



Implementation of Computer Vision for Detecting Diseases in Chili and Tomato Plants Using the CNN Method

Implementasi *Computer Vision* Dalam Mendeteksi Penyakit Pada Tanaman Cabai dan Tomat Menggunakan Metode CNN

Deris Pakiding¹, Ahmad Selao², Wahyuddin³

^{1,2,3}Program studi teknik informatika, Fakultas Teknik Universitas Muhammadiyah Parepare, Indonesia

E-Mail: ¹pakidingd86@gmail.com, ²ahmadselao@umpar.ac.id, ³wahyuddin081090@gmail.com

Abstract

Computer vision is widely applied in agriculture for automatic plant disease detection. This study develops a web-based application using Convolutional Neural Networks (CNN) to classify diseases in chili and tomato plants. The application, built with HTML, CSS, Python, JavaScript, and Flask, enables users to upload plant images for classification. The CNN model achieved 91% training accuracy and 90% validation accuracy. When tested on a separate dataset, it reached high accuracy. However, after being implemented in the application for real-world detection, accuracy dropped to 75%, likely due to lighting conditions, image quality, camera angles, and dataset distribution differences. Performance analysis using precision, recall, and F1-score shows that Tomato Early Blight has the highest recall (1.000), while Tomato Normal has the highest precision (0.9500). The lowest F1-score (0.8516) is observed in Normal Chili, indicating classification challenges. Heatmap analysis reveals that the model focuses on specific image features but does not always correlate with disease characteristics. Future improvements, such as data preprocessing optimization, image augmentation techniques, and advanced model architectures, are needed to enhance accuracy and reduce misclassification in real-world conditions.

Keyword: Computer Vision, CNN, Disease, Plant, Python

Abstrak

Computer Vision banyak diterapkan dalam bidang pertanian untuk mendeteksi penyakit tanaman secara otomatis. Penelitian ini mengembangkan aplikasi berbasis web menggunakan metode Convolutional Neural Networks (CNN) untuk mendeteksi penyakit pada tanaman cabai dan tomat. Aplikasi ini dibangun dengan HTML, CSS, Python, JavaScript, dan Flask, memungkinkan pengguna mengunggah gambar tanaman untuk diklasifikasikan. Model CNN mencapai akurasi pelatihan 91% dan akurasi validasi 90%. Saat diuji menggunakan dataset terpisah, model memperoleh akurasi tinggi. Namun, setelah diterapkan dalam aplikasi untuk deteksi nyata, akurasi menurun menjadi 75%, yang bisa disebabkan oleh kondisi pencahayaan, kualitas gambar, dan perbedaan dataset. Kesalahan klasifikasi terutama terjadi pada kelas Cabai Leaf Curl (50%) dan Tomat Normal (40%). Analisis kinerja menggunakan precision, recall, dan F1-score menunjukkan bahwa Tomat Early Blight memiliki recall tertinggi (1.000), sedangkan Tomat Normal memiliki precision tertinggi (0.9500). F1-score terendah (0.8516) terdapat pada Cabai Normal, menunjukkan tantangan dalam klasifikasi. Analisis heatmap mengungkapkan bahwa model berfokus pada fitur gambar tertentu tetapi tidak selalu pada area penyakit. Peningkatan kinerja melalui optimasi preprocessing data, augmentasi gambar, dan arsitektur model yang lebih kompleks diperlukan untuk meningkatkan akurasi serta mengurangi kesalahan klasifikasi.

Kata Kunci: Computer Vision, CNN, Penyakit, Tanaman, Python

1. PENDAHULUAN

Computer vision merupakan bidang yang memungkinkan mesin untuk "melihat" dengan menggunakan kamera dan komputer sebagai pengganti mata manusia untuk mengidentifikasi, melacak, dan mengukur objek sebelum diproses lebih lanjut secara digital [1]. Kemajuan teknologi di bidang Computer Vision telah memberikan berbagai solusi inovatif dalam berbagai sektor, termasuk pertanian. Salah satu tantangan utama dalam pertanian adalah penyakit tanaman, yang dapat menurunkan produktivitas dan kualitas hasil panen secara signifikan. Cabai dan tomat merupakan komoditas pertanian yang memiliki nilai ekonomi tinggi, namun rentan terhadap berbagai penyakit, seperti Leaf Curl pada cabai, serta Late Blight dan Early Blight pada tomat. Jika tidak terdeteksi secara dini, penyakit-penyakit ini dapat menyebabkan kerugian besar bagi petani [2].

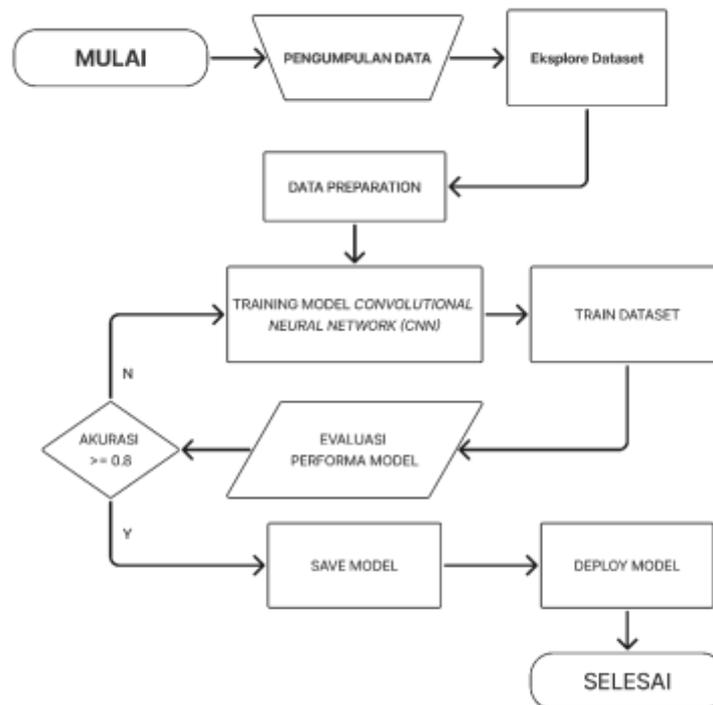
Deteksi penyakit tanaman secara manual sering kali memerlukan keahlian khusus dan membutuhkan waktu yang lama, sehingga kurang efisien jika diterapkan dalam skala luas. Oleh karena itu, diperlukan sistem otomatis yang mampu mengidentifikasi penyakit tanaman dengan akurasi tinggi. Dalam beberapa tahun terakhir, Convolutional Neural Networks (CNN) telah menjadi salah satu pendekatan deep learning yang banyak digunakan dalam analisis citra, termasuk dalam deteksi penyakit tanaman [1] [3]. CNN mampu mengenali pola visual pada gambar tanaman untuk membedakan antara kondisi sehat dan kondisi yang terinfeksi penyakit.

Penelitian ini bertujuan untuk mengembangkan aplikasi berbasis web yang menggunakan metode CNN untuk mendeteksi penyakit pada tanaman cabai dan tomat. Aplikasi ini memungkinkan pengguna mengunggah gambar tanaman yang kemudian diklasifikasikan oleh model berdasarkan kondisi kesehatannya. Model CNN yang digunakan telah dilatih menggunakan dataset gambar tanaman yang mencakup berbagai kondisi penyakit dan tanaman sehat.

Namun, berdasarkan hasil pengujian, akurasi deteksi penyakit pada aplikasi ini mencapai 75%, lebih rendah dibandingkan akurasi model saat pelatihan yang mencapai 89,18%. Disparitas ini kemungkinan disebabkan oleh faktor-faktor seperti perbedaan pencahayaan, kualitas gambar, sudut pengambilan gambar, serta distribusi data antara dataset pelatihan dan data uji. Pendekatan yang dapat diterapkan untuk meningkatkan akurasi dan mengurangi misklasifikasi antara lain adalah optimalisasi prapemrosesan data, teknik augmentasi gambar, serta penggunaan arsitektur model yang lebih kompleks. Dengan pengembangan lebih lanjut, diharapkan sistem ini dapat memberikan solusi yang lebih akurat dan efisien dalam mendeteksi penyakit tanaman cabai dan tomat di dunia nyata.

2. METODE PENELITIAN

Penelitian ini bertujuan untuk melakukan implementasi computer vision dalam mendeteksi penyakit tanaman melalui citra menggunakan metode convolutional neural network (CNN). Proses dan tahapan penelitian dapat dilihat pada Gambar 1.



Gambar 1. Metodologi Penelitian

2.1. Pengumpulan Data (Data Acquisition)

pengumpulan data dalam penelitian ini dilakukan melalui dua metode utama, yaitu pemanfaatan dataset publik dan pengambilan gambar secara mandiri.

1. Dataset Publik (PlantVillage - Kaggle)

Dataset ini digunakan sebagai sumber utama karena memiliki label yang jelas dan variasi gambar yang luas. Sebelum digunakan, dataset diproses dengan pengecekan kualitas, penyesuaian ukuran, dan

pengelompokan ulang menjadi enam kelas utama, Cabai Antraknosa, Cabai Normal, Cabai Leaf Curl, Tomat Normal, Tomat Early Blight, dan Tomat Late Blight.

2. Dataset Mandiri

Gambar diambil langsung menggunakan kamera dengan berbagai pencahayaan, sudut, dan latar belakang untuk meningkatkan performa model. Ini membantu mengidentifikasi tantangan seperti perbedaan kualitas gambar dan variasi lingkungan.

2.2. Explore Dataset

Eksplorasi data dilakukan untuk mendapatkan wawasan lebih dalam mengenai karakteristik dataset yang digunakan [4]. Proses ini mencakup analisis distribusi kategori penyakit tanaman, jumlah sampel per kelas, serta pemeriksaan faktor-faktor yang dapat mempengaruhi performa model, seperti variasi pencahayaan, sudut pengambilan gambar, dan latar belakang gambar. Dengan memahami pola-pola ini, peneliti dapat mengidentifikasi potensi ketidakseimbangan data, menentukan langkah-langkah preprocessing yang diperlukan, serta menyesuaikan strategi augmentasi guna meningkatkan akurasi dan generalisasi model dalam mendeteksi penyakit tanaman.

2.3. Data Preprocessing

Preprocessing gambar sangat penting untuk meningkatkan akurasi pengenalan dalam model CNN. teknik preprocessing guna memastikan model dapat mengenali gambar dengan lebih baik [5].

1. Rotation

Mengubah orientasi gambar agar model dapat mengenali objek dari berbagai sudut. Ini membantu model beradaptasi dengan variasi posisi dalam pengenalan gambar.

2. Resizing

Menyesuaikan dimensi gambar tanpa merusak kualitas agar sesuai dengan input model. Proses ini juga mempertahankan connected components, yaitu bagian terhubung dalam gambar yang mewakili objek utama. resizing diperlukan untuk memastikan semua gambar berada dalam format yang benar untuk diproses [6].

3. Brightness & Contrast

Mengatur tingkat kecerahan dan kontras untuk meningkatkan kualitas gambar, terutama dalam kondisi pencahayaan yang kurang optimal. Peningkatan kontras membantu membedakan objek dari latar belakang, sementara penyesuaian brightness memastikan detail tetap terlihat jelas.

2.4. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) merupakan jenis arsitektur jaringan saraf tiruan atau turunan dari ilmu neural network. CNN telah merevolusi bidang ML dengan kemampuannya untuk secara otomatis mengekstraksi fitur-fitur yang relevan dari citra data. komponen utama CNN, termasuk lapisan-lapisan yang ada di dalamnya [7].

1. Lapisan Konvolusi (Convolutional Layer)

Digunakan untuk mengekstraksi fitur dari gambar dengan operasi konvolusi, menggantikan perkalian matriks yang biasa digunakan dalam jaringan saraf tiruan tradisional. Operasi konvolusi memungkinkan jaringan mempelajari hubungan antara input dan output dengan lebih efisien, sehingga dapat menangkap fitur penting dalam gambar [8]. Lapisan ini bekerja dengan menerapkan sejumlah filter konvolusi (kernel) yang digeser di seluruh citra input. Proses ini menghasilkan feature map, yang menyoroti keberadaan fitur tertentu seperti tepi, garis, tekstur, atau pola yang lebih kompleks. Setiap filter memiliki bobot (weight) yang dapat dipelajari, yang diperbarui selama pelatihan untuk mengoptimalkan pengenalan pola [4]. Persamaan Output neuron dari convolutional layer dapat ditulis sebagai berikut :

$$z_{i,j,k} = b_k + \sum_{u=0}^{f_h-1} \sum_{v=0}^{f_w-1} \sum_{k'=0}^{f_n-1} x_{i',j',k'} \cdot W_{u,v,k,k} \quad (1)$$

$$\text{Dengan} = \begin{cases} i' = i \times s_h + u \\ j' = j \times s_w + v \end{cases} \quad (2)$$

2. Lapisan Aktivasi (Activation Layer)

Setelah lapisan konvolusi, fungsi aktivasi non-linear seperti ReLU (Rectified Linear Unit) diterapkan untuk memperkenalkan non-linearitas dan membantu jaringan mempelajari hubungan kompleks antara fitur. ReLU mengubah nilai negatif menjadi nol, sementara nilai positif tetap, sehingga memperkuat fitur penting dan mempercepat proses pembelajaran. Keunggulan utama ReLU adalah beban komputasi yang lebih rendah dibandingkan fungsi aktivasi lainnya [7].

$$f(x) = \max(0, x) \quad (3)$$

3. Lapisan Pooling

Lapisan pooling digunakan untuk mengurangi dimensi spasial dari fitur-fitur yang ditemukan oleh lapisan konvolusi. Metode pooling yang umum adalah max pooling. Pooling membantu mengurangi jumlah parameter dalam jaringan dan menghasilkan fitur representasi yang lebih tahan terhadap pergeseran dan deformasi kecil pada citra. Ini juga membantu dalam menghindari overfitting dan membuat model lebih umum.

4. Lapisan Fully Connected

menghubungkan semua neuron dari lapisan sebelumnya ke setiap neuron di lapisan berikutnya, seperti pada jaringan saraf tiruan biasa. Lapisan ini bertugas menggabungkan fitur yang diekstraksi untuk mempelajari hubungan kompleks dan melakukan klasifikasi. Pada tahap akhir, fungsi aktivasi Softmax dapat digunakan untuk mengubah output menjadi distribusi probabilitas, sehingga model dapat menentukan kelas dengan probabilitas tertinggi. Softmax menghitung nilai probabilitas setiap kelas dengan rumus:

$$P(y_i) = \frac{e^{z_i}}{\sum_j e^{z_j}} \quad (4)$$

2.5. Evaluasi Performa Model

Evaluasi model pada CNN (Convolutional Neural Network) adalah proses untuk menilai kinerja dan akurasi model dalam mengklasifikasi objek atau matriks. Dilakukan melalui metode evaluation matrix yang menggunakan tabel Confusion Matrix. Confusion Matrix yakni metode visualisasi untuk hasil algoritma klasifikasi. Dalam kata lain ini merupakan tabel yang memecah jumlah instansi kebenaran dasar dari kelas tertentu terhadap jumlah instansi kelas yang diprediksi [9] [3].

Adapun evaluasi matriks yang dapat digunakan untuk menghitung nilai dari Accuracy, Precision, Recall dan F-1 Score dari performa model Machine Learning yang telah dilatih [10].

1. Accuracy

Akurasi dari pengklasifikasi diukur dengan metrik ini. Jumlah data yang diklasifikasikan dengan benar dibagi oleh total jumlah data untuk menghitung akurasi. Rumus untuk menghitung akurasi adalah sebagai berikut:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (5)$$

2. Precision

Presisi menunjukkan seberapa banyak data yang diprediksi sebagai positif yang diprediksi dengan benar. Dengan kata lain, presisi yang tinggi berarti lebih sedikit false positives. Rumus untuk menghitung presisi adalah sebagai berikut :

$$Precision = \frac{TP}{TP+FP} \quad (6)$$

3. Recall

Recall adalah metrik untuk menentukan kelengkapan dari pengklasifikasi. Recall yang lebih tinggi menunjukkan false negatives yang lebih rendah, sementara recall yang lebih rendah menunjukkan false negatives yang lebih tinggi. Presisi sering kali menurun dengan peningkatan recall. Rumus untuk menghitung recall adalah sebagai berikut:

$$Recall = \frac{TP}{TP+FN} \quad (7)$$

4. F1-Score

Untuk mendapatkan F1-Score, hasil kali dari recall dan precision dibagi oleh jumlah dari recall dan precision. Rumus untuk menghitung F1-Score adalah sebagai berikut:

$$F1 - Score = 2 \times \frac{Recall \times Precision}{Recall+Precision} \quad (8)$$

2.6. Deployment

Deployment adalah proses menerapkan model yang telah dilatih ke dalam aplikasi agar dapat digunakan oleh pengguna. Dalam penelitian ini, model CNN untuk deteksi penyakit tanaman diterapkan dalam aplikasi berbasis web dengan menggunakan Flask sebagai backend. Flask berfungsi untuk menangani permintaan dari pengguna dan menjalankan model AI dalam menghasilkan prediksi [11]. Python digunakan sebagai bahasa pemrograman utama untuk membangun model, mengolah data, serta mengintegrasikan model dengan sistem [12]. Sementara itu, JavaScript berperan dalam meningkatkan interaktivitas pada frontend, seperti menampilkan hasil prediksi secara dinamis di halaman web, sehingga pengalaman pengguna menjadi lebih responsif [13].

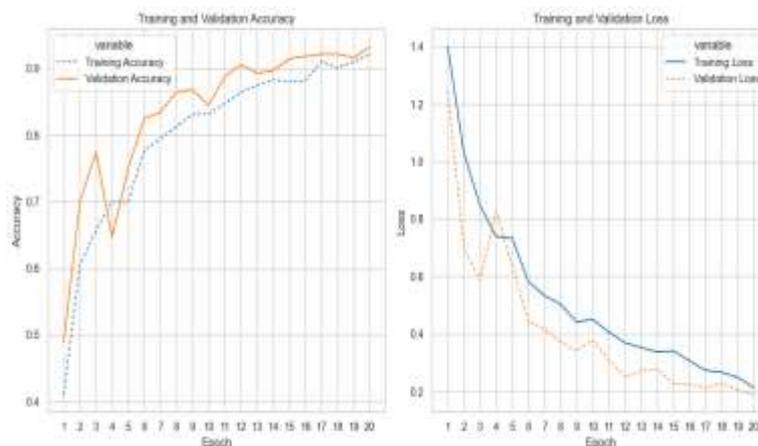
3.1. Hasil dan Pembahasan

Pengumpulan data dilakukan dengan menggunakan dataset publik Plant Village dari kaggle dan dataset yang diambil secara mandiri menggunakan kamera gawai. Dataset yang digunakan dalam penelitian ini berjumlah 3.734 data terbagi menjadi 6 (enam) kelas dengan data yang terdiri dari citra daun tomat dan buah cabai dengan kondisi normal dan kondisi terserang penyakit, dengan distribusi data sebagai berikut :

Tabel 1. Distribusi data

Kelas	Jumlah Data
Cabai Normal	600
Cabai Antraknosa	620
Cabai <i>leaf curl</i>	621
Tomat Normal	670
Tomat <i>Early Blight</i>	612
Tomat <i>Late Blight</i>	611

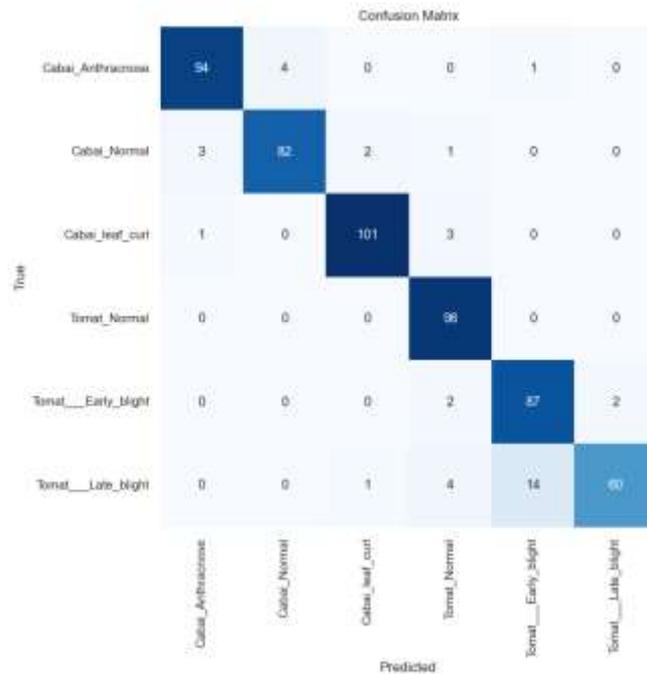
Dataset tersebut dibagi menjadi 3 (tiga) bagian (split dataset) yakni Train Dataset, Validation Dataset, dan Test Dataset dengan rasio 70:15:15 dan juga dilakukan augmentasi data dengan teknik flipping, dan rotasi, contrast dan brightness.



Gambar 2. Training Accuracy dan Loss

Grafik akurasi menunjukkan tren positif di mana akurasi untuk data training dan validasi meningkat seiring bertambahnya epoch. Peningkatan ini mencerminkan bahwa model mampu mempelajari pola dalam data secara efektif. Pada tahap akhir training, terutama pada epoch ke-18 hingga ke-20 berada di kisaran 91%, akurasi training dan validasi berada pada tingkat yang hampir sama. Hal ini mengindikasikan bahwa model telah mencapai titik konvergensi, di mana kemampuan model untuk mempelajari data training dan generalisasi terhadap data validasi berada dalam keseimbangan. Menariknya, pada beberapa epoch awal, akurasi validasi terlihat lebih tinggi dibandingkan akurasi training. Fenomena ini menunjukkan bahwa model mampu melakukan generalisasi dengan baik pada data baru.

Grafik loss memperlihatkan penurunan nilai loss untuk data training dan validasi seiring bertambahnya epoch. Penurunan ini menunjukkan bahwa model berhasil meminimalkan error dengan baik selama proses training. Pada sebagian besar epoch, loss validasi sedikit lebih tinggi dibandingkan loss training, yang merupakan hal umum karena model diuji pada data yang belum pernah dilihat sebelumnya. Selain itu, pada epoch-epoch akhir, nilai loss validasi cenderung stabil dan tidak mengalami peningkatan yang signifikan.



Gambar 3. Confusion Matrix

Nilai pada diagonal utama (dari kiri atas ke kanan bawah) menunjukkan jumlah deteksi yang benar untuk setiap kelas.

Tabel 2. Deteksi benar dataset test

Kelas Aktual	Deteksi Benar
Cabai Antraknosa	94
Cabai Normal	82
Cabai <i>Leaf Curl</i>	101
Tomat Normal	98
Tomat <i>Early Blight</i>	87
Tomat <i>Late Blight</i>	60

Tabel 3. Deteksi salah pada dataset test

Kelas Aktual	Kesalahan Kelas deteksi	Jumlah Kesalahan	Keterangan
Cabai Antraknosa	Cabai Normal	4	Beberapa sampel salah diklasifikasikan sebagai Cabai_Normal.
Cabai Antraknosa	Tomat <i>Early Blight</i>	1	Satu sampel salah diklasifikasikan sebagai Tomat <i>Early Blight</i>
Cabai Normal	Cabai Antraknosa	3	Sebagian kecil prediksi salah diarahkan ke Cabai_Anthracnose.
Cabai Normal	Cabai <i>Leaf Curl</i>	2	Sebagian kecil kesalahan diarahkan ke Cabai <i>leaf curl</i> .
Cabai Normal	Tomat Normal	1	Satu kesalahan deteksi diarahkan ke Tomat Normal
Cabai <i>Leaf Curl</i>	Cabai Antraknosa	1	Kesalahan kecil kelas diarahkan ke Cabai <i>Anthracnose</i>

Cabai <i>Leaf Curl</i>	Tomat Normal	3	Beberapa kesalahan diarahkan ke Tomat Normal
Tomat <i>Early Blight</i>	Tomat Normal	2	kesalahan ke kelas Tomat Normal yang berbeda.
Tomat <i>Early Blight</i>	Tomat <i>Late Blight</i>	2	Kesalahan cukup besar ke kelas yang mirip secara visual.
Tomat <i>Late Blight</i>	Cabai <i>Leaf Curl</i>	1	Kesalahan diarahkan ke kelas Cabai <i>Leaf Curl</i>
Tomat <i>Late Blight</i>	Tomat Normal	4	Sebagian kecil Kesalahan diarahkan ke Tomat Normal
Tomat <i>Late Blight</i>	Tomat <i>Early Blight</i>	14	Kesalahan terbesar antara dua kelas yang mirip secara visual.

Kesalahan klasifikasi yang terjadi menunjukkan bahwa model mengalami kesulitan dalam membedakan beberapa kelas, terutama yang memiliki kemiripan visual. Pada tanaman cabai, kesalahan klasifikasi paling banyak terjadi pada Cabai Normal, di mana 3 sampel salah diklasifikasikan sebagai Cabai Antraknosa, 2 sampel sebagai Cabai *Leaf Curl*, dan 1 sampel sebagai Tomat Normal. Hal ini menunjukkan bahwa model mengalami kesulitan dalam membedakan kondisi cabai yang sehat dan yang mengalami penyakit tertentu, terutama yang memiliki gejala serupa seperti perubahan warna dan tekstur daun. Selain itu, Cabai Antraknosa juga mengalami beberapa kesalahan, dengan 4 sampel salah diklasifikasikan sebagai Cabai Normal dan 1 sampel salah terdeteksi sebagai Tomat *Early Blight*, yang menandakan adanya fitur visual yang tumpang tindih antara kelas tersebut. Cabai *Leaf Curl* juga mengalami kesalahan klasifikasi, di mana 1 sampel salah diklasifikasikan sebagai Cabai Antraknosa dan 3 sampel salah dikenali sebagai Tomat Normal, yang menunjukkan bahwa model kurang mampu membedakan ciri khas dari kondisi daun yang menggulung dengan kondisi normal tanaman tomat.

Pada tanaman tomat, kesalahan terbesar terjadi antara Tomat *Early Blight* dan Tomat *Late Blight*, di mana masing-masing memiliki kesalahan silang dengan 2 sampel salah diklasifikasikan ke kelas yang berlawanan. Hal ini menunjukkan bahwa gejala visual kedua penyakit ini cukup mirip, seperti bercak dan perubahan warna daun, yang membuat model kesulitan dalam membedakannya. Selain itu, Tomat *Early Blight* juga memiliki 2 kesalahan klasifikasi ke Tomat Normal, yang menunjukkan bahwa beberapa sampel mungkin memiliki gejala ringan sehingga terdeteksi sebagai kondisi sehat. Pada Tomat *Late Blight*, terdapat 1 sampel yang salah diklasifikasikan sebagai Cabai *Leaf Curl*, yang bisa terjadi akibat pola visual tertentu yang menyerupai gejala penyakit pada cabai. Selain itu, Tomat *Late Blight* juga memiliki kesalahan yang cukup signifikan, dengan 4 sampel salah diklasifikasikan sebagai Tomat Normal dan 14 sampel salah dikenali sebagai Tomat *Early Blight*, yang memperkuat indikasi bahwa model mengalami kesulitan dalam membedakan penyakit yang memiliki karakteristik visual serupa. Data yang diperoleh dari Confusion Matrix digunakan untuk mendapatkan nilai dari Precision, Recall, dan F1-Score dari tiap kelas, sebagai berikut :

Tabel 4. Precision, Recall, F1-Score

Kelas	<i>Precision</i>	<i>Recall</i>	F1 - Score
Cabai Antraknosa	0.9591	0.9494	0.9543
Cabai Normal	0.9534	0.9318	0.9425
Cabai <i>Leaf Curl</i>	0.9711	0.9619	0.9665
Tomat Normal	0.9074	0.95600	0.9514
Tomat <i>Early Blight</i>	0.8529	0.9560	0.9015
Tomat <i>Late Blight</i>	0.9677	0.7594	0.8510

Berdasarkan evaluasi Precision, Recall, dan F1-Score, model menunjukkan performa yang cukup baik dalam mendeteksi penyakit pada cabai dan tomat, meskipun terdapat beberapa kelas yang perlu ditingkatkan.

Pada Cabai Antraknosa, model memiliki keseimbangan performa dengan Precision 0.9591 dan Recall 0.9494, menghasilkan F1-Score 0.9543. Hal ini menunjukkan bahwa model dapat mendeteksi sebagian besar sampel dengan akurasi tinggi dan hanya sedikit kesalahan dalam prediksi. Cabai Normal memiliki F1-Score terendah (0.9425) dibandingkan kelas lainnya, dengan Recall lebih rendah (0.9318), menunjukkan adanya

kesulitan dalam membedakan kelas ini dari yang lain, kemungkinan karena kemiripan fitur atau distribusi data yang kurang merata.

Untuk Cabai Leaf Curl, model bekerja sangat baik dengan Precision 0.9711 dan Recall 0.9619, menghasilkan F1-Score 0.9665. Hal ini menunjukkan bahwa model mampu mengenali kelas ini dengan baik dan memiliki sedikit kesalahan prediksi.

Pada Tomat Normal, model memiliki Recall tinggi (0.9560), menunjukkan bahwa hampir semua sampel terdeteksi dengan benar, namun Precision yang lebih rendah (0.9074) mengindikasikan adanya False Positives. Dengan F1-Score 0.9514, performanya cukup baik tetapi masih dapat ditingkatkan.

Kelas Tomat Early Blight memiliki Recall tinggi (0.9560), menandakan model mampu mengenali sebagian besar sampel, tetapi Precision 0.8529 menunjukkan bahwa masih ada kesalahan dalam klasifikasi. F1-Score sebesar 0.9015 mencerminkan bahwa model cukup baik, namun masih bisa diperbaiki untuk mengurangi kesalahan prediksi.

Sementara itu, Tomat Late Blight memiliki Precision tertinggi (0.9677), tetapi Recall yang rendah (0.7594) menunjukkan bahwa model masih melewatkan sejumlah besar sampel yang sebenarnya termasuk dalam kelas ini. F1-Score sebesar 0.8510 menandakan adanya ketidakseimbangan, di mana model sangat yakin dalam prediksinya tetapi kurang sensitif dalam mendeteksi semua kasus yang ada.



Gambar 4. Tampilan utama website one-page

Setelah proses deployment, dilakukan pengujian kembali untuk mengevaluasi performa aplikasi dalam mendeteksi penyakit tanaman cabai dan tomat. Pengujian ini bertujuan untuk memastikan bahwa model bekerja dengan baik dalam lingkungan nyata, menguji respons aplikasi terhadap berbagai kondisi input.

Tabel 5. Pengujian deteksi aplikasi.

Kelas Aktual	Deteksi Benar	Deteksi Salah
Cabai Antraknosa	10	0
Cabai Normal	9	1
Cabai <i>Leaf Curl</i>	5	5
Tomat Normal	6	4
Tomat <i>Early Blight</i>	7	3
Tomat <i>Late Blight</i>	8	2

Hasil pengujian ini diperoleh dari percobaan deteksi menggunakan data nyata yang diambil langsung melalui kamera. Data ini mencakup enam kelas, yaitu Cabai Antraknosa, Cabai Normal, Cabai Leaf Curl, Tomat Normal, Tomat Early Blight, dan Tomat Late Blight. Setiap kelas memiliki jumlah sampel tertentu yang digunakan untuk mengukur kemampuan model dalam mengklasifikasikan kondisi tanaman dengan benar atau salah.

$$\begin{aligned}
 \text{Akurasi} &= \frac{\text{Jumlah deteksi benar}}{\text{Total sampel}} \times 100\% \\
 \text{Akurasi} &= \frac{(10 + 9 + 5 + 6 + 7 + 8)}{(10) + (9 + 1) + (5 + 5) + (6 + 4) + (7 + 3) + (8 + 2)} \times 100\% = \frac{42}{60} \times 100\% = 75\%
 \end{aligned}$$

berdasarkan pengujian manual yang dilakukan setelah implementasi model ke dalam aplikasi berbasis web, akurasi deteksi penyakit tanaman cabai dan tomat mencapai 75%. Hasil ini menunjukkan adanya perbedaan dengan akurasi yang diperoleh selama pelatihan model, yang sebelumnya mencapai 91%. Perbedaan ini bisa disebabkan oleh beberapa faktor, seperti kondisi pencahayaan, kualitas gambar yang diambil melalui kamera, sudut pengambilan gambar, serta kemungkinan adanya perbedaan distribusi data.



Gambar 5. Gagal deteksi objek non kategori



Gambar 6. Visualisasi feature map layers

Visualisasi feature map ini menunjukkan bagaimana model CNN memberikan perhatian pada area tertentu dalam gambar yang tidak termasuk dalam kategori yang telah dilatih, namun tetap diklasifikasikan sebagai cabai antraknosa. Dari pola aktivasi yang terlihat, model tampaknya mengidentifikasi bagian dengan tekstur atau kontras tertentu yang menyerupai karakteristik penyakit pada cabai, sehingga terjadi salah klasifikasi. Aktivasi yang kuat pada area tertentu menunjukkan bahwa model memiliki pola spesifik yang dijadikan acuan untuk mengenali cabai antraknosa, yang mungkin juga ditemukan dalam gambar input ini. Kesalahan ini bisa disebabkan oleh keterbatasan dataset pelatihan yang tidak memiliki cukup variasi, sehingga model kurang mampu membedakan objek yang tidak relevan. Selain itu, model tidak dilatih untuk mengenali kategori "di luar kelas" (out of distribution), sehingga setiap input yang diberikan tetap akan dipetakan ke salah satu kelas yang ada.

4. Kesimpulan

Penelitian ini mengimplementasikan teknologi computer vision pada aplikasi berbasis web untuk mendeteksi penyakit pada tanaman cabai dan tomat menggunakan metode Convolutional Neural Networks (CNN). Aplikasi ini memungkinkan pengguna untuk mengunggah gambar tanaman melalui antarmuka berbasis HTML, CSS, Python, JavaScript, dan Flask, yang kemudian diproses oleh model CNN untuk mengidentifikasi jenis penyakit tanaman. Model yang dibangun mampu mengklasifikasikan enam kategori, yaitu cabai normal, cabai leaf curl, cabai antraknosa, tomat normal, tomat early blight, dan tomat late blight. Hasil evaluasi menggunakan dataset uji menunjukkan bahwa model memiliki performa yang cukup baik dengan akurasi 91% pada pelatihan. Namun, dalam tahap implementasi pada data nyata melalui kamera secara real-time, model mengalami sedikit penurunan performa dengan akurasi sebesar 75%.

Hasil pengujian lebih lanjut menunjukkan bahwa kesalahan prediksi cenderung terjadi pada kelas-kelas yang memiliki karakteristik visual yang mirip, seperti tomat early blight dan tomat late blight, serta cabai normal yang sering diklasifikasikan sebagai cabai leaf curl. Selain itu, model mengalami kesulitan dalam menangani gambar yang tidak termasuk dalam kategori pelatihan, sehingga beberapa objek di luar dataset dapat salah diklasifikasikan. Analisis menggunakan Confusion Matrix menunjukkan adanya misclassifications yang signifikan pada beberapa kelas tertentu. Visualisasi feature map juga memperlihatkan bahwa model lebih banyak fokus pada fitur tertentu dalam gambar, tetapi terkadang area yang diperhatikan kurang relevan dengan karakteristik penyakit yang sebenarnya. Hal ini menunjukkan bahwa meskipun model sudah cukup baik dalam mendeteksi penyakit tanaman, masih diperlukan peningkatan lebih lanjut agar model lebih robust terhadap data dunia nyata.

REFERENSI

- [1] H. Tian, T. Wang, Y. Liu, X. Qiao and Y. Li, "Computer Vision Technology in Agricultural Automation," *INFORMATION PROCESSING IN AGRICULTURE vol.7*, pp. 1-19, 2020.
- [2] C. Wati, A. T. Karenina, R. Y. Nirwanto, I. Nurcahya, D. Meilani, D. Astuti, D. Septiarini, S. R. F. Purba, E. P. Ramdan dan D. Nurul, *Hama dan Penyakit Tanaman*, bogor: Yayasan Kita Menulis, 2021.

-
- [3] A. Taner, Y. B. Oztekin dan H. Duran, "Performance Analysis of Deep Learning CNN Models for Variety Classification in Hazelnut," *Sustainability*, 2021.
- [4] A. Geron, *Hands-On Machine Learning With Scikit-Learn, Keras & TensorFlow*, Canada: O'Reilly Media, 2019.
- [5] T. Mulia, M. Kallista dan P. D. Wibawa, "Preprocessing Gambar Sampah untuk Sistem Pemilah Sampah Otomatis menggunakan Roboflow," *e-Proceeding of Engineering*, vol. 11, pp. 6743-6247, 2024.
- [6] A. Dubey, A. Lazarus dan D. Mangal, "Handwritten Digit Recognition using Image Preprocessing and CNN," *IJSRCSEIT*, 2020.
- [7] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Djuaili, Y. Duan, O. Al-Shama, S. J. dan L. Farhan, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *Journal of Big Data*, 2021.
- [8] J. Sanjaya dan M. Ayub, "Augmentasi Data Pengenalan Citra Mobil Menggunakan Pendekatan Random Crop, Rotate, dan Mixup," *Jurnal Teknik Informatika dan Sistem Informasi*, 2020.
- [9] . J. Murel Ph.D. dan E. Kavlakoglu, "What Is Confusion a Matrix," 19 01 2024. [Online]. Available: <https://www.ibm.com/topics/confusion-matrix>.
- [10] A. Rahman, *Brain Tumor Detection and Classification by Using CNN*, Finland: UEF eRepository, 2024.
- [11] A. Amanzadi dan M. Karim, *Comparison of Machine Learning Models Used for Swedish Text Classification in Chat Messaging*, Stockholm: DiVa, 2022.
- [12] S. Raschka, "Machine Learning in Python: Main Developments and Technology Trends In Data Science, Machine Learning and Artificial Intelligence," *Information*, 2020.
- [13] Y. Supriadi, *SEMUA BISA MENJADI PROGRAMMER JAVASCRIPT & NODE.JS*, Jakarta: PT Elex Media Komputindo, 2020.