

Penggunaan Deep Learning dengan Convolutional Neural Network untuk Klasifikasi Kondisi Buah Tomat

Muhammad Ferry Saputra^{a1}, I Gusti Agung Gede Arya Kadyanan^{a2}

^aProgram Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,
Universitas Udayana
Jalan Raya Kampus Udayana, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia
¹saputra2308561109.student.unud.ac.id
²gungde.unud.ac.id

Abstract

The quality and ripeness of tomatoes are crucial factors determining their market value and shelf life. This poses a significant challenge for farmers, who often rely on subjective manual sorting. This research proposes an automated solution using deep learning to improve the efficiency and objectivity of sorting. We developed a Convolutional Neural Network (CNN) model to classify tomatoes into three distinct categories: ripe, unripe, and rotten. The dataset used was sourced from Kaggle, consisting of 300 training images and 60 testing images, with a balanced distribution across the three classes. The CNN architecture was optimized using the Random Search method on a predefined hyperparameter space. The best-performing model achieved an overall accuracy of 80% on the test data. Confusion matrix analysis demonstrated excellent performance in identifying ripe (100% recall) and unripe (95% recall) tomatoes, although it faced challenges in correctly identifying rotten tomatoes (45% recall). These results demonstrate the great potential of CNN-based systems as accessible and effective tools for farmers to automate quality control, reduce post-harvest losses, and increase overall productivity.

Keywords: Deep Learning, Convolutional Neural Network, Image Classification, Tomato Ripeness, Hyperparameter Tuning.

1. Pendahuluan

Tomat (*Solanum lycopersicum*) merupakan salah satu komoditas hortikultura yang paling vital di dunia, baik untuk konsumsi langsung maupun sebagai bahan baku industri pengolahan makanan. Kualitas buah tomat, yang sangat ditentukan oleh tingkat kematangannya, adalah faktor krusial yang mempengaruhi harga jual, daya simpan, dan penerimaan oleh konsumen. Petani secara tradisional melakukan proses sortasi secara manual, yang sangat bergantung pada persepsi visual subjektif dan pengalaman individu. Metode ini memiliki beberapa kelemahan signifikan, antara lain inkonsistensi hasil, membutuhkan waktu dan tenaga kerja yang besar, serta rentan terhadap kelelahan manusia yang dapat menyebabkan kesalahan dalam klasifikasi [1]. Kesalahan dalam sortasi dapat mengakibatkan kerugian ekonomi, di mana tomat yang terlalu matang cepat membusuk, sementara tomat yang belum matang memiliki kualitas rasa yang rendah.

Seiring dengan kemajuan teknologi Industri 4.0, kecerdasan buatan (AI) menawarkan solusi inovatif untuk mengatasi tantangan di sektor pertanian. Salah satu sub-bidang AI yang paling menjanjikan adalah *deep learning*, khususnya *Convolutional Neural Network* (CNN). CNN merupakan arsitektur yang terbukti sangat andal dalam tugas-tugas pengenalan objek dan klasifikasi gambar [2]. Kemampuannya untuk secara otomatis mempelajari fitur-fitur hirarkis dari gambar, mulai dari tepi dan tekstur sederhana hingga bentuk objek yang kompleks, menjadikannya kandidat ideal untuk mengotomatisasi proses sortasi tomat berdasarkan kondisi visualnya [2].

Penelitian ini bertujuan untuk merancang, mengimplementasikan, dan mengevaluasi sebuah model CNN untuk mengklasifikasikan kondisi buah tomat ke dalam tiga kelas utama: matang, mentah (belum matang), dan rusak. Dengan mengembangkan sistem yang objektif dan efisien melalui optimasi hyperparameter, penelitian ini diharapkan dapat memberikan solusi teknologi praktis bagi para petani untuk meningkatkan akurasi sortasi, menekan kerugian pasca panen, dan pada akhirnya meningkatkan profitabilitas usaha tani mereka.

2. Metode Penelitian

2.1 Pengumpulan Data

Sumber data merupakan fondasi dari setiap proyek *machine learning*. Untuk penelitian ini, digunakannya dataset citra tomat yang tersedia secara *open source* di website Kaggle [4]. Pemilihan Kaggle didasarkan pada aksesibilitasnya yang mudah dan ketersediaan dataset yang cukup beragam untuk keperluan eksperimen. Dataset yang digunakan terdiri dari tiga kelas yang telah dilabeli sebelumnya:

- **Matang (*Ripe*):** Tomat yang berwarna merah merata dan siap untuk dikonsumsi.
- **Mentah (*Unripe/Belum Matang*):** Tomat yang masih berwarna hijau atau oranye.
- **Rusak (*Rotten*):** Tomat yang menunjukkan tanda-tanda kebusukan, memar, atau berjamur.

Dataset dibagi menjadi dua bagian dengan komposisi sebagai berikut:










- **Data Latih (*Training Data*):** 100 gambar untuk setiap kelas, sehingga total menjadi 300 gambar lalu di lakukan split 0.2 sebagai validation set sehingga menjadi 240 total data train dan 60 validation set.
- **Data Uji (*Testing Data*):** 20 gambar untuk setiap kelas, sehingga total menjadi 60 gambar.

Tabel 1. Distribusi citra tiap kelas

Kelas	Format	Jumlah Citra
Matang	jpg	100
Mentah	jpg	100
Rusak	jpg	100

Pembagian ini dilakukan untuk melatih model pada sebagian besar data dan kemudian menguji kemampuannya dalam menggeneralisasi pada data baru yang belum pernah dilihat sebelumnya. Untuk visualisasi dari beberapa contoh citra yang digunakan pada dataset bisa di lihat di tabel 1.

Tabel 2. Contoh citra dari ke tiga kelas

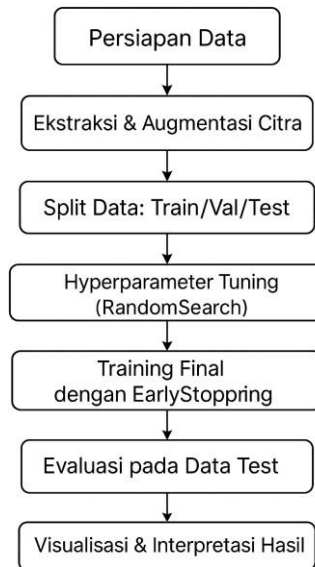
Tomat Matang	Tomat Belum Matang	Tomat Rusak/Busuk
		
		
		

2.2 Pipeline CNN

Alur kerja model yang diusulkan terdiri dari beberapa tahapan berurutan yang dirancang secara sistematis. Proses diawali dengan Persiapan Data, yang mencakup pengumpulan dan augmentasi citra untuk memperkaya variasi data latih. Selanjutnya, dataset dibagi menjadi tiga bagian: data latih (*train*), data validasi (*validation*), dan data uji (*test*).

Tahap optimasi model dilakukan melalui *Hyperparameter Tuning* dengan metode *RandomSearch* untuk menemukan konfigurasi arsitektur yang paling efektif. Berdasarkan hasil tuning, dilakukan Training Final pada model dengan menerapkan mekanisme *EarlyStopping* untuk mencegah *overfitting* dan memastikan model dapat melakukan generalisasi dengan baik. Performa model yang telah dilatih kemudian dievaluasi secara objektif pada data test untuk mengukur kinerjanya pada data yang belum pernah dilihat sebelumnya. Tahap terakhir adalah visualisasi dan Interpretasi Hasil untuk menganalisis

metrik performa dan memahami kapabilitas model secara mendalam.



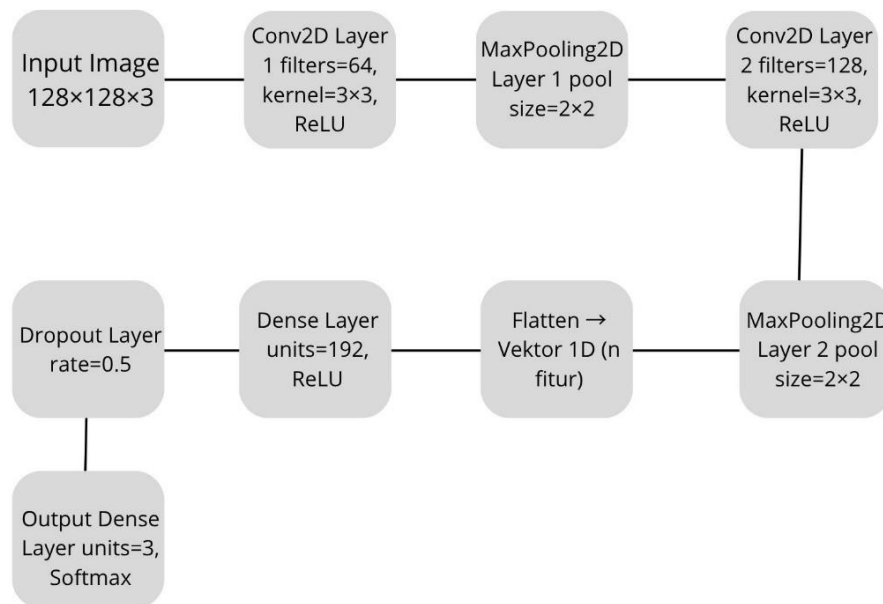
Gambar 1. Pipeline model

a. Persiapan Citra Buah Tomat

Pada tahap persiapan citra, seluruh gambar tomat yang dikumpulkan dari Kaggle [3] terlebih dahulu diubah ukurannya (“*resize*”) menjadi resolusi $128 \times 128 \times 3$ piksel dengan tiga channel warna (RGB), sehingga memenuhi syarat input pada layer pertama model CNN. Setelah itu, nilai intensitas piksel dinormalisasi ke rentang $[0, 1]$ dengan membagi setiap komponen R, G, dan B oleh 255, guna mempercepat konvergensi saat *training* dan menghindari dominasi gradien akibat skala nilai yang besar.

Karena dataset latih relatif terbatas (hanya 100 gambar per kategori), diterapkan beragam teknik augmentasi untuk meningkatkan keberagaman contoh yang dipelajari model. Metode augmentasi meliputi rotasi acak hingga $\pm 20^\circ$, pembalikan horizontal (*flip*), serta variasi kecerahan dan kontras ringan. Augmentasi ini tidak hanya menambah jumlah total data efektif, tetapi juga membantu model memahami fitur-fitur kunci (warna, tekstur, bentuk) pada berbagai variasi, sehingga mengurangi risiko overfitting dan meningkatkan kemampuan generalisasi ketika dihadapkan pada gambar uji di luar data latih.

b. Arsitektur CNN



Gambar 2. Arsitektur CNN

Arsitektur CNN ini sendiri dapat di bagi beberapa hal yaitu:

- Input: citra RGB ukuran $128 \times 128 \times 3$.
- Conv2D + ReLU: dua blok konvolusi (64 dan 128 filter), kernel 3×3 , aktivasi ReLU.
- MaxPooling2D: dua kali pooling 2×2 untuk mereduksi dimensi spasial.
- Flatten: ubah fitur 3D menjadi vektor 1D berdimensi n.
- Dense + Dropout: layer fully-connected 192 neuron + dropout 50%.
- Output Softmax: 3 neuron menghasilkan probabilitas ke tiga kelas.

c. Ekstraksi Fitur

Sebagaimana CNN secara hierarkis mengekstraksi ciri dari citra, model cnn ini belajar fitur-fitur tomat mulai dari level paling dasar hingga yang paling kompleks. Pada lapisan konvolusi pertama, kernel 3×3 mengekstrak fitur warna seperti gradasi merah oranye pada tomat matang atau hijau pada tomat belum matang dengan menghitung kombinasi linear piksel RGB dalam patch kecil. Di saat bersamaan, filter lain “menangkap” *edge features*, yang menyoroti kontur (batas luar bentuk objek) buah dan batas noda pada tomat rusak. Setelah beberapa blok konvolusi pooling, lapisan tengah mulai mempelajari tekstur lokal: pola granular, bercak, atau pori-pori halus pada permukaan tomat. Misalnya, kombinasi filter akan merespon tinggi pada area yang memiliki kontras tinggi (noda gelap vs latar cerah) atau homogenitas (permukaan halus) yang berbeda antar kelas. Di lapisan lebih dalam, representasi fitur menjadi semakin abstrak neuron-neuron dense memadukan informasi warna, tepi, dan tekstur untuk membentuk fitur high-level seperti “permukaan halus & bulat” untuk tomat matang, atau “noda + bentuk tak teratur” untuk tomat rusak. Proses dropout di sini bertindak sebagai seleksi fitur implisit, memaksa model tidak bergantung pada satu jenis fitur saja. Akhirnya, pada lapisan Flatten dan Dense terakhir, seluruh fitur laten ini diringkas menjadi vektor berdimensi 192 yang menjadi input bagi fungsi softmax. Vektor inilah yang mengandung kombinasi fitur warna, bentuk, tekstur, dan kontur paling informatif menyediakan dasar bagi prediksi tiga kelas (matang, belum matang, rusak).

d. Setting Hyperparameter

Untuk melatih model CNN dari awal, parameter pelatihan utama ditetapkan dengan mempertimbangkan kestabilan proses pelatihan serta kompleksitas jaringan. Jumlah epoch pelatihan ditentukan sebanyak 20, yang cukup untuk memungkinkan model mencapai konvergensi tanpa *overfitting*. Ukuran *batch* minimum ditetapkan sebesar 32 agar setiap pembaruan bobot tetap representatif dan efisien dalam memori. Proses *tuning hyperparameter* dilakukan menggunakan pendekatan Random Search melalui pustaka keras-tuner. Dalam metode ini, sistem mencoba secara acak kombinasi parameter dari ruang pencarian tertentu, lalu mengevaluasi kinerja berdasarkan akurasi validasi. Adapun ruang pencarian yang ditentukan meliputi:

Hyperparameter	Ruang nilai yang dicoba
num_conv_layers	{1, 2, 3}
filters per layer	{32, 64, 128}
dense_units	{64, 128, 192, 256}
dropout_rate	{0.3, 0.5, 0.7}
learning_rate	{0.01, 0.001, 0.0001}
optimizer	Adam
validation_split	0.2 (20% dari data latih digunakan sebagai validasi)
early_stopping patience	3 epochs tanpa peningkatan
max_trials	5 (konfigurasi terbaik dipilih dari 5 percobaan)

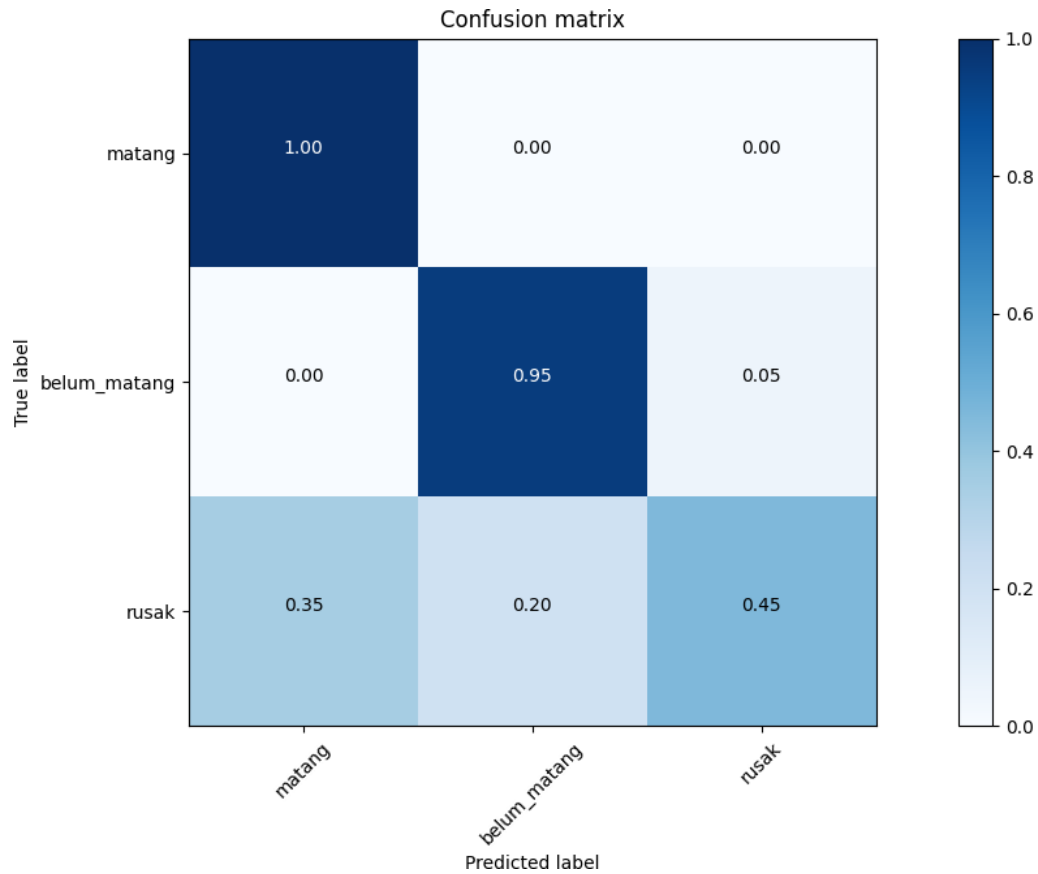
3. Hasil dan Pembahasan

3.1 Evaluasi Kinerja Model

Model akhir dievaluasi menggunakan 60 gambar dari data uji yang terpisah. Hasilnya disajikan dalam *confusion matrix* (Gambar 4) dan laporan klasifikasi (Tabel 1).

Confusion matrix memberikan wawasan mendalam tentang kinerja model:

- Kelas 'Matang': Model berhasil mengklasifikasikan semua (100%) gambar tomat matang dengan benar.
- Kelas 'Belum Matang': Model menunjukkan kinerja sangat baik dengan mengenali 95% gambar dengan benar.
- Kelas 'Rusak': Kelas ini menjadi yang paling menantang, di mana model hanya berhasil mengidentifikasi 45% gambar dengan benar. Kesalahan klasifikasi mayoritas terjadi ke arah kelas 'matang' (35%).



Gambar 3. Confusion Matrix Hasil Kinerja Model

Confusion matrix memberikan informasi mendalam tentang kinerja model:

- **Kelas 'Matang':** Model berhasil mengklasifikasikan semua (100%) gambar tomat matang dengan benar.
- **Kelas 'Belum Matang':** Model menunjukkan kinerja sangat baik dengan mengenali 95% gambar dengan benar.
- **Kelas 'Rusak':** Kelas ini menjadi yang paling menantang, di mana model hanya berhasil mengidentifikasi 45% gambar dengan benar. Kesalahan klasifikasi mayoritas terjadi ke arah kelas 'matang' (35%).

Tabel berikut merangkum metrik performa utama dari draft pengujian.

Tabel Kinerja Klasifikasi Model

Kelas	Precision	Recall	F1-Score	Jumlah Sampel
Matang	0.74	1.00	0.85	20
Belum Matang	0.83	0.95	0.88	20
Rusak	0.90	0.45	0.60	20
Akurasi Keseluruhan			0.80	60
Rata-rata Makro	0.82	0.80	0.78	60
Rata-rata Tertimbang	0.82	0.80	0.78	60

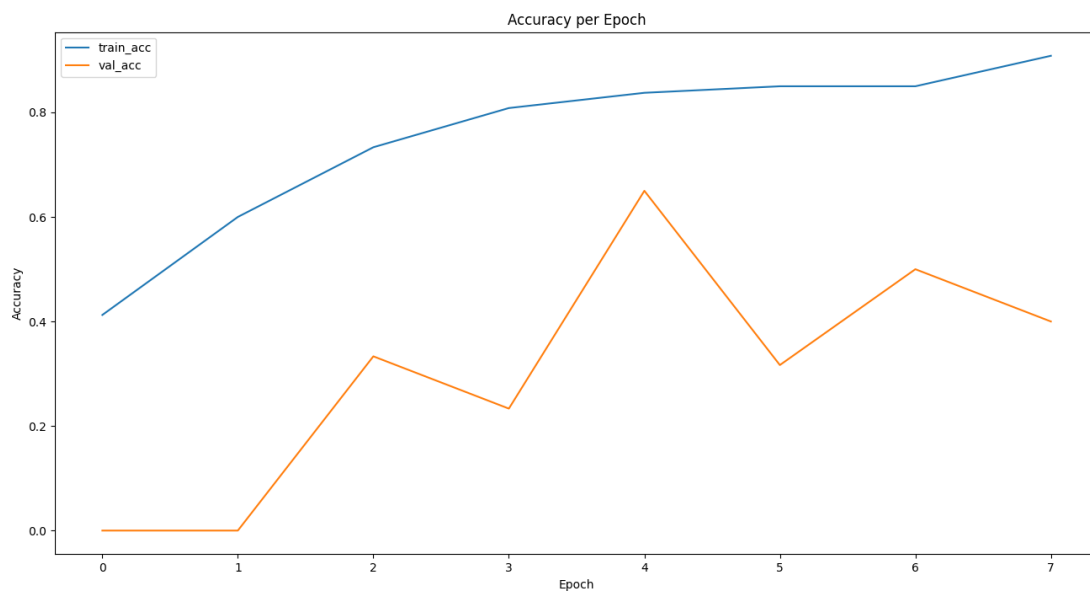
Recall 1.00 untuk kelas 'matang' menandakan model sangat andal dalam menemukan semua tomat matang. Sebaliknya, **Recall** yang rendah (0.45) untuk kelas 'rusak' menunjukkan bahwa model masih kesulitan mendeteksi semua variasi kerusakan. Tingginya **Precision** (0.90) pada kelas 'rusak' berarti jika model memprediksi sebuah tomat sebagai 'rusak', prediksi itu kemungkinan besar benar.

3.2 Detail Pelatihan dan Optimasi Model

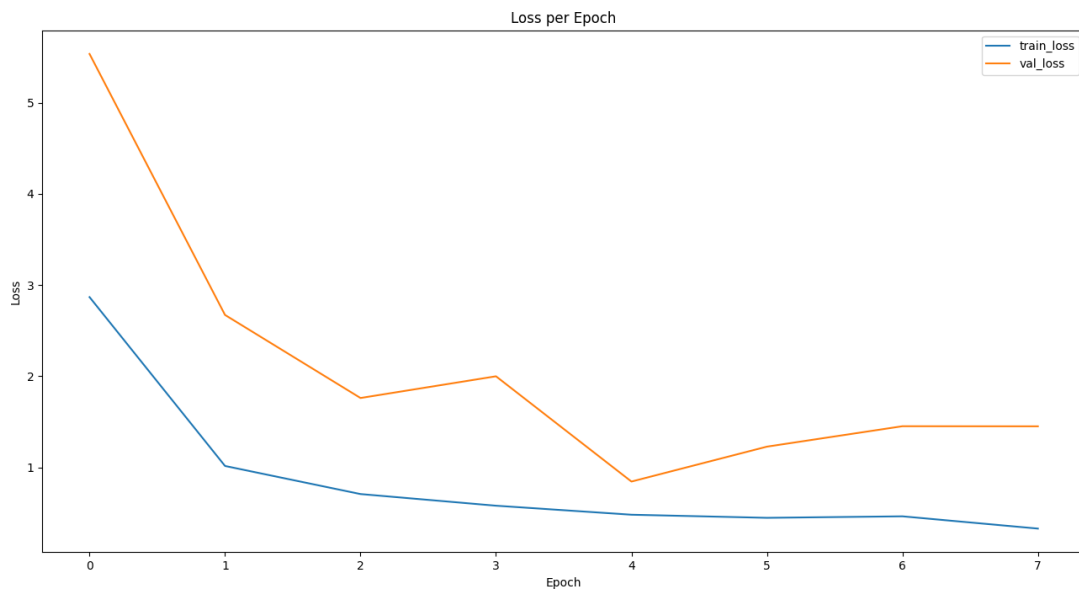
Setelah proses *hyperparameter tuning* selesai, kombinasi terbaik yang ditemukan adalah:

- Jumlah Lapis Konvolusi: 2
- Filter: 64 untuk lapisan pertama, 128 untuk lapisan kedua.
- Unit Dense: 192
- Dropout Rate: 0.5
- Learning Rate: 1e-3 (0.001)

Model dengan *hyperparameter* terbaik ini kemudian dilatih ulang pada seluruh data latih selama 20 epoch. Proses pelatihan dipantau melalui grafik akurasi dan *loss*, seperti yang direferensikan pada Gambar 2 dan Gambar 3.



Gambar 4. Akurasi per epoch



Gambar 5. Loss per epoch

Grafik-grafik tersebut menunjukkan bahwa model berhasil belajar dengan baik, ditandai dengan meningkatnya akurasi dan menurunnya *loss* pada data latih. Meskipun terdapat sedikit celah antara kurva latih dan validasi yang mengindikasikan *overfitting* ringan, penggunaan *dropout* dan *early stopping* telah membantu mengendalikannya.

4. Kesimpulan

Penelitian ini telah berhasil mengembangkan dan mengoptimalkan sebuah model *Convolutional Neural Network* (CNN) yang mampu mengklasifikasikan kondisi buah tomat dengan **akurasi keseluruhan 80%**. Melalui *hyperparameter tuning* menggunakan *RandomSearch*, kami berhasil mengidentifikasi arsitektur yang efektif. Model menunjukkan performa sangat baik untuk kelas **matang** dan **belum matang**, namun menghadapi tantangan pada kelas **rusak** yang kemungkinan disebabkan oleh tingginya variasi visual kerusakan dan dataset yang terbatas.

Sebagai *concern* perbaikan di masa depan, beberapa perbaikan dapat dilakukan:

- Penambahan Dataset: Mengumpulkan lebih banyak data, terutama untuk kelas 'rusak'.
- Augmentasi Data: Menerapkan teknik augmentasi data yang lebih agresif.
- Pengembangan Arsitektur: Mengeksplorasi arsitektur CNN yang lebih kompleks seperti *Dilated CNN* [5] atau yang menggabungkan mekanisme *attention* [4]. Implementasi *Transfer Learning* [6] juga menjadi opsi

Daftar Pustaka

- [1] S. Srinivasan, D. Francis, S. K. Mathivanan, H. Rajadurai, B. D. Shivahare, and M. A. Shah, "A hybrid deep CNN model for brain tumor image multi-classification," *BMC Med Imaging*, vol. 24, no. 1, Dec. 2024, <https://doi.org/10.1186/s12880-024-01195-7>.
- [2] F. Yu *et al.*, "Progress in the Application of CNN-Based Image Classification and Recognition in Whole Crop Growth Cycles," Jun. 01, 2023, *MDPI*. <https://doi.org/10.3390/rs15122988>.
- [3] Nguyen Quoc Khang, "Tomatoes Dataset Classification of Tomato State: Ripe, Unripe, Old, and Damaged." Accessed: Jul. 05, 2025. [Online]. Available: <https://www.kaggle.com/datasets/enalis/tomatoes-dataset>
- [4] Z. Xiangyu, "Wavelet-Attention CNN for Image Classification," Jan. 2022, [Online].

Available: <http://arxiv.org/abs/2201.09271>

- [5] L. Chen, S. Li, Q. Bai, J. Yang, S. Jiang, and Y. Miao, "Review of image classification algorithms based on convolutional neural networks," Nov. 01, 2021, *MDPI*. <https://doi.org/10.3390/rs13224712>.
- [6] Chen, L., Li, S., Bai, Q., Yang, J., Jiang, S., & Miao, Y. (2021). Review of Image Classification Algorithms Based on Convolutional Neural Networks. *Remote Sensing*, 13(22), 4712. <https://doi.org/10.3390/rs13224712>