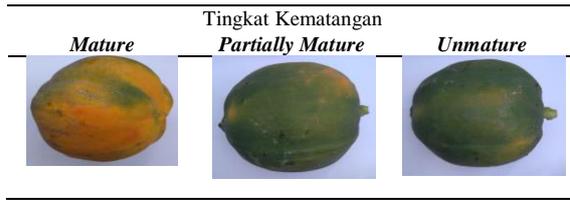
Selain itu, dalam upaya ini, referensi dari buku, jurnal, artikel *online*, dan sumber-sumber relevan akan digunakan sebagai bahan acuan kunci untuk memahami konsep yang terkait dengan pemrosesan citra, pengenalan pola, dan teknik *deep learning* yang mendasari penelitian ini. Dengan cara ini, studi literatur akan memastikan bahwa penelitian berada pada garis depan pengetahuan dan dapat menghadirkan kontribusi yang signifikan dalam pemecahan masalah yang dituju.

2.2.2 Pengumpulan Data

Pada penelitian yang dilakukan, akan menggunakan *dataset* yang berisi kumpulan gambar pepaya yaitu *Papaya Classification* yang tersedia di Kaggle (Shahane, 2020). *Dataset* buah pepaya dapat diakses melalui <https://www.kaggle.com/datasets/saurabhshahane/papaya-classification>.

Tabel 1. Contoh Buah Pepaya

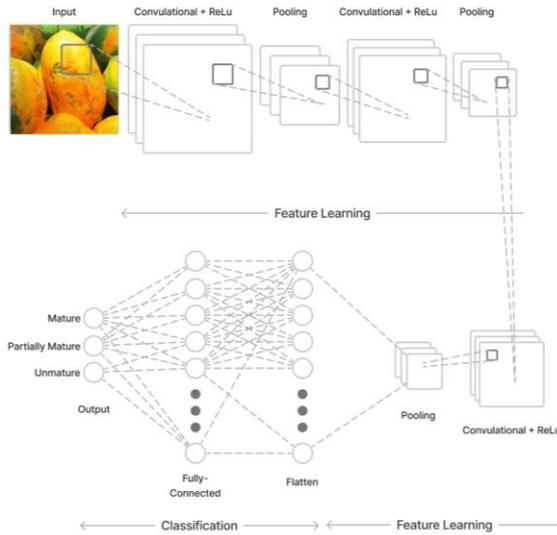
Tingkat Kematangan		
<i>Mature</i>	<i>Partially Mature</i>	<i>Unmature</i>
		
		
		
		



Pada tabel 1 merupakan contoh dari data gambar yang digunakan. Dataset tersebut memiliki 300 gambar, yang terdiri dari 100 gambar matang (*mature*), 100 gambar setengah matang (*partially mature*), dan 100 gambar mentah (*unmature*). Total data gambar yang diolah yaitu sebanyak 300 gambar. Dalam penelitian selanjutnya akan dibagi lagi sebagai data *training* dan data *test*, dengan 240 sebagai gambar *train* dan 60 sebagai gambar *test*.

2.2.3 Model CNN

Tahap ketiga yaitu memulai pembuatan model. Pada penelitian ini, model yang dibuat menggunakan layer *Convolutional*, *Pooling*, *Flatten*, dan *Fully Connected*. Berikut merupakan gambaran model CNN yang akan digunakan dalam melakukan deteksi kematangan buah pepaya.



Gambar 2. Ilustrasi Lapisan Model CNN

Gambar 2 di atas merupakan ilustrasi dari lapisan atau *layer* yang digunakan model CNN yang digunakan pada penelitian ini. Ilustrasi tersebut menggunakan 3 lapisan *layer* CNN, 1 *layer* *flatten*, dan 1 *layer* *fully connected*.

Selanjutnya dilakukan perhitungan parameter pada setiap layernya. Untuk menghitung parameter tersebut, dapat menggunakan rumus matematika berikut.

- Conv/2D Layer

$$Parameter = (D_f \times D_f \times C_{in} + 1) \times F$$

- MaxPooling2D Layer

$$Parameter = 0$$

- Flatten Layer

$$Parameter = 0$$

- Dense Layer

$$Parameter = (N_{neuron\ input} + 1) \times N_{neuron\ output}$$

Pada penelitian ini akan menggunakan model yang bervariasi. Model yang dibuat memiliki 3 variasi dengan jumlah *layer* *Convolution* yang berbeda-beda.

Tabel 2. Model dengan 1 layer Convolutional

Layer	Output Shape	Parameter
Input size	150x150x3	0
conv2d (Conv2D)	(None, 148, 148, 32)	896
max_pooling2d (MaxPooling2D)	(None, 74, 74, 32)	0
flatten (Flatten)	(None, 175.232)	0
dense (Dense)	(None, 128)	22.429.824
dense_1 (Dense)	(None, 3)	387
Total parameter: 22.431.107		

Berdasarkan tabel 2, model CNN pada penelitian ini dibuat dengan menggunakan total 4 *layer* yang terdiri dari 1 *layer* *convolutional*, 1 *layer* *pooling*, 1 *layer* *flatten*, dan *layer* *fully connected*. Total parameter dalam model ini sebanyak 22.431.107.

Tabel 3. Model dengan 2 layer Convolutional

Layer	Output Shape	Parameter
Input size	150x150x3	0
conv2d (Conv2D)	(None, 148, 148, 32)	896
max_pooling2d (MaxPooling2D)	(None, 74, 74, 32)	0
conv2d_1 (Conv2D)	(None, 72, 72, 128)	36.992
max_pooling2d_1 (MaxPooling2D)	(None, 36, 36, 128)	0
flatten (Flatten)	(None, 165.888)	0
dense (Dense)	(None, 128)	21.233.792
dense_1 (Dense)	(None, 3)	387
Total parameter: 21.272.067		

Pada tabel 3, model CNN pada penelitian ini dibuat dengan menggunakan total 6 *layer* yang terdiri dari 2 *layer* *convolutional*, 2 *layer* *pooling*, 1 *layer* *flatten*, dan *layer* *fully connected*. Total parameter dalam model ini sebanyak 21.272.067.

Tabel 4. Model dengan 3 layer Convolutional

Layer	Output Shape	Parameter
Input size	150x150x3	0
conv2d (Conv2D)	(None, 148, 148, 32)	896
max_pooling2d (MaxPooling2D)	(None, 74, 74, 32)	0
conv2d_1 (Conv2D)	(None, 72, 72, 64)	18.496
max_pooling2d_1 (MaxPooling2D)	(None, 36, 36, 64)	0
conv2d_2 (Conv2D)	(None, 34, 34, 128)	73.856
max_pooling2d_2 (MaxPooling2D)	(None, 17, 17, 128)	0
flatten (Flatten)	(None, 36.992)	0
dense (Dense)	(None, 128)	4.735.104
dense_1 (Dense)	(None, 3)	387
Total parameter: 4.828.739		

Berdasarkan tabel 4, model CNN pada penelitian ini dibuat dengan menggunakan total 8 layer yang terdiri dari 3 layer convolutional, 3 layer pooling, 1 layer flatten, dan layer fully connected. Total parameter dalam model ini sebanyak 4.828.739.

2.2.4 Pengujian

Untuk mengukur performa modelnya, maka peneliti menggunakan confusion matrix. Confusion matrix adalah teknik yang digunakan untuk menghitung Recall, Precision, dan F1-score yang berguna dalam menilai performa model klasifikasi (Gunawan, 2023).

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

Gambar 3. Confusion Matrix

Gambar 3 merupakan contoh confusion matrix dengan keterangan sebagai berikut:

- TP : Kelas positif yang diprediksi positif.
- TN : Kelas positif yang diprediksi negatif.
- FP : Kelas negatif yang diprediksi positif.
- FN :Kelas negatif yang diprediksi negatif.

Berikut merupakan rumus untuk mencari nilai dalam metode confusion matrix.

- Recall merupakan proporsi kasus positif yang secara aktual teridentifikasi. Rumus untuk mengetahui recall yaitu:

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

- Precision merupakan rasio antara kasus yang diidentifikasi sebagai positif dengan benar dibandingkan dengan keseluruhan kasus yang diprediksi sebagai positif. Rumus dari Precision yaitu:

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

- Accuracy adalah perbandingan antara jumlah kasus yang diidentifikasi dengan benar dan keseluruhan jumlah kasus. Rumus dari accuracy yaitu:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

- F1-score didapatkan dari rata-rata presisi dan recall yang harmonis. Metrik F1-score digunakan saat Anda mencari keseimbangan antara presisi dan perolehan. Rumus dari F1-score yaitu:

$$F1 - score = \frac{2 * Precision * Recall}{Precision + Recall}$$

Nilai akurasi dibutuhkan untuk menentukan apakah model tersebut sudah akurat atau belum. Untuk penelitian ini, peneliti menetapkan target akurasi sebesar $\geq 90\%$ sehingga jika belum mencapai target maka peneliti akan memperbaiki kembali model yang telah dirancang, dan kemudian melakukan proses training lagi.

3. TINJAUAN PUSTAKA DAN TEORI

3.1 Klasifikasi

Klasifikasi melibatkan proses identifikasi pola atau fungsi yang dapat digunakan untuk membedakan antara kelas yang berbeda. Tujuan utama klasifikasi adalah untuk menentukan kelas suatu objek yang belum diketahui kelasnya. (Han & Kamber, 2012). Terdapat dua proses dalam klasifikasi data, yaitu proses training dan proses testing. Proses training merupakan proses pelatihan data yang digunakan dalam membuat prediksi. Proses testing adalah proses pengujian dataset untuk menentukan tingkat akurasi dari model tersebut.

3.2 Convolutional Neural Network (CNN)

Pada tahun 1989, LeCun menerapkan backpropagation pada lapisan neural konvolusional yang berbagi bobot dan mirip dengan Neocognitron dengan koneksi adaptif. Kombinasi ini, ditambah dengan Max-Pooling, dan dipercepat pada kartu grafis, telah menjadi bahan penting dari Deep Learner visual feedforward modern yang banyak memenangkan kompetisi. Karyanya ini juga memperkenalkan kumpulan data MNIST dari digit tulisan tangan yang seiring waktu telah menjadi benchmark pada dunia pembelajaran mesin yang paling terkenal (Schmidhuber, 2014). Salah satu algoritma Deep Learning yang dikembangkan dari Multilayer Perceptron (MLP) adalah Combined Convolutional Neural Network (CNN) (Rahmadhani & Marpaung, 2023). Algoritma ini dirancang untuk mengklasifikasikan data citra berupa gambar atau suara dengan metode supervised learning.

CNN dapat memproses data yang berbentuk beberapa array, salah satu contohnya adalah gambar berwarna yang terdiri dari tiga array 2D yang berisi intensitas piksel dalam tiga saluran warna. Berbagai modalitas data berbentuk beberapa array, seperti 1D yang diterapkan pada sinyal dan urutan, termasuk bahasa, 2D pada gambar atau spectrogram audio, dan 3D pada video atau gambar volumetrik. Memanfaatkan karakteristik sinyal alami (natural signals), CNN didasarkan pada empat konsep utama, antara lain koneksi lokal (local connections), bobot bersama (shared weights), penggabungan (pooling), dan penggunaan banyak lapisan. Sejak awal tahun 2000-an, CNN telah diterapkan dengan sukses besar untuk deteksi, segmentasi, dan pengenalan objek dan wilayah dalam gambar (LeCun, Bengio, & Hinton, 2015).

Terdapat dua bagian utama dalam tahapan proses CNN yaitu feature learning dan classification.

Untuk menyelesaikan kasus penelitian ini, peneliti membuat model *machine learning* dengan bantuan *library* keras, tensorflow, numpy, sklearn, matplotlib, dan seaborn. Model tersebut dibuat menggunakan algoritma CNN dengan total 8 *layer* yang terdiri dari 3 *layer convolutional*, 3 *layer pooling*, 1 *layer flatten*, dan 1 *layer fully connected*.

3.2.1 Feature Learning

Feature Learning ialah proses *encoding* dari sebuah gambar menjadi fitur berupa nilai yang mencerminkan gambar tersebut. Proses ini melibatkan beberapa lapisan yang saling bekerja sama untuk mengambil ciri dari sebuah gambar (Ramba, 2020).

1. Convolution Layer

Convolutional Layer ialah lapisan pertama dalam arsitektur CNN yang digunakan untuk memproses masukan citra pada sistem. Lapisan ini menerima masukan citra dan diproses menjadi keluaran berupa vektor. Proses yang dilakukan yaitu operasi konvolusi yang akan mengekstraksi fitur dari masukan citra dan diolah hingga menghasilkan keluaran berupa *feature map* (Ilahiyah & Nilogiri, 2018).

2. Pooling Layer

Pooling layer merupakan bagian penting dari proses pembelajaran fitur pada CNN yang bertugas untuk mengurangi dimensi dari matriks hasil konvolusi. Tujuannya adalah untuk mempercepat proses komputasi dengan mengurangi jumlah parameter yang perlu diperbarui, mengurangi risiko *overfitting*. Seperti halnya lapisan konvolusi, *pooling layer* juga menggunakan filter dengan ukuran tertentu untuk melakukan operasi *sliding window* pada matriks input. Dua jenis *pooling* umum yang digunakan adalah *max pooling* dan *average pooling*. *Max pooling* mengambil nilai terbesar dalam cakupan filter dari matriks input, sementara *average pooling* mengambil nilai rata-rata dalam cakupan filternya. (Ramba, 2020).

3.2.2 Klasifikasi

Proses klasifikasi adalah proses untuk menentukan kelas atau kategori dari setiap neuron yang telah diidentifikasi pada proses ekstraksi fitur. Bagian ini terdiri dari beberapa lapisan yang saling berhubungan satu sama lain (Ramba, 2020). Berikut penjelasan terkait bagian tersebut.

1. Flatten

Proses ekstraksi fitur menghasilkan *feature map* yang berbentuk multidimensional array. Namun, *fully connected layer* hanya menerima input berupa vektor. Oleh karena itu, diperlukan fungsi untuk mengubah multidimensional array menjadi vektor. *Flatten* adalah fungsi yang berfungsi untuk mengubah multidimensional array menjadi vektor. Tujuannya adalah agar nilai

tersebut dapat digunakan sebagai input pada *fully connected layer* (Ramba, 2020).

2. Fully Connected Layer

Fully connected layer adalah bagian dari *forward neural networks* yang meliputi lapisan tersembunyi, fungsi aktivasi, lapisan output, dan *loss function*. Biasanya, *fully connected layer* digunakan dalam *multi layer perceptron* untuk mentransformasi dimensi data agar dapat diolah secara linier. Lapisan ini menerima input dari hasil pembelajaran fitur. Input tersebut dalam bentuk vektor setelah melalui proses *flatten* sebelumnya, seperti yang dijelaskan oleh Ramba pada tahun 2020.

3.3 Penelitian Terdahulu

Telah dilakukan studi pada penelitian terdahulu mengenai pendeteksian warna kematangan buah pepaya. Dalam penelitian yang berjudul *Klasifikasi Tingkat Kematangan Buah Pepaya California Dalam Ruang Warna Hsv (Hue Saturation Value) Dengan Algoritma K-Nearest Neighbors*, menghasilkan sebuah kesimpulan yaitu hasil percobaan dan pengujian terhadap dataset 65 citra buah pepaya california yang diklasifikasikan berdasarkan kematangan dengan algoritma K-NN memberikan akurasi sebesar 86,6667% dengan kriteria klasifikasi baik menggunakan matriks konfusi. Guna meningkatkan kinerja dan melengkapi penelitian yang telah ada, penulis memberikan beberapa saran diantaranya untuk penelitian selanjutnya disarankan untuk memperbanyak data yang digunakan untuk pengujian, agar hasil pengujian yang dilakukan lebih lengkap. Kemudian, ekstraksi dengan metode lain memberikan hasil prediksi yang lebih baik. Ekstraksi fitur HSV disini dapat dikombinasikan dengan model lain, seperti GLCM karena GLCM memiliki tingkat deteksi yang tinggi berdasarkan nilai kontras, korelasi, homogenitas dan energi (Masruroh, Sorikhi, & Syauqi, 2023). Pada sebuah penelitian lain yang menggunakan sensor warna TCS3200 untuk klasifikasi tingkat kematangan buah pepaya berdasarkan warna kulit menghasilkan beberapa kesimpulan, antara lain:

- 1) Hasil pengujian sensor TCS3200 menunjukkan bahwa sensor dapat mendeteksi hingga 100 nilai RGB.
- 2) Algoritma *K-Nearest Neighbour* bekerja dengan baik.
- 3) Hasil dari proses klasifikasi kematangan buah pepaya menggunakan algoritma *K-Nearest Neighbor* menunjukkan bahwa nilai $K=15$ memiliki persentase keberhasilan tertinggi, yaitu 93%. Oleh karena itu, nilai ini dapat digunakan sebagai nilai acuan prediksi (Jayadi & Meilinda, 2023).

4 HASIL DAN PEMBAHASAN

Penelitian ini menggunakan Google Colab untuk mempermudah kolaborasi antar anggota tim

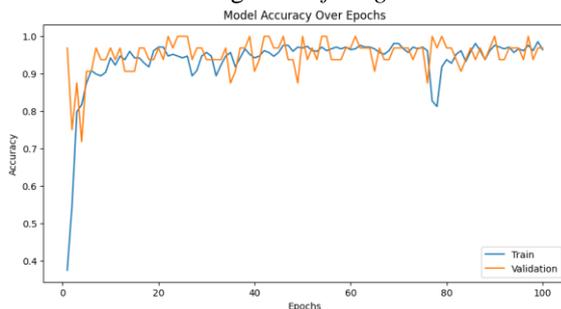
dalam melakukan pemrograman. *Library* Keras diimpor melalui TensorFlow dalam membangun model jaringan saraf tiruan atau *neural network* untuk mengklasifikasikan buah pepaya ke dalam tiga kategori, yakni mentah, setengah matang, dan matang.

Awalnya dilakukan persiapan data dengan memisahkan gambar buah pepaya dari *dataset* "Papaya Classification" dari Kaggle oleh Saurabh Shahane agar sesuai dengan tiga kategori yang sudah disebutkan sebelumnya. Selanjutnya dilakukan pembagian data atau *split data* secara acak dengan menggunakan 80% data untuk pelatihan (*train data*) dan 20% untuk pengujian (*test data*).

Selanjutnya dilakukan pembuatan 3 model yang berbeda. Pada model pertama dilakukan pembuatan model dengan menggunakan 1 *layer convolutional*, model kedua menggunakan 2 *layer convolutional*, dan model ketiga menggunakan 3 *layer convolutional*.

Setelah model CNN dibuat, dilakukan proses kompilasi dengan menggunakan *optimizer* yang berbeda-beda. Pada percobaan ini, terdapat 3 *optimizer* yang digunakan yakni Adam atau *Adaptive Moment Estimation*, Adamax, dan Nadam atau *Adam with Nesterov momentum*. Selanjutnya nilai *learning rate* diatur sebesar 0,001. Selain itu terdapat fungsi kerugian (*loss function*) "categorical_crossentropy" dan metrik "accuracy." untuk menentukan cara model akan mempelajari pola data.

Data pelatihan diperkuat (*augmented*) menggunakan *ImageDataGenerator* yang melibatkan perubahan data pelatihan seperti rotasi, pergeseran, perataan, dan pencerminan untuk menciptakan variasi dalam *training data*. Pada penelitian ini, peneliti menggunakan *rescale* sebesar 1,00/255,00 yang artinya mengubah intensitas piksel dalam gambar menjadi rentang 0 hingga 1 dengan membagi setiap nilai piksel oleh 255. Kemudian melakukan rotasi gambar sebesar 40 derajat. Melakukan pergeseran gambar horizontal dan vertikal hingga 20% dari lebar dan tinggi gambar asli. Memungkinkan peregangan gambar 20%. Memungkinkan *zoom in* atau *out* pada gambar hingga 20%. Memungkinkan *flip* horizontal pada gambar. Terakhir terdapat metode yang digunakan untuk mengisi piksel yang mungkin kosong setelah transformasi dengan piksel yang terdekat. Hal ini membantu model belajar dengan lebih baik dan mencegah *overfitting*.



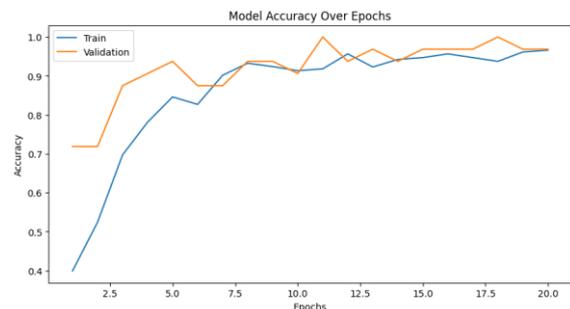
Gambar 4. Akurasi Model 3 Layer dengan 100 Epoch

Pada gambar 4, peneliti melakukan percobaan melatih model menggunakan 3 layer dengan jumlah epoch sebanyak 100. Model tersebut memiliki akurasi yang cenderung diatas 90%. Maka dari itu, peneliti memutuskan untuk melakukan *training data* dengan 20 *epoch* untuk menghindari terjadinya *overfitting* pada model dan mempersingkat waktu pada proses *training*. Pada setiap *epoch*, model memproses *training data* dengan memperbarui bobot (*weights*) dengan informasi yang didapatkan dari kumpulan data pelatihan untuk mengenali pola-pola yang ada.

Tabel 5. Hasil Training

Optimizer	Akurasi		
	1 Layer	2 Layer	3 Layer
Adam	95,19%	93,75%	94,23%
Adamax	95,19%	95,67%	94,23%
Nadam	91,83%	96,63%	96,63%

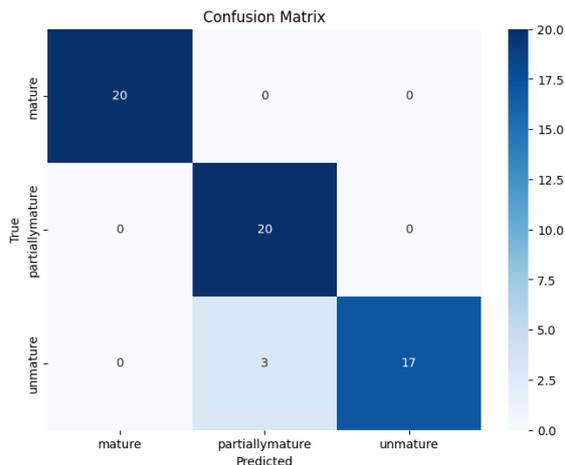
Berdasarkan tabel 5, hasil pelatihannya menghasilkan akurasi sekitar 91,83% hingga 96,63% yang menunjukkan kemampuan model CNN yang dibuat memiliki tingkat keakuratan yang tinggi dalam mengklasifikasikan gambar pepaya. Model yang menghasilkan akurasi tertinggi yakni 96,63% yaitu model yang dibangun dengan menggunakan *optimizer* nadam dengan jumlah *layer cnn* sebanyak 2 dan 3.



Gambar 5. Akurasi Model 3 Layer dengan Optimizer Nadam

Pada Gambar 5, penulis melakukan visualisasi hasil akurasi model tertinggi untuk setiap *epoch*-nya. Model yang telah dibuat mengalami *overfitting*. Hal ini dikarenakan pada grafik tersebut terlihat bahwa pada *epoch* ke 7, data pelatihan mengalami peningkatan sedangkan untuk data validasi mengalami penurunan. Selain itu, pada *epoch* ke 11 dan 14, data pelatihan mengalami peningkatan yang stabil sedangkan untuk data validasi mengalami penurunan yang signifikan.

Penelitian ini dilakukan pengujian lebih lanjut dengan menguji model pada gambar yang dimuat untuk mengevaluasi sejauh mana model mampu memprediksi kategori kematangan buah pepaya. Penelitian menggunakan 60 buah gambar dengan 20 gambar matang, 20 gambar setengah matang, dan 20 gambar mentah.



Gambar 6. Hasil Confusion Matrix

Pada gambar 6, penelitian melakukan evaluasi model dengan menggunakan *confusion matrix*. Model dapat memprediksi benar 20 gambar matang, 20 gambar setengah matang, dan 17 gambar mentah.



Gambar 7. Contoh Gambar Prediksi Pepaya Benar



Gambar 8. Contoh Gambar Prediksi Pepaya Salah

Berdasarkan hasil pengujian model yang telah dilakukan, didapatkan bahwa gambar 7 merupakan hasil prediksi yang benar yang dilakukan oleh model yang telah dibuat. Lalu terdapat 3 gambar yang menghasilkan prediksi yang salah, prediksi yang salah tersebut dapat dilihat pada gambar 8. Model tersebut gagal untuk melakukan beberapa prediksi gambar dengan tingkat kematangan mentah.

Tabel 6. *Classification Report Model*

	Precision	Recall	F1-score	Support
Mature	1,00	1,00	1,00	20
Partially mature	0,87	1,00	0,93	20
Unmature	1,00	0,85	0,92	20
Accuracy Macro avg	0,96	0,95	0,95	60
Weighted avg	0,96	0,95	0,95	60

Pada tabel 6, hasil pengujian ini berupa *classification report* yang menunjukkan tingkat

akurasi sebesar 0,95 yang berarti bahwa 95% dari semua klasifikasi adalah benar. *Precision* untuk kelas Mature adalah 1,00 yang berarti bahwa semua klasifikasi yang diprediksi sebagai Mature benar-benar Mature. *Precision* untuk kelas Partially Mature adalah 0,87 yang berarti bahwa 87% dari klasifikasi yang diprediksi sebagai Partially Mature benar-benar Partially Mature. *Precision* untuk kelas Unmature adalah 1,00 yang berarti bahwa semua klasifikasi yang diprediksi sebagai Unmature benar-benar Unmature. *Recall* untuk kelas Mature adalah 1,00 yang berarti bahwa semua sampel Mature diprediksi sebagai Mature. *Recall* untuk kelas Partially Mature adalah 1,00 yang berarti bahwa semua sampel Partially Mature diprediksi sebagai Partially Mature. *Recall* untuk kelas Unmature adalah 0,85 yang berarti bahwa 85% dari sampel Unmature diprediksi sebagai Unmature. *F1-score* untuk kelas Mature adalah 1,00 yang berarti bahwa model klasifikasi bekerja sangat baik untuk mengidentifikasi sampel Mature. *F1-score* untuk kelas Partially Mature adalah 0,93 yang berarti bahwa model klasifikasi bekerja dengan baik untuk mengidentifikasi sampel Partially Mature. *F1-score* untuk kelas Unmature adalah 0,92 yang berarti bahwa model klasifikasi bekerja dengan baik untuk mengidentifikasi sampel Unmature.

Penelitian CNN mendapatkan akurasi tertinggi sebesar 96,63%. Selanjutnya dilakukan perbandingan dengan penelitian terdahulu.

1. 3,63% lebih tinggi dibandingkan dengan algoritma YOLO.
2. 46,63% lebih tinggi dibandingkan dengan metode RGB.
3. 10% lebih tinggi dibandingkan dengan metode HSV.

5 KESIMPULAN

Sebagai kesimpulan dari hasil penelitian yang mendeteksi kematangan buah pepaya yang dibahas pada bab sebelumnya, dapat disimpulkan bahwa:

1. Pada studi kasus ini, proses mendeteksi tingkat kematangan buah pepaya dengan menggunakan CNN menghasilkan akurasi yang baik.
2. Secara keseluruhan, model yang dibuat menggunakan 3 *layer convolutional*, 3 *layer pooling*, 1 *layer flatten*, dan *layer fully connected* memiliki tingkat akurasi yang baik.
3. Dalam hal mendeteksi tingkat kematangan buah pepaya, *optimizer* Nadam lebih akurat daripada *optimizer* Adam dan Adamax.
4. Pelatihan model menggunakan *training* data dengan 20 *epoch* menghasilkan akurasi pengujian maksimal sebesar 96,63% yang menunjukkan kemampuan model CNN yang dibuat memiliki tingkat keakuratan yang tinggi dalam mengklasifikasikan gambar-gambar pepaya.

Saran dalam pengembangan kedepannya dari peneliti, yaitu:

1. Penelitian ini hanya menggunakan 300 *dataset*, sedangkan penelitian ini menggunakan metode

- Convolutional Neural Network* yang dimana metode tersebut umumnya harus menggunakan puluhan ribu *dataset*. Pengembang kedepannya diharap untuk menambah *dataset* agar tingkat akurasinya meningkat.
- Model yang dibuat masih mengalami *overfitting*. Hal ini karena *dataset* yang digunakan hanyalah sebanyak 240 *dataset* untuk proses *training*. Pengembang kedepannya diharapkan membuat model tanpa mengalami *overfitting*.
 - Penelitian ini menggunakan metode *Convolutional Neural Network*. Pengembang kedepannya diharap untuk melakukan penelitian dengan metode yang lebih baru lagi.
- ### DAFTAR PUSTAKA
- ALFIKRI, R.H., UTOMO, M.S., FEBRUARIYANTI, H., dan NURWAHYUDI, E., 2022. Pembangunan Aplikasi Penerjemah Bahasa Isyarat Dengan Metode Cnn Berbasis Android. *Jurnal Teknoinfo*, 16(2), pp.183-197.
- AGUSTINA, F., dan SUKRON, M., 2022. Deteksi Kematangan Buah Pepaya Menggunakan Algoritma YOLO Berbasis Android. *Jurnal Ilmiah Infokam*, 18(2), pp.70-78.
- BADAN PUSAT STATISTIK, 2023. Produksi Tanaman Buah-Buahan: Pengumpulan data hortikultura, [online] Tersedia di: <<https://www.bps.go.id/indicator/55/62/1/produksi-tanaman-buah-buahan.html>> [Diakses 15 September 2023]
- CHRISTIAN, J., dan AL IDRUS, S.I., 2023. Introduction to Citrus Fruit Ripens Using the Deep Learning Convolutional Neural Network (CNN) Learning Method. *Asian Journal of Applied Education (AJAE)*, 2(3), pp.459-470.
- CHOLLET, F., 2021. *Deep learning with Python*. Simon and Schuster.
- DAMAYANTI, S.A., ARKADIA, A., dan PRASVITA, D.S., 2021. Klasifikasi Buah Mangga Badami Untuk Menentukan Tingkat Kematangan dengan Metode CNN. In *Prosiding Seminar Nasional Mahasiswa Bidang Ilmu Komputer dan Aplikasinya*, 2(2), pp.158-165.
- GUNAWAN, B.T., 2023. Klasifikasi Untuk Menentukan Kematangan Buah Pisang Menggunakan Algoritma Convolutional Neural Network. *Doctoral dissertation*. Institut Sains & Teknologi AKPRIND Yogyakarta.
- HAN, J., KAMBER, M., dan PEI, J., 2012. *Data Mining Concepts and Techniques*. 3rd Edition. Waltham, USA: Morgan Kaufmann Publishers.
- ILAHYAH, S., dan NILOGIRI, A., 2018. Implementasi Deep Learning Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan Convolutional Neural Network. *JUSTINDO (Jurnal Sistem Dan Teknologi Informasi Indonesia)*, 3(2), pp.49-56.
- JAYADI, A., dan MEILINDA, D., 2023. Klasifikasi Tingkat Kematangan Buah Pepaya Berdasarkan Warna Kulit Menggunakan Sensor Warna Tcs3200, 3(2), pp.1-13.
- LECUN, Y., BENGIO, Y., dan HINTON, G., 2015. *Deep learning*. *Nature*, 521(7553), pp.436-444.
- MASRUROH, A.I., SORIKHI, dan SYAUQI A., 2023. Klasifikasi Tingkat Kematangan Buah Pepaya California Dalam Ruang Warna HSV (Hue Saturation Value) Dengan Algoritma K-Nearest Neighbors. *Jurnal Informatika dan Riset (IRIS)*.
- RAHMADHANI, U.S., dan MARPAUNG, N.L., 2023. Klasifikasi Jamur Berdasarkan Genus dengan Menggunakan Metode CNN. *Jurnal Informatika: Jurnal Pengembangan IT*, 8(2), pp.169-173.
- RAMBA, L.S., 2020. Perancangan Sistem Home Automation Dengan Kendali Perintah Suara Menggunakan Deep Learning Convolutional Neural Network (DI-Cnn). *Doctoral dissertation*. Universitas Komputer Indonesia.
- SHAHANE, S., 2020. *Papaya Classification*, [online] Tersedia di: <<https://www.kaggle.com/datasets/saurabhshahane/papaya-classification?resource=download>> [Diakses 5 Oktober 2023]
- SCHMIDHUBER, J., 2014. Deep learning in neural networks: An overview. *Neural networks*, 61, pp.85-117.
- WAN NG., 2018. *New digital technology in education*. Switzerland: Springer.
- WARDANI, L.A., WIJAYA, I.G.P.S., dan BIMANTORO, F., 2022. Klasifikasi Jenis Dan Tingkat Kematangan Buah Pepaya Berdasarkan Fitur Warna, Tekstur Dan Bentuk Menggunakan Support Vector Machine. *Jurnal Teknologi Informasi, Komputer, dan Aplikasinya (JTIKA)*, 4(1), pp.75-87.
- WIDYASARI, K.B.D.R.N., ROSIANI, U.D., dan PRAMUDHITA, A.N., 2021. Implementasi Sistem Pendeteksi Tingkat Kematangan Buah Pepaya Menggunakan Metode RGB. *SMATIKA JURNAL: STIKI Informatika Jurnal*, 11(01), pp.32-36.
- WASKOM, M.L., 2021. Seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60), p.3021.
- ZACCONE, G., dan KARIM, M.R., 2018. *Deep Learning with TensorFlow: Explore neural networks and build intelligent systems with Python*. Packt Publishing Ltd.

Halaman ini sengaja dikosongkan.