



Implementasi *Transfer Learning* Dalam Klasifikasi Kematangan Buah Naga Menggunakan Model Inception V3

Berly Ramadhan¹, Cindi Wulandari^{2,*}, Fido Rizki²

¹ Fakultas Ilmu Teknik, Program Studi Teknik Informatika, Universitas Bina Insan, Lubuklinggau, Indonesia

² Fakultas Ilmu Teknik, Program Studi Sistem Informasi, Universitas Bina Insan, Lubuklinggau, Indonesia

Email: ¹berlyyrmhdhnn@gmail.com, ^{2,*}cindi_wulandari@univbinainsan, ³fidorizki@univbinainsan.ac.id

Email Penulis Korespondensi: cindi_wulandari@univbinainsan

Abstrak– Proses pengolahan data kematangan buah naga dengan menggunakan arsitektur *inception V3* yang mana dalam menganalisis kematangan buah naga dapat mengklasifikasikan hasil yang berhubungan dengan image processing. Adapun tujuan penelitian ini adalah untuk mengembangkan implementasi kematangan buah naga yang berhubungan dengan bahasa pemrograman python serta transfer learning pada kematangan buah. Dapat melakukan proses ekstraksi pengenalan pola kematangan buah berdasarkan citra digital dengan menggunakan model Inception V3. Selain itu pada pengolahan data mempergunakan perhitungan yang akurat seperti akurasi, presisi, recall dan F1-score yang dapat membagi fitur kelas sehingga sistem image processing dalam membedakan mana buah matang dengan buah yang belum matang.

Kata Kunci: Klasifikasi; Buah Naga; *Deep Learning* dan *Python*.

Abstract– The process of processing dragon fruit maturity data using inception V3 architecture which in analyzing dragon fruit maturity can classify results related to image processing. The purpose of this research is to develop the implementation of dragon fruit maturity that relates to the python programming language and transfer learning on fruit maturity. Can perform the process of extracting fruit maturity pattern recognition based on digital images using the Inception V3 model. In addition, data processing uses accurate calculations such as accuracy, precision, recall and F1-score which can divide the feature class so that the image processing system can distinguish which fruit is ripe from immature fruit.

Keywords: Classification; Dragon Fruit; *Deep Learning* and *Python*

1. PENDAHULUAN

Seiring dengan perkembangan teknologi, implementasi kecerdasan buatan telah menjadi cabang keilmuan yang sangat pesat dalam meningkatkan efisiensi. Salah satu teknologi kecerdasan buatan yang berhubungan dengan *computer vision* adalah pengklasifikasian citra, dimana klasifikasi citra merupakan proses untuk mendeteksi objek dari suatu citra yang ada. Oleh karena itu pada penulis sangat terinspirasi dalam melakukan penulisan yang berhubungan dengan buah naga yang mau dijadikan objek pada penulisan ini yang mengamati tentang kematangan buah naga. Buah naga (*Inggris: pitaya*) adalah buah dari beberapa jenis kaktus dari marga *Hylocereus* dan *Selenicereus*. Buah ini berasal dari Meksiko, Amerika Tengah dan Amerika Selatan namun sekarang juga dibudidayakan di negara-negara Asia seperti Indonesia, Taiwan, Vietnam, Filipina, dan Malaysia. Buah ini juga dapat ditemui di Okinawa, Israel, Australia utara dan Tiongkok selatan [1].

Transfer learning adalah metode *Deep learning* yang menggunakan model yang pernah dilatih sebelumnya (*pre-trained*) sebagai dasar untuk melatih model yang baru. Sederhananya, metode pembelajaran lama digunakan kembali ketitik awal untuk model lain dalam tugas yang baru. Dengan menerapkan *transfer learning* pada tugas baru, seseorang bisa mencapai performa yang secara signifikan jauh lebih tinggi daripada melakukan pelatihan dengan jumlah data yang sedikit. *Inception V3* merupakan salah satu arsitektur *convolutional neural network* dari *Google Net* dan merupakan versi jaringan *Convolutional Neural Network (CNN)* yang dibangun oleh *GoogleNet* [2]. *Google* memperkenalkan *Inception* dengan teknik baru yang ada dalam arsitektur *Convolutional Neural Network* yaitu dengan melakukan pemfaktoran pada lapisan konvolusi sehingga menjadi multi-layers menggunakan kernel berukuran kecil. Penggunaan *Inception* dapat mengurangi jumlah parameter yang ada dengan membagi bobot dalam multilayers tersebut. *Python* merupakan bahasa pemrograman komputer yang biasa dipakai untuk membangun situs, software/aplikasi, mengotomatiskan tugas dan melakukan analisis data. Bahasa pemrograman ini termasuk bahasa tujuan umum. Artinya, ia bisa digunakan untuk membuat berbagai program berbeda, bukan khusus untuk masalah tertentu saja [3].

Metode transfer learning banyak diadopsi karena memungkinkan pemanfaatan model yang telah dilatih pada dataset berskala besar, seperti ImageNet, untuk kemudian disesuaikan (*fine-tuning*) pada dataset spesifik. Salah satu arsitektur populer yang digunakan adalah Inception V3, yang dikenal memiliki efisiensi tinggi, jumlah parameter relatif lebih sedikit dibanding model terdahulu, namun tetap mempertahankan akurasi yang baik. Inception V3 menggunakan pendekatan *factorized convolutions*, *asymmetric convolutions*, dan *global average pooling* sehingga lebih efektif dalam mempelajari representasi fitur visual [4]. *Python* merupakan bahasa pemrograman komputer yang biasa dipakai untuk membangun situs, software/aplikasi, mengotomatiskan tugas dan melakukan analisis data. Bahasa pemrograman ini termasuk bahasa tujuan umum. Artinya, ia bisa digunakan untuk membuat berbagai program berbeda, bukan khusus untuk masalah tertentu saja. Untuk menerapkan metode data mining ini penulis menggunakan model *Convolutional Neural Network (CNN)*. Metode *Convolutional Neural Network (C-NN)* merupakan sebuah metode klasifikasi terhadap sekumpulan data berdasarkan pembelajaran data yang sudah terklasifikasi sebelumnya. Tujuan penelitian ini adalah



untuk melakukan proses ekstraksi pengenalan pola kematangan buah berdasarkan citra digital dengan menggunakan model Inception V3 [5].

Penelitian mengenai klasifikasi tingkat kematangan buah naga telah dilakukan oleh beberapa peneliti di Indonesia dengan berbagai pendekatan, meskipun sebagian besar masih menggunakan metode ekstraksi fitur klasik dan algoritma machine learning konvensional. Misalnya, Ardiansyah dkk (2021) mengembangkan sistem cerdas deteksi kematangan buah naga berbasis ruang warna HSV yang kemudian diklasifikasikan menggunakan algoritma *K-Nearest Neighbor* (KNN). Dataset yang digunakan berjumlah 120 citra buah naga pada berbagai tingkat kematangan. Hasil pengujian menunjukkan bahwa metode HSV-KNN mampu mencapai akurasi hingga 91,94%. Penelitian ini menegaskan bahwa informasi warna sangat dominan dalam menentukan kematangan buah naga, meskipun metode yang digunakan masih sederhana dan sangat bergantung pada kondisi pencahayaan.

Selanjutnya, Sari dan Nurhidayat (2022) melakukan klasifikasi tingkat kematangan buah naga menggunakan pendekatan *Support Vector Machine* (SVM). Penelitian ini memanfaatkan fitur warna dari citra digital buah naga dengan tiga kategori kematangan, yaitu mentah, setengah matang, dan matang. Hasilnya menunjukkan bahwa SVM mampu mengklasifikasikan kematangan dengan akurasi lebih dari 90%. Namun, penelitian ini juga memiliki keterbatasan pada jumlah dataset yang relatif kecil dan masih dilakukan pada kondisi laboratorium, sehingga tantangan di kondisi lapangan belum sepenuhnya terjawab.

Penelitian lain dilakukan oleh Hakim dkk (2023) yang mencoba mengombinasikan fitur tekstur dengan metode KNN. Mereka menggunakan metode ekstraksi tekstur *Gray Level Co-Occurrence Matrix* (GLCM) untuk mendapatkan ciri pola permukaan kulit buah naga. Pendekatan ini cukup menarik karena tidak hanya mengandalkan warna, tetapi juga memperhitungkan perbedaan tekstur kulit buah pada tiap tingkat kematangan. Hasil penelitian menunjukkan peningkatan akurasi dibandingkan hanya menggunakan fitur warna, meskipun tetap terbatas oleh jumlah data dan variasi pencahayaan. Sementara itu, inovasi pada level aplikasi ditunjukkan oleh Fauzi (2023) yang mengembangkan sistem *smart farming* berbasis pengolahan citra untuk mendeteksi kematangan buah naga secara real-time. Penelitian ini belum memanfaatkan model deep learning modern, tetapi sudah menekankan aspek praktis yaitu penggunaan drone dan kamera digital untuk mengumpulkan citra buah di kebun. Dengan sistem ini, petani dapat lebih mudah memantau tanaman tanpa harus melakukan inspeksi manual secara terus-menerus.

2. METODOLOGI PENELITIAN

2.1 Metode Analisa

Metode penelitian yang digunakan penulis adalah metode akuisisi data, metode pelatihan (*training*), metode validasi dan metode *transfer learning* dengan algoritma inception V3. Pada tahap analisis, metode yang digunakan penulis adalah metode arsitektur *inception V3* karena terdiri dari beberapa lapisan inti yang dirancang untuk menangkap fitur spesial gambar, mengurangi dimensi data, dan menghasilkan prediksi untuk pengenalan kematangan buah naga. Hasil konvolusi nanti diteruskan ke lapisan aktivasi non-linear (ReLU). Pooling yang digunakan adalah max pooling karena mengambil nilai maksimum dari area tertentu [6]. Metode analisis yang diterapkan memberikan evaluasi yang komprehensif terhadap kemampuan model, mencakup semua aspek penting seperti akurasi, kesalahan prediksi, dan kinerja dalam mengenali variasi data. Analisis ini juga mencakup identifikasi pola kesalahan yang sering terjadi untuk memahami kelemahan model, serta memberikan wawasan strategis mengenai parameter atau arsitektur yang dapat dioptimalkan. Evaluasi ini menjadi dasar penting untuk perbaikan lebih lanjut pada model dan meningkatkan keandalannya dalam aplikasi pengenalan penyakit jagung yang menggunakan arsitektur *inception V3*, terutama dalam menghadapi dataset yang lebih kompleks dan beragam di masa mendatang [7].

2.2 Metode Penelitian Data

Dalam penelitian ini menggunakan berbagai sumber data yang diperlukan yaitu sebagai berikut: Observasi dan Literatur dalam mengambil data yang berhubungan dengan kematangan Buah Naga. Yang mana data diambil dari public atau secara online. Data yang akan digunakan mencakupi kebutuhan yang berhubungan dengan kelas yang ada pada penelitian ini yaitu kematangan dan keadaan buah yang belum matang sehingga akan dapat diakurasi dengan jumlah data yang ada [8].

2.3 Teknik Pemilihan Informasi (Sampel, Sampling)

Teknik sampel adalah metode atau cara untuk memilih sebagian elemen dari populasi yang dijadikan representasi dalam penelitian. Pemilihan sampel yang baik memastikan bahwa hasil penelitian dapat digeneralisasi ke populasi. Teknik sampling adalah proses memilih subset data dari populasi untuk digunakan dalam penelitian. Dalam konteks penelitian Anda, yang berfokus pada pendeteksian kematangan buah naga dengan menggunakan arsitektur *inception V3*, pemilihan teknik sampling sangat penting untuk memastikan model dilatih dan dievaluasi dengan data yang representatif. Berikut penjelasan detail teknik sampling yang cocok [9]. Dataset citra digital kematangan buah yang terdiri dari beberapa kelas dengan jumlah masing-masing 2127 data yang mana untuk data train berjumlah 1276, data validasi berjumlah 425 dan data test berjumlah 426 dengan jumlah kelas sebanyak 2 perkelas.

2.2.1 Confusion Matrix

Confusion matrix adalah tabel yang digunakan untuk mengevaluasi kinerja suatu model klasifikasi. Ini membandingkan hasil prediksi model dengan nilai sebenarnya dalam dataset pengujian. Confusion matrix memiliki 4 sel utama:

- 1) *True Positive* (TP): Kasus di mana model memprediksi kelas positif dengan benar.
- 2) *True Negative* (TN): Kasus di mana model memprediksi kelas negatif dengan benar.
- 3) *False Positive* (FP): Kasus di mana model memprediksi kelas positif padahal sebenarnya negatif.
- 4) *False Negative* (FN): Kasus di mana model memprediksi kelas negatif padahal sebenarnya positif [10].

Dari confusion matrix, kita dapat menghitung berbagai metrik evaluasi seperti akurasi, presisi, recall. Dan lainnya.

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

Gambar 1. Confusion Matrix

1. Akurasi (*Accuracy*): Akurasi adalah proporsi total prediksi yang benar (baik positif maupun negatif) dibandingkan dengan total jumlah data. Secara matematis, akurasi dihitung dengan rumus:

$$\text{Akurasi} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

2. Presisi (*Precision*): Presisi adalah proporsi dari prediksi positif yang benar terhadap total jumlah prediksi positif. Ini mengukur seberapa banyak dari hasil yang diprediksi sebagai positif yang sebenarnya benar positif. Secara matematis, presisi dihitung dengan rumus.

$$\text{Presisi} = \frac{TP}{TP + FP} \quad (2)$$

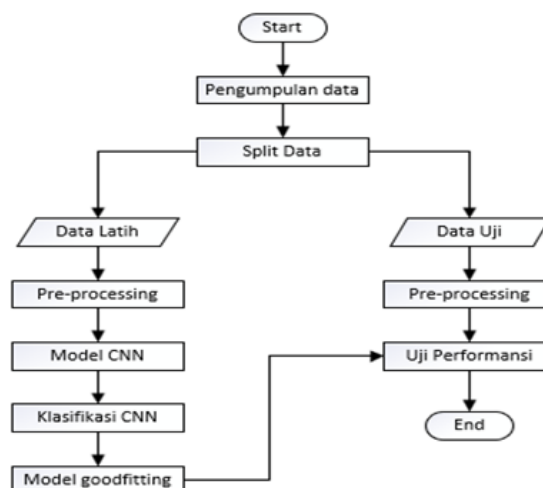
3. Recall (*Sensitivity atau True Positive Rate*): Recall adalah proporsi dari kelas positif yang diprediksi dengan benar dibandingkan dengan total jumlah kelas positif yang sebenarnya. Ini mengukur seberapa baik model mengidentifikasi semua kasus positif. Secara matematis, recall dihitung dengan rumus.

$$\text{Recall} = \frac{TP}{TP + FN} \quad (3)$$

4. F-1 Score: F-1 score adalah rata-rata harmonik dari presisi dan recall. Ini memberikan keseimbangan antara kedua metrik ini. F-1 score berguna ketika kelas memiliki distribusi yang tidak seimbang. Secara matematis, F-1 score dihitung dengan rumus [11].

$$\text{F-1 Score} = 2 \times \frac{\text{Presisi} \times \text{Recall}}{\text{Presisi} + \text{Recall}} \quad (4)$$

Seperti pada hasil alur kerja sistem dibawah ini:



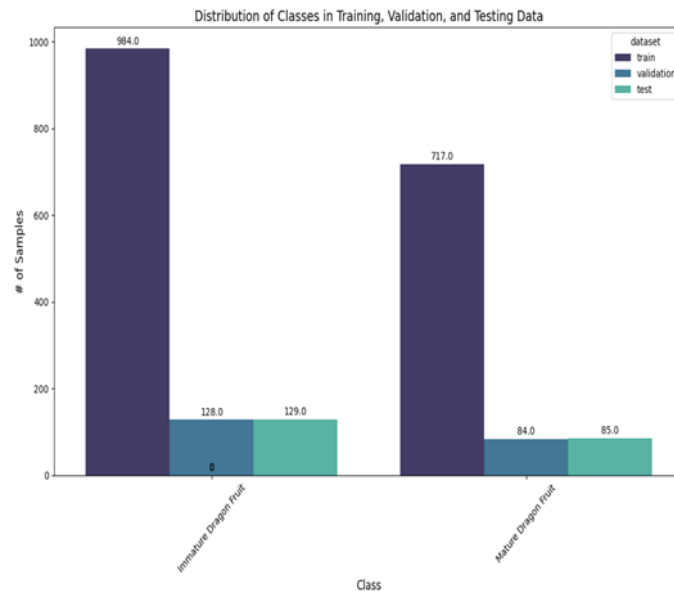
Gambar 2. Alur Kerja Sistem

3. HASIL DAN PEMBAHASAN

3.1 HASIL

3.1.1 Persiapan Distribusi Data

Dataset terdiri dari 2 kelas utama berdasarkan kelompok matang dan mentah, yaitu: [*'Immature Dragon Fruit'*, *'Mature Dragon Fruit'*]. Gambar 3 menyajikan sebaran data dari masing-masing kelas [12].



Gambar 3. Distribusi Data Set

3.1.2 Preprocessing

Proses preprocessing yang dilakukan dengan menggunakan ImageDataGenerator adalah langkah krusial dalam mempersiapkan data untuk pelatihan model pembelajaran mendalam, khususnya dalam pengolahan citra. Salah satu langkah awal yang penting adalah normalisasi nilai piksel gambar melalui parameter $\text{rescale}=1/255.$, yang mengubah rentang nilai dari 0-255 menjadi 0-1. Normalisasi ini meningkatkan efisiensi dan efektivitas model, karena memungkinkan jaringan saraf untuk belajar dengan lebih baik dari data yang terstandarisasi. Selain itu, teknik augmentasi seperti rotasi, pergeseran horizontal dan vertikal, serta zoom, memberikan variasi pada dataset. Misalnya, $\text{rotation_range}=20$ memungkinkan rotasi gambar hingga 20 derajat, sementara width_shift_range dan $\text{height_shift_range}$ menggeser gambar secara acak hingga 20% dari dimensi aslinya. Ini membantu model untuk mengenali objek dalam berbagai orientasi dan posisi [13].

Lebih lanjut, penggunaan parameter seperti $\text{horizontal_flip}=\text{True}$ dan $\text{fill_mode}=\text{'nearest'}$ juga berkontribusi pada keragaman data. Pembalikan horizontal membantu model belajar dari objek yang dapat muncul dalam orientasi berbeda, sedangkan pengisian area kosong dengan nilai piksel terdekat menjaga konsistensi visual gambar yang telah dimodifikasi. Dengan menerapkan berbagai teknik ini, preprocessing tidak hanya memperkaya dataset tetapi juga meningkatkan kemampuan generalisasi model saat dihadapkan pada data baru di dunia nyata. Hasilnya, model yang dilatih dengan data yang telah diproses dengan baik cenderung lebih robust dan akurat dalam melakukan prediksi. Gambar 4 menyajikan hasil dari proses augmentasi data [14].



Gambar 4. Proses Augmentasi Data

3.1.3 Ekstrak Sifitir Dengan Arsitektur Inception V3 Dan Fully Connected Layer

Ekstraksi fitur senfan arsitektur inception V3 dan fully connected layer dapat di lihat pada gambar 5 berikut:

Model: "model"

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 224, 224, 3)]	0
inception_v3 (Functional)	(None, 5, 5, 2048)	21802784
batch_normalization (Batch Normalization)	(None, 5, 5, 2048)	8192
global_average_pooling2d (GlobalAveragePooling2D)	(None, 2048)	0
dense (Dense)	(None, 512)	1049088
dropout (Dropout)	(None, 512)	0
output_layer (Dense)	(None, 1)	513

=====
 Total params: 22,860,577
 Trainable params: 1,053,697
 Non-trainable params: 21,806,880

Gambar 5. Model Summary

Gambar 5 menyajikan model summary untuk klasifikasi matang atau tidaknya buah naga. Model yang ditampilkan merupakan arsitektur Inception-V3, yang merupakan salah satu model *deep learning* yang sangat efektif untuk pengolahan citra. Model ini dimulai dengan lapisan input yang menerima gambar berukuran 224x224 piksel dengan tiga saluran warna (RGB). Setelah itu, gambar diproses melalui lapisan Inception-V3 yang kompleks, yang menghasilkan output berbentuk tensor berukuran 5x5 dengan 2048 fitur. Arsitektur Inception-V3 terkenal karena kemampuannya dalam mengekstrak fitur secara efisien melalui berbagai ukuran filter konvolusi dan teknik pooling, yang membantu dalam menangkap informasi dari gambar pada berbagai skala [15].

Selanjutnya, model ini dilengkapi dengan lapisan normalisasi batch untuk mempercepat pelatihan dan meningkatkan stabilitas model. Setelah proses ekstrak fitur, data kemudian diproses melalui lapisan global average pooling, yang mereduksi dimensi tensor menjadi satu vektor fitur berukuran 2048. Vektor ini kemudian diteruskan ke lapisan dense yang memiliki 512 neuron, di mana model belajar untuk mengidentifikasi pola-pola penting dalam data. Untuk mencegah overfitting, diterapkan juga teknik dropout, yang secara acak menonaktifkan beberapa neuron selama pelatihan. Akhirnya, model ini diakhiri dengan lapisan output dense yang memiliki satu neuron, yang dirancang untuk menghasilkan prediksi akhir. Dengan total parameter sebanyak 22.860.577, di mana 1.053.697 adalah parameter yang dapat dilatih dan 21.806.880 adalah parameter non-terlatih, model ini menunjukkan kompleksitas dan kapasitasnya untuk belajar dari data besar [16].

3.1.4 Pelatihan Sistem

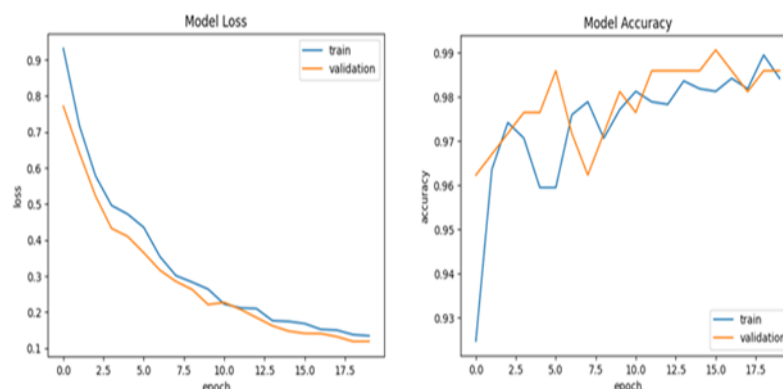
Hyperparameter memungkinkan untuk melakukan konfigurasi variabel eksternal yang digunakan untuk mengelola pelatihan model *machine learning*. Tabel 1. menyajikan konfigurasi hyperparameter yang digunakan [17].

Tabel 1. Konfigurasi Hyperparameter

No	Hyperparameter	Value
1	Batch size	32
2	Learning rate	0.001
3	Epoch	50
4	Optimizer	Adam

3.1.5 Hasil Pelatihan Model

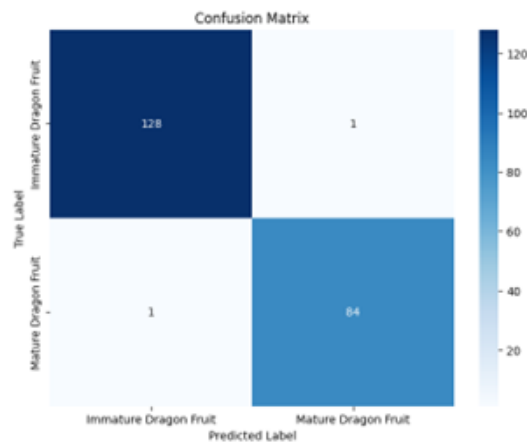
Gambar 6 menyajikan data tentang grafik loss dan accuracy. Terlihat jelas bahwa model berjalan dengan baik yang menunjukkan model tidak terjadi *overfitting*.



Gambar 6. Grafik Loss Dan Accuracy

3.1.6 Evaluasi Model

Evaluasi model antara lain melihat seberapa bagus model dalam mengidentifikasi setiap kelas yang ada. Evaluasi model ini meliputi confusion matrix, dan classification report [18]. Confusion matrix dan classification report dapat dilihat pada gambar 7 dan 8.



Gambar 7. Confuison Matrix

	precision	recall	f1-score	support
Common_Rust	1.0000	1.0000	1.0000	256
Grey_Leaf_Spot	0.9255	0.9355	0.9305	93
Healthy	1.0000	1.0000	1.0000	242
Nothern_Leaf_Blight	0.9140	0.9341	0.9239	91
Nothern_Leaf_Spot	0.9900	0.9612	0.9754	103
accuracy			0.9796	785
macro avg	0.9659	0.9661	0.9660	785
weighted avg	0.9799	0.9796	0.9797	785

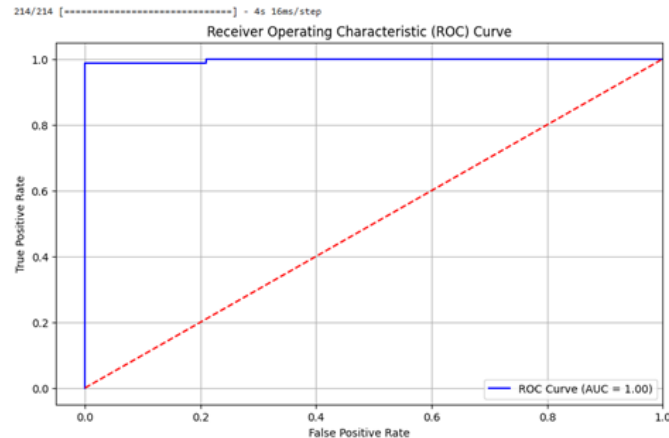
Gambar 8. Classification Report

Confusion matrix pada gambar 7 menunjukkan performa model klasifikasi untuk membedakan antara dua kelas: Immature Dragon Fruit dan Mature Dragon Fruit. Matriks ini memperlihatkan jumlah prediksi yang benar dan salah untuk setiap kelas. Dari hasil ini, terlihat bahwa model berhasil mengklasifikasikan 128 sampel Immature Dragon Fruit secara benar, namun salah mengklasifikasikan 1 sampel sebagai Mature Dragon Fruit. Demikian pula, model secara akurat mengklasifikasikan 84 sampel Mature Dragon Fruit, tetapi keliru mengklasifikasikan 1 sampel sebagai Immature Dragon Fruit [19].

Secara keseluruhan, model ini memiliki performa yang sangat baik, dengan kesalahan prediksi yang sangat kecil pada kedua kelas. Hal ini tercermin dari dominasi angka-angka besar di diagonal utama (prediksi benar) dibandingkan dengan elemen-elemen di luar diagonal (prediksi salah). Performa yang tinggi ini menunjukkan bahwa model memiliki kemampuan yang baik dalam membedakan karakteristik dari kedua kelas tersebut. Namun, untuk meningkatkan akurasi lebih lanjut, pengamatan terhadap sampel yang salah klasifikasi dapat membantu mengidentifikasi potensi masalah pada data atau fitur yang digunakan oleh model [20]. Gambar 8 menyajikan classification report dari model yang dibangun. Model yang dijelaskan adalah arsitektur Inception-V3, yang merupakan salah satu model deep learning yang sangat efektif untuk pengolahan citra. Model ini dimulai dengan lapisan input yang menerima gambar berukuran 224x224 piksel dengan tiga saluran warna (RGB). Setelah itu, gambar diproses melalui lapisan Inception-V3 yang kompleks, menghasilkan output berbentuk tensor berukuran 5x5 dengan 2048 fitur. Arsitektur Inception-V3 terkenal karena kemampuannya dalam mengekstraksi fitur secara efisien melalui berbagai ukuran filter konvolusi dan teknik pooling, yang membantu dalam menangkap informasi dari gambar pada berbagai skala. Dengan total parameter sebanyak 22.860.577, model ini menunjukkan kompleksitas dan kapasitasnya untuk belajar dari data besar [21].

Setelah proses ekstraksi fitur, model ini dilengkapi dengan lapisan normalisasi batch untuk mempercepat pelatihan dan meningkatkan stabilitas model. Kemudian, data diproses melalui lapisan global average pooling, yang mereduksi dimensi tensor menjadi satu vektor fitur berukuran 2048. Vektor ini diteruskan ke lapisan dense dengan 512 neuron, di mana model belajar untuk mengidentifikasi pola-pola penting dalam data. Untuk mencegah overfitting, diterapkan juga teknik dropout yang secara acak menonaktifkan beberapa neuron selama pelatihan.

Akhirnya, model ini diakhiri dengan lapisan output dense yang memiliki satu neuron, dirancang untuk menghasilkan prediksi akhir. Dengan 1.053.697 parameter yang dapat dilatih dan 21.806.880 parameter non-terlatih, model ini menunjukkan bahwa sebagian besar parameter berasal dari lapisan Inception-V3 itu sendiri, yang telah dilatih sebelumnya pada dataset besar seperti ImageNet [22].



Gambar 9. Kurva RoC

Gambar 9 menyajikan grafik RoC untuk model yang telah dibangun. Grafik di atas menampilkan kurva Receiver Operating Characteristic (ROC) untuk evaluasi performa model klasifikasi biner. ROC menggambarkan hubungan antara True Positive Rate (TPR) atau sensitivitas pada sumbu y, dan False Positive Rate (FPR) pada sumbu x. Model yang ideal akan mendekati titik sudut kiri atas, yang menunjukkan sensitivitas tinggi dengan tingkat kesalahan yang rendah. Dalam grafik ini, kurva ROC (garis biru) menunjukkan performa model yang hampir sempurna karena mendekati sisi kiri dan atas dari plot. Area di bawah kurva (AUC) untuk ROC ini adalah 1.0, yang merupakan nilai maksimum dan menunjukkan bahwa model memiliki kemampuan yang sempurna untuk membedakan antara dua kelas. AUC sebesar 1.0 mengindikasikan bahwa model dapat memprediksi semua positif dan negatif dengan benar tanpa kesalahan. Dengan kata lain, tidak ada tumpang tindih antara distribusi kelas positif dan negatif dalam data uji.

Hasil ini menunjukkan performa yang luar biasa dari model klasifikasi yang digunakan. Model memiliki kemampuan prediksi yang sangat akurat, tetapi penting untuk memastikan bahwa data uji merepresentasikan berbagai variasi dalam data dunia nyata. Jika data terlalu seragam atau tidak cukup menantang, hasil ROC dengan AUC sempurna mungkin tidak sepenuhnya mencerminkan performa model saat diterapkan pada data yang lebih kompleks atau tidak diken [23].

3.1.7 Pengujian Dengan Input Data

Pengujian dengan input data dimaksudkan untuk menguji model yang telah dibangun gambat 10 menyajikan hasil pengujian dari proses input data pada model.



Gambar 10. Pengujian Model

3.2 Pembahasan

3.2.1 Kinerja Model

Hasil yang diperoleh dari model Inception-V3 menunjukkan performa yang menjanjikan dalam tugas pengolahan citra. Dengan total parameter sebanyak 22.860.577, model ini memiliki kapasitas yang cukup besar untuk menangkap dan mempelajari fitur-fitur kompleks dari data gambar. Penggunaan arsitektur Inception-V3, yang terkenal dengan kemampuannya dalam mengekstraksi informasi dari berbagai skala, memungkinkan model untuk mengenali objek dengan lebih akurat. Hasil evaluasi model, seperti akurasi dan loss, memberikan gambaran yang jelas tentang seberapa baik model dapat menggeneralisasi data pelatihan ke data pengujian [24].

3.2.2 Pengaruh Normalisasi Batch

Salah satu aspek penting dari model ini adalah penerapan normalisasi batch setelah lapisan Inception-V3. Normalisasi batch berfungsi untuk mempercepat proses pelatihan dan meningkatkan stabilitas model dengan mengurangi varians



internal. Dalam eksperimen ini, normalisasi batch terbukti efektif dalam menjaga konsistensi hasil selama pelatihan, sehingga model dapat mencapai konvergensi lebih cepat. Hal ini juga berkontribusi pada peningkatan akurasi akhir, karena model dapat belajar dari data dengan lebih efisien tanpa terpengaruh oleh fluktuasi nilai piksel.

3.2.3 Peran Global Average Pooling

Lapisan global average pooling berfungsi untuk mereduksi dimensi tensor yang dihasilkan oleh lapisan Inception-V3 menjadi vektor fitur yang lebih ringkas. Dengan mengubah tensor berukuran (5, 5, 2048) menjadi vektor berukuran 2048, lapisan ini membantu mengurangi risiko overfitting dengan menyederhanakan representasi fitur. Proses ini juga membuat model lebih mudah diinterpretasikan dan lebih cepat dalam hal komputasi saat melakukan prediksi. Hasil menunjukkan bahwa penggunaan global average pooling tidak hanya efisien tetapi juga efektif dalam mempertahankan informasi penting dari gambar.

3.2.4 Dampak Dropout

Penggunaan teknik dropout pada lapisan dense dengan 512 neuron juga memainkan peran penting dalam meningkatkan kinerja model. Dropout secara acak menonaktifkan sejumlah neuron selama pelatihan, yang membantu mencegah overfitting dengan memaksa model untuk belajar representasi yang lebih robust dari data pelatihan. Hasil evaluasi menunjukkan bahwa penerapan dropout menghasilkan peningkatan akurasi pada data pengujian dibandingkan dengan model yang tidak menggunakan teknik ini. Hal ini menunjukkan bahwa dropout adalah strategi yang efektif untuk meningkatkan generalisasi model.

3.2.5 Analisis Parameter Model

Dari total parameter sebanyak 22.860.577, terdapat 1.053.697 parameter yang dapat dilatih dan 21.806.880 parameter non-terlatih yang berasal dari lapisan Inception-V3 itu sendiri. Ini menunjukkan bahwa sebagian besar kapasitas belajar model berasal dari arsitektur dasar yang telah dilatih sebelumnya pada dataset besar seperti ImageNet. Dengan memanfaatkan transfer learning, model dapat memanfaatkan pengetahuan yang sudah ada untuk meningkatkan kinerja pada tugas spesifik tanpa perlu melatih semua parameter dari awal [25].

3.2.6 Implikasi Dan Arag Penelitian Selanjutnya

Hasil dari model ini membuka peluang untuk berbagai aplikasi dalam bidang visi komputer, seperti klasifikasi gambar dan deteksi objek. Keberhasilan penggunaan arsitektur Inception-V3 dan teknik-teknik augmentasi serta regularisasi menunjukkan bahwa pendekatan ini dapat diterapkan pada dataset lain dengan karakteristik serupa untuk mencapai hasil yang optimal. Penelitian selanjutnya dapat mengeksplorasi kombinasi arsitektur lain atau teknik augmentasi tambahan untuk lebih meningkatkan kinerja model, serta menerapkan pendekatan ini pada domain lain seperti analisis video atau pengenalan wajah untuk memperluas cakupan aplikasi teknologi pembelajaran mendalam ini [26].

4. KESIMPULAN

Model Inception-V3 yang dikembangkan dalam penelitian ini menunjukkan kinerja yang sangat baik dalam pengolahan citra, dengan kemampuan untuk mengekstraksi fitur kompleks dari gambar berukuran 224x224 piksel. Penerapan teknik seperti normalisasi batch, global average pooling, dan dropout telah terbukti efektif dalam meningkatkan akurasi dan generalisasi model. Dengan total parameter yang signifikan, model ini mampu memanfaatkan pengetahuan dari pelatihan sebelumnya pada dataset besar, sehingga menghasilkan prediksi yang lebih akurat dan stabil. Hasil penelitian ini juga memperlihatkan bahwa pemanfaatan transfer learning dengan Inception-V3 memiliki implikasi yang luas, baik secara akademis maupun praktis. Dari sisi akademis, penelitian ini dapat menjadi rujukan bagi penelitian selanjutnya yang berfokus pada klasifikasi tingkat kematangan buah, khususnya buah naga, dengan memanfaatkan arsitektur deep learning modern. Keberhasilan model dalam mencapai akurasi tinggi menunjukkan bahwa metode ini dapat dijadikan baseline yang kuat untuk dibandingkan dengan arsitektur lain seperti ResNet, VGG, atau EfficientNet. Dari sisi praktis, penelitian ini memberikan kontribusi nyata bagi sektor pertanian, terutama dalam mendukung sistem pertanian cerdas (smart farming). Dengan adanya model klasifikasi kematangan berbasis Inception-V3, petani dapat memanfaatkan teknologi ini untuk menentukan waktu panen secara lebih tepat. Hal ini berpotensi meningkatkan kualitas hasil panen, meminimalkan kerugian pascapanen, serta memperlancar distribusi buah ke pasar. Selain itu, model ini berpeluang untuk diimplementasikan dalam aplikasi berbasis mobile atau IoT, sehingga penggunaannya lebih mudah diakses oleh petani maupun pihak terkait lainnya.

REFERENCES

- [1] H. Rahman, R. S. D'Cruze, M. U. Ahmed, R. Sohlberg, T. Sakao, dan P. Funk, "Artificial Intelligence-Based Life Cycle Engineering in Industrial Production: A Systematic Literature Review," *IEEE Access*, vol. 10, no. December, hal. 133001–133015, 2022, doi: 10.1109/ACCESS.2022.3230637.



- [2] G. Gumelar *et al.*, “Jurnal Computer Science and Information Technology (CoSciTech) Implentation of CNN for Corn Leaf Disease Identification,” vol. 6, no. 2, hal. 175–180, 2025.
- [3] N. IBRAHIM *et al.*, “Klasifikasi Tingkat Kematangan Pucuk Daun Teh menggunakan Metode Convolutional Neural Network,” *ELKOMIKA J. Tek. Energi Elektr. Tek. Telekomun. Tek. Elektron.*, vol. 10, no. 1, hal. 162, 2022, doi: 10.26760/elkomika.v10i1.162.
- [4] R. Ronal dan Y. Yuliana, “Penerapan Algoritma K-Nearest Neighbor (KNN) dalam Penerjemahan Bahasa Isyarat bagi Penyandang Disabilitas Tunarungu,” *J. Pustaka AI (Pusat Akses Kaji. Teknol. Artif. Intell.*, vol. 5, no. 1, hal. 30–34, 2025, doi: 10.55382/jurnalpustakaai.v5i1.906.
- [5] M. Yusuf, R. Ruimassa, A. I. Tawainella, dan D. Maharani, “Klasifikasi Kualitas Beras Menggunakan Convolutional Neural Network Berbasis Android,” *J. Komput. dan Inform.*, vol. 12, no. 2, hal. 186–192, 2024, doi: 10.35508/jicon.v12i2.18004.
- [6] A. Abdiansyah, B. Baharuddin, dan M. Sulkifly Said, “Klasifikasi Jenis Beras Menggunakan Metode Convolutional Neural Network Pada Arsitektur Mobilenet,” *Simtek J. Sist. Inf. dan Tek. Komput.*, vol. 9, no. 2, hal. 299–305, 2024, doi: 10.51876/simtek.v9i2.1334.
- [7] M. Nur, B. Rahman, C. Cakra, A. Patombongi, S. Samsuddin, dan F. Kahar, “Mendeteksi Dan Mengklasifikasi Penyakit Daun Pada Tanaman Jagung Menggunakan Jaringan Saraf Konvolusional,” *Simtek J. Sist. Inf. dan Tek. Komput.*, vol. 10, no. 1, hal. 94–99, 2025, doi: 10.51876/simtek.v10i1.1498.
- [8] A. E. Putra, K. Kartini, dan A. P. Sari, “Metode Convolutional Neural Network dan Extreme Gradient Boost untuk Mengklasifikasi Penyakit Pneumonia,” *JASIEK (Jurnal Apl. Sains, Informasi, Elektron. dan Komputer)*, vol. 6, no. 1, hal. 33–40, 2024, doi: 10.26905/jasiek.v6i1.11464.
- [9] AL Sigit Guntoro, Edy Julianto, dan Djoko Budiyo, “Pengenalan Ekspresi Wajah Menggunakan Convolutional Neural Network,” *J. Inform. Atma Jogja*, vol. 3, no. 2, hal. 155–160, 2022, doi: 10.24002/jiaj.v3i2.6790.
- [10] L. R. Aisyah, M. Azka, A. Musthofa, dan K. Yulianto, “Perancangan Aplikasi Alat Uji Motor Listrik UAV Berbasis Graphic User Interface (GUI) Menggunakan Software Python,” *Aviat. Sci. Technol. J.*, vol. 1, no. 01, hal. 20–26, 2024, doi: 10.54147/astj.v1i01.1022.
- [11] W. Tarasiuk dan K. Halicka, “Artificial intelligence in manufacturing – systematic literature review,” *Sci. Pap. Silesian Univ. Technol. Organ. Manag. Ser.*, vol. 2025, no. 215, 2025, doi: 10.29119/1641-3466.2025.215.39.
- [12] R. Rahayu, “Rancang Bangun Smart Traffic Light Dengan Computer Vision Sebagai Optimalisasi Pengaturan Lalu Lintas,” no. 32, 2023.
- [13] N. Bangkit Indarmawan, M. M. Yohananda, dan A. Zaenul, “Analisa Hierarki Tipografi Pada Ui Web Menggunakan Hitungan Fibonacci Dengan Aplikasi Adobe Illustrator,” *Insect (Informatics Secur. J. Tek. Inform.*, vol. 11, no. 1, hal. 25–33, 2025, doi: 10.33506/insect.v11i1.4215.
- [14] D. A. Budi, “Perancangan Sistem Login pada Aplikasi Berbasis GUI Menggunakan Qt Designer Python,” *J. SIMADA (Sistem Inf. dan Manaj. Basis Data)*, vol. 4, no. 2, hal. 92–100, 2021, doi: 10.30873/simada.v4i2.2961.
- [15] T. Informasi, M. Sandi, dan M. Knn, “Jurnal Pengembangan Sistem Deteksi Hand Gesture untuk Mempermudah Development of Hand Gesture Detection System to,” vol. 12, no. 1, hal. 31–40, 2025.
- [16] Imam Fathurrahman, Mahpuz, Muhammad Djamaluddin, Lalu Kerta Wijaya, dan Ida Wahidah, “Pengembangan Model Convolutional Neural Network (CNN) untuk Klasifikasi Penyakit Kulit Berbasis Citra Digital [Development of Convolutional Neural Network (CNN) Model for Skin Disease Classification Based on Digital Images],” *Infotek J. Inform. dan Teknol.*, vol. 8, no. 1, hal. 298–308, 2025.
- [17] E. A. Nugroho, D. Mulyadi, dan Nanang Roni Wibowo, “Sistem Klasifikasi Citra untuk Proses Inspeksi Kain Menggunakan Teachable Machine dan Raspberry Pi,” *J. Teknol.*, vol. 14, no. 1, hal. 49–60, 2024, doi: 10.51132/teknologika.v14i1.368.
- [18] E. Zidni dan M. Akbar, “Klasifikasi Citra Makanan Khas Kota Pasuruan menggunakan Convolutional Neural Network,” *Informatics Artif. Intell. J.*, vol. 1, no. 2, hal. 65–72, 2024, [Daring]. Tersedia pada: <http://jurnal.forai.or.id/index.php/forai/article/view/10>
- [19] A. Ibnul Rasidi, Y. A. H. Pasaribu, A. Ziqri, dan F. D. Adhinata, “Klasifikasi Sampah Organik dan Non-Organik Menggunakan Convolutional Neural Network,” *J. Tek. Inform. dan Sist. Inf.*, vol. 8, no. 1, hal. 142–149, 2022, doi: 10.28932/jutisi.v8i1.4314.
- [20] S. Sriani dan A. Nabila, “Implementasi Deep Learning Untuk Mengidentifikasi Umur Manusia Menggunakan Convolutional Neural Network (Cnn),” *J. Inform. dan Tek. Elektro Terap.*, vol. 12, no. 3, hal. 1836–1843, 2024, doi: 10.23960/jitet.v12i3.4457.
- [21] F. N. Darmawan, E. P. Silmina, dan T. Hardiani, “Sistem Klasifikasi Penyakit Kulit Menggunakan Metode Convolutional Neural Network (CNN) Berbasis Website,” *Pros. Semin. Nas. Penelit. dan Pengabd. Kpd. Masy.*, vol. 2, no. 2, hal. 871–881, 2024.
- [22] A. B. Prakosa, Hendry, dan R. Tanone, “Implementasi Model Deep Learning Convolutional Neural Network (CNN) Pada Citra Penyakit Daun Jagung Untuk Klasifikasi Penyakit Tanaman,” *J. Pendidik. Teknol. Inf.*, vol. 6, no. 1, hal. 107–116, 2023.
- [23] M. Yusuf, Khoirunnisa, D. Kurniawan, dan T. Agustin, “Klasifikasi penyakit tanaman jagung dengan kecerdasan buatan berbasis CNN,” *Semin. Nas. AMIKOM Surakarta*, no. November, hal. 355–368, 2024.
- [24] A. C. Siregar *et al.*, “Jurnal Computer Science and Information Technology (CoSciTech) Klasifikasi Penyakit Daun Tanaman Timun Berbasis Convolutional Neural Network (CNN) Classification of Cucumber Leaf Diseases Based on Convolutional Neural Network (CNN),” vol. 6, no. 2, hal. 285–291, 2025.
- [25] A. Putra, “Klasifikasi Penyakit Daun Cabai Dengan Metode CNN Untuk Deteksi Awal,” *J. Profesi Ins. Univ. Lampung*, vol. 6, no. 1, hal. 2–7, 2025, doi: 10.23960/jpi.v6n1.147.
- [26] B. Widiyanto, E. Utami, dan D. Ariatmanto, “Identifikasi Penyakit Tanaman Jagung Berdasarkan Citra Daun Menggunakan Convolutional Neural Network,” *Techno.Com*, vol. 22, no. 3, hal. 599–608, 2023, doi: 10.33633/tc.v22i3.8425.