

Klasifikasi Penyakit daun pada tanaman jagung menggunakan Convolutional Neural Network

Ilsa Hidayat, SyafriArlis

Fakultas Ilmu Komputer, Program Studi Magister Teknik Informatika
Universitas Putra Indonesia YPTK Padang, Indonesia
Kota Padang, Indonesia
Ilsahidayat2202@gmail.com, Safri_alis@upiyptk.ac.id

Abstract- This study aims to evaluate the performance of Convolutional Neural Network (CNN) models in detecting pests on caisim (Chinese mustard) plants using different architectural approaches, namely CNN from scratch, VGG16, and Xception. The dataset used consists of 1,000 images classified into several disease categories and a healthy class. Five experiments were conducted to compare the effectiveness of the models based on evaluation metrics such as accuracy, precision, recall, and F1-score. The results show that CNN models trained from scratch produced varying levels of performance. The first and second experiments experienced underfitting, with accuracies of 30.25% and 62.29%, respectively. A significant improvement was observed in the third experiment, achieving an accuracy of 83.73% and an F1-score of 0.82, indicating that the model began to better recognize data patterns. The best performance was achieved in the fourth (VGG16) and fifth (Xception) experiments, with accuracies of 91.41% and 92.12%, respectively, and balanced precision, recall, and F1-score values above 0.90. Factors contributing to model success include an optimal proportion of training data, appropriate architectural selection, hyperparameter tuning, and the use of callbacks such as early stopping and model checkpoint. This study demonstrates that selecting the appropriate CNN architecture can significantly improve the accuracy of image classification systems for pest detection in plants.

Keywords: Classification, Convolutional Neural Network, Corn Leaves.

Abstrak- Penelitian ini bertujuan untuk mengevaluasi performa model Convolutional Neural Network (CNN) dalam mendekripsi penyakit pada jagung dari daunnya menggunakan pendekatan arsitektur yang berbeda, yaitu CNN dari awal, VGG16, dan Xception. Data yang digunakan terdiri dari 577,614 gambar yang dibagi ke dalam beberapa kelas penyakit dan kondisi sehat. Lima percobaan dilakukan untuk membandingkan efektivitas model berdasarkan metrik evaluasi seperti akurasi, precision, recall, dan F1-score. Hasil penelitian menunjukkan bahwa model CNN yang dilatih dari awal menghasilkan performa bervariasi. Percobaan pertama dan kedua mengalami underfitting dengan akurasi masing-masing sebesar 30.25% dan 62.29%. Peningkatan signifikan terjadi pada percobaan ketiga dengan akurasi 83.73% dan F1-score 0.82, yang menunjukkan model mulai mampu mengenali pola data dengan lebih baik. Performansi terbaik diperoleh pada percobaan keempat (VGG16) dan kelima (Xception), dengan akurasi masing-masing sebesar 91.41% dan 92.12%, serta nilai precision, recall, dan F1-score yang seimbang di atas 0.90. Faktor-faktor yang berpengaruh terhadap keberhasilan model meliputi proporsi data latih yang optimal, pemilihan arsitektur yang sesuai, pengaturan hyperparameter, serta penggunaan callback seperti early stopping dan model checkpoint. Penelitian ini menunjukkan bahwa pemanfaatan arsitektur CNN yang tepat dapat meningkatkan akurasi sistem klasifikasi gambar untuk deteksi penyakit pada tanaman jagung.

Kata Kunci: Klasifikasi, Convolutional Neural Network, Daun Jagung

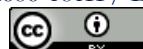
1. Pendahuluan

Sebagai komoditas pangan utama secara global jagung memiliki kontribusi penting dalam sistem pertanian dan ketahanan pangan nasional di Indonesia [1]. Jagung manis merupakan salah satu jenis jagung hasil budidaya yang umum ditanam sebagai komoditas palawijaya. Tanaman ini berasal dari famili Poaceae dan termasuk dalam genus *Zea*, dengan spesies *Zea mays* varietas *saccharata* [2]. Pertumbuhan optimal tanaman ini dicapai pada wilayah dengan intensitas curah hujan rendah hingga sedang serta

didukung oleh pola presipitasi yang relatif konsisten sepanjang tahun [3]. Wilayah budidaya jagung di Indonesia mencakup berbagai provinsi, seperti Sumatera Utara, Sumatera Selatan, Lampung, Jawa Tengah, Jawa Timur, Nusa Tenggara, Sulawesi Utara, Sulawesi Selatan, serta Maluku [4]. Data dalam jangka waktu lima tahun ke belakang menunjukkan bahwa volume panen jagung domestik terus mengalami pertumbuhan positif [5]. Namun demikian, produktivitas tanaman jagung masih

Vol.16 no.2 | Desember 2025

EXPLORE : ISSN: 2087-2062, Online ISSN: 2686-181X / DOI: <http://dx.doi.org/10.36448/jsit.v16i2.4448>



This work is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/)

menghadapi tantangan, terutama akibat serangan penyakit yang menyerang bagian daun pada tanaman jagung [6]. Penyakit daun seperti blight (hawar), common rust (karat daun umum) dan gray leaf spot (bercak daun abu-abu) dapat menyebabkan penurunan kualitas dan kuantitas hasil panen secara signifikan [7]. Identifikasi penyakit daun secara manual oleh petani atau tenaga ahli seringkali memakan waktu, bersifat subjektif, dan tidak efisien jika dilakukan dalam skala besar [8]. Oleh karena itu, dibutuhkan solusi berbasis teknologi yang mampu mengidentifikasi dan mengklasifikasikan penyakit daun jagung secara cepat, akurat, dan otomatis [9].

Salah satu pendekatan teknologi yang kini berkembang pesat dalam bidang pengolahan citra dan klasifikasi visual yakni *Convolutional Neural Network* [10]. CNN dikelompokkan sebagai bagian dari kategori arsitektur *deep learning* yang dirancang secara spesifik untuk menganalisis dan mengidentifikasi pola pada data visual, terutama gambar atau citra. Dalam penelitian ini, CNN digunakan sebagai metode untuk memproses gambar daun jagung dan mendeteksi jenis penyakit tertentu yang muncul, bukan sekadar mengenali daun yang rusak secara umum. CNN telah terbukti unggul dalam berbagai studi klasifikasi gambar karena kemampuannya mengekstraksi fitur visual secara otomatis tanpa perlu intervensi manusia [11]. Sejumlah penelitian sebelumnya telah membuktikan keunggulan CNN dalam klasifikasi penyakit tanaman.

Penelitian oleh Cendekia Luthfieta Nazalia dkk. menggunakan metode Convolutional Neural Network (CNN) untuk mendeteksi hama pada daun caisim. Proses penelitian mencakup pengumpulan dan preprocessing data, augmentasi citra, pemodelan CNN dengan tiga percobaan, serta evaluasi menggunakan Confusion Matrix. Dataset dibagi menjadi beberapa skema, yaitu Eksperimen A (40% training, 30% validation, 30% testing), Eksperimen B (80% training, 10% validation, 10% testing), dan Eksperimen C (90% training, 5% validation, 5% testing). Hasil penelitian ini sejalan dengan studi sebelumnya yang menegaskan bahwa pembagian dataset dan arsitektur CNN berpengaruh signifikan terhadap performa model klasifikasi citra.

Selain itu penelitian yang dilakukan oleh Timilsina dkk, mengembangkan model *CNDD-Net*, yaitu CNN berbobot ringan berbasis perhatian (*attention-based CNN*) yang mampu mengklasifikasikan tujuh kelas gangguan daun jagung, termasuk defisiensi nutrisi dan penyakit daun. Model ini berhasil mencapai rata-rata akurasi sebesar 96,71% dengan jumlah parameter hanya 48.041 dan ukuran model sebesar 0,24 MB. Kelebihan utama dari model ini adalah efisiensinya secara komputasional, menjadikannya sangat cocok untuk diterapkan di perangkat portabel guna mendukung pertanian berkelanjutan secara langsung di lapangan[12].

Sementara, penelitian Yin dkk, menunjukkan bahwa model *DISE-Net*, arsitektur CNN dengan *dilated-inception* module dan attention mechanism, mampu mengklasifikasikan penyakit *maize small leaf spot* secara akurat pada lima tingkat keparahan. Model dilatih pada 7.500 citra hasil augmentasi dari 1.268 gambar daun jagung lapangan dengan variasi kondisi visual. Dengan

akurasi uji 97,12%, *DISE-Net* melampaui kinerja *VGG16*, *ResNet50*, *MobileNetV2*, dan *DenseNet121*. Visualisasi Grad-CAM mengonfirmasi kemampuan model dalam mengenali area gejala secara presisi, menjadikannya efektif untuk klasifikasi fine-grained penyakit daun jagung [13].

Penelitian oleh Shah dkk, menunjukkan bahwa sistem deteksi penyakit tanaman berbasis CNN dengan model *EfficientNet-B3* mampu mengidentifikasi penyakit daun secara otomatis dengan akurasi tinggi sebesar 98,80%. Integrasi sistem ke dalam aplikasi mobile dan web, serta penggunaan citra dari drone dan kamera ponsel, membuktikan efektivitasnya dalam kondisi lapangan. Selain akurat, sistem ini juga praktis dan mudah digunakan, sehingga berpotensi mendukung monitoring pertanian presisi secara luas, terutama oleh petani di lapangan tanpa memerlukan keahlian teknis [14].

Dari berbagai penelitian sebelumnya ditegaskan bahwa Convolutional Neural Network (CNN) merupakan pendekatan yang potensial dan efektif dalam mendeteksi serta mengklasifikasikan penyakit pada tanaman jagung secara otomatis dan akurat [15]. Meskipun sejumlah penelitian telah mengusulkan arsitektur CNN dengan akurasi tinggi, sebagian besar masih terbatas pada model tunggal dengan dataset kecil atau menggunakan citra hasil augmentasi yang belum sepenuhnya merepresentasikan kondisi lapangan. Belum banyak kajian komparatif yang secara langsung mengevaluasi performa berbagai arsitektur CNN dengan data internal maupun eksternal untuk memperoleh model yang paling optimal dan aplikatif dalam konteks pertanian presisi di Indonesia [16].

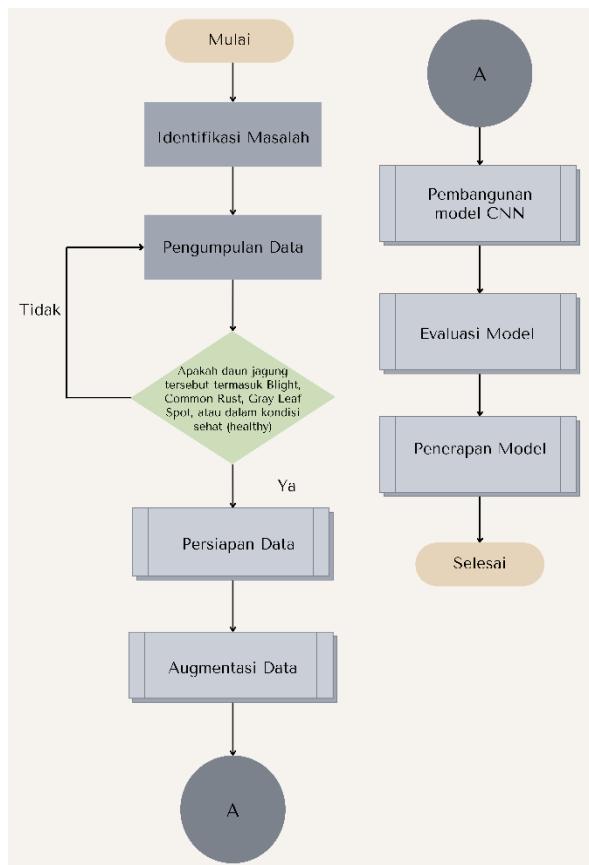
Proses klasifikasi yang dimaksud mencakup identifikasi apakah daun jagung dalam kondisi sehat atau terjangkit penyakit[17]. dari tiga kategori utama, serta mengelompokkan jenis penyakit berdasarkan pola gejala visual yang muncul [18]. Kebaruan penelitian ini terletak pada pengujian berbagai model CNN hingga diperoleh performa terbaik, sekaligus melakukan kombinasi beberapa percobaan untuk menentukan model yang paling tepat dalam mencapai akurasi optimal.

Berdasarkan pertimbangan tersebut, penelitian ini difokuskan pada penerapan dan pengujian beberapa model CNN secara langsung dalam klasifikasi penyakit daun jagung [19]. Dengan memanfaatkan data Kaggle sebagai data internal serta data eksternal untuk validasi [20]. Penelitian ini bertujuan menghasilkan sistem deteksi penyakit berbasis citra digital yang dapat digunakan sebagai alat bantu dalam mendukung praktik pertanian presisi. penelitian ini tidak hanya menawarkan solusi berbasis teknologi terhadap permasalahan penyakit tanaman jagung, tetapi juga menjadi kontribusi nyata dalam penerapan kecerdasan buatan di sektor pertanian. Berdasarkan latar belakang tersebut peneliti membangun model klasifikasi penyakit daun pada tanaman jagung menggunakan CNN sebagai inovasi yang dapat membantu petani dalam mendeteksi penyakit secara lebih cepat dan tepat.



2. Metodologi

Penelitian ini diselesaikan melalui beberapa tahapan yang dirancang secara sistematis. Alur tahapan tersebut menggambarkan proses penelitian mulai dari pengumpulan data hingga evaluasi model. Rincian alur kerja penelitian ditunjukkan pada Gambar 1.



Gambar 1. Alur Kerja Penelitian

A. Identifikasi Masalah

Pada tahap ini, melakukan indentifikasi yang mendalam dan penting terhadap urgensi masalah, struktur kategorisasi, dan keperluan tersebut harus dilakukan. Tujuan dari proses klasifikasi harus jelas dan sudah direncana jauh sebelum penelitian. Penelitian ini menekankan pentingnya pemahaman dalam konteks pengembangan sistem klasifikasi yang relevan, di mana kualitas klasifikasi yang diperoleh berpotensi untuk memberikan manfaat besar bagi petani. Ini termasuk kemampuan petani, jika mengetahui kategori penyakit pada jagung lebih cepat, dapat merekomendasikan pupuk yang terbaik terhadap tanamannya, supaya di kemudian hari pada saat panen semakin bagus hasil jagung yang telah di panen. Penentuan jenis penyakit daun jagung ini bertujuan untuk menciptakan model klasifikasi dari 3 jenis penyakit daun jagung dan 1 jenis daun jagung sehat, yang kemudian akan dikembangkan menjadi sebuah alat manajemen pertanian yang efektif.

B. Pengumpulan Data

Proses pengumpulan data dimulai dengan mencari dataset yang sesuai untuk penelitian ini, dengan fokus pada identifikasi data yang relevan dan berkualitas. Fase ini juga bertujuan untuk mengevaluasi dan mengatasi potensi masalah terkait kualitas data. Data yang digunakan dalam penelitian ini diperoleh dari Kaggle, dengan mengunduh dataset gambar yang terdiri dari format JPG. Dataset ini mencakup gambar daun dengan berbagai kondisi penyakit, yaitu daun blight, daun common rust, daun gray leaf spot, dan daun sehat, dengan total gambar sebanyak 4188. Dataset tersebut dapat diakses melalui tautan berikut: <https://www.kaggle.com/datasets/smaranjitghose/corn-or-maize-leaf-disease-dataset/data>.

C. Persiapan Data

Ini adalah tahap pra-pemrosesan data, yang mencakup perubahan ukuran gambar, menormalkan gambar, membagi rangkaian data, validasi data, dan pengujian data. Untuk membuat data siap untuk diproses pada tahap selanjutnya yaitu tahap peningkatan data, ada beberapa langkah harus dilakukan:

- 1) Data yang telah diproses sebelumnya dibagi menjadi tiga jenis, yaitu pelatihan, validasi, dan tes. Data pelatihan memiliki komposisi yang lebih besar untuk melatih model dengan lebih banyak data, sementara data validasi digunakan untuk mengevaluasi apakah model mengalami overfitting atau underfitting selama pelatihan. Setelah itu, data tes digunakan untuk menguji kinerja model yang telah dilatih. Pembagian data pada setiap percobaan menunjukkan variasi dalam proporsi data pelatihan, validasi, dan testing. Pada percobaan pertama data dibagi menjadi 40% pelatihan, 30% validasi, dan 30% testing. Pada percobaan kedua data dibagi menjadi 80% pelatihan, 10% validasi, dan 10% testing. Pada percobaan ketiga data dibagi menjadi 90% pelatihan, 5% validasi, dan 5% testing. Pada percobaan keempat data dibagi menjadi 80% pelatihan, 10% validasi, dan 10% testing. Pada percobaan kelima data dibagi menjadi 70% pelatihan, 15% validasi, dan 15% testing.
- 2) Percobaan pertama mengubah ukuran foto menjadi 128 x 128, dan percobaan kedua mengubahnya menjadi 224 x 224. Ini dilakukan karena data gambar dalam dataset data_coba memiliki orientasi, sudut pandang, dan ukuran yang terlalu besar atau kecil. Oleh karena itu, untuk mempercepat proses pelatihan, perlu mengubah ukuran dataset yang ada agar data memiliki ukuran yang sama sehingga bisa mempercepat proses melatih data.

D. Melakukan Augmentasi Data

Mesin pembelajaran umumnya menunjukkan kinerja yang lebih baik apabila dilatih dengan data yang lebih banyak dan bervariasi. Jumlah data pelatihan yang sedikit dapat menimbulkan masalah seperti *overfitting*, sedangkan penambahan variasi data dapat meningkatkan generalisasi model. Salah satu pendekatan untuk meningkatkan variasi data adalah *data augmentation*, yaitu memodifikasi citra pelatihan melalui transformasi seperti rotasi, zoom,



horizontal flip, serta pergeseran lebar dan tinggi. Meskipun jumlah citra asli tidak bertambah secara permanen, variasi data yang dipelajari model meningkat dan dapat berdampak positif pada hasil pelatihan.

Tabel 1 Pembagian Dataset Setelah Augmetasi

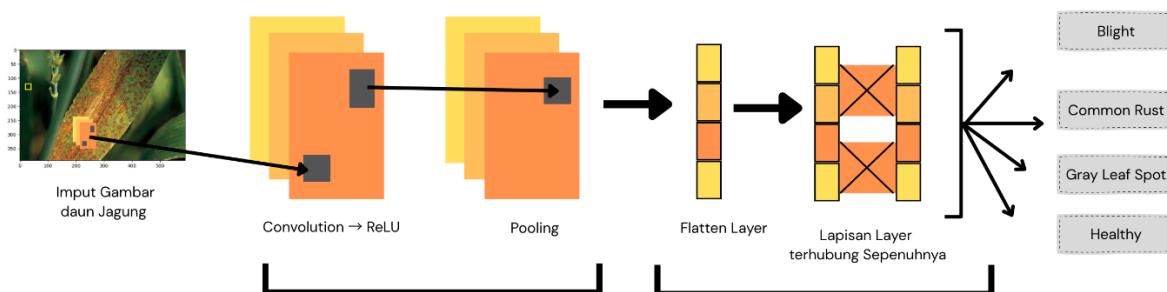
No	Sampel Data	Train setelah augmentasi	Validasi	testing	Total Data Digunakan
1	Percobaan 1 CNN	1675	1257	1256	4188
2	Percobaan 2 CNN	3350	419	419	4188
3	Percobaan 3 CNN	3769	210	209	4188
4	Percobaan 4 VGG 16	3350	419	419	4188
5	Percobaan 5 Xception	3735	801	800	5336

Pada percobaan 1 hingga 4, augmentasi data yang digunakan bersifat *on-the-fly*, tidak menambah jumlah dataset secara permanen. Pendekatan ini membuat jumlah data pada subset pelatihan, validasi, dan pengujian tetap sama seperti yang ditunjukkan pada tabel total data digunakan. Model hanya menerima variasi baru dari citra pada setiap epoch, sementara sumber data citra yang digunakan tetap identik. Percobaan ke-5 dengan model Xception diterapkan teknik class balancing melalui penduplikasian citra pada kelas *Gray Leaf Spot* sebanyak dua kali, sehingga distribusi kelas menjadi lebih seimbang. Proses ini menyebabkan jumlah dataset bertambah 5336 citra. Teknik tersebut sejalan dengan konsep Random Oversampling (RandOS), yaitu metode penyeimbangan kelas dengan cara menggandakan data minoritas.

Abdulsadig dan Rodriguez-Villegas menjelaskan kelas minoritas ditingkatkan jumlahnya dengan cara memilih secara acak dan menduplikasi data dari kelas tersebut [21].

E. Pembangunan Model CNN

Gambar daun jagung digunakan untuk memodelkan sistem deteksi penyakit pada tanaman jagung. CNN biasanya terdiri dari beberapa lapisan, seperti lapisan konvolusional, lapisan pooling, lapisan dropout, lapisan flatten layer, dan lapisan yang terhubung sepenuhnya. Gambar 2 di bawah ini menunjukkan gambaran umum tentang pemodelan menggunakan algoritma CNN.



Gambar 2. Pemodelan Algoritma CNN Daun Jagung

1. Lapisan Konvolusi

Output neuron yang terhubung ke area lokal input, titik antara bobotnya, dan area kecil yang terhubung ke volume input masing-masing akan dihitung oleh lapisan ini. Beberapa hiperparameter, seperti ukuran filter, langkah, dan bantalan, dapat disesuaikan di setiap lapisan. Parameter yang menentukan seberapa banyak filter bergerak adalah stride. Kemungkinan input akan mendapatkan lebih banyak informasi dari langkah yang lebih kecil, tetapi langkah yang lebih besar membutuhkan lebih banyak perhitungan. Padding merupakan lapisan tambahan yang dapat ditambahkan ke batas gambar dengan meningkatkan ukuran piksel di sekitar data input sehingga reseptif tidak terlalu kecil dan informasi tidak hilang. Zero padding terjadi ketika nilai ini biasanya nol. Output yang dihasilkan oleh proses konvolusi ini akan digunakan sebagai input untuk lapisan konvolusi

berikutnya. Dengan memperkenalkan non-linearity serta dapat membangun representasi dan model. Dengan menggunakan fungsi $f(x) = \max(0, x)$, fungsi aktivasi ReLU akan menghilangkan gradien yang hilang. Selain itu, aktivasi elemen akan terjadi ketika elemen berada di ambang batas 0.

2. Lapisan pooling

Dengan mengurangi dimensi peta fitur, lapisan pooling dapat mempercepat perhitungan karena lebih sedikit parameter yang perlu diperbarui. Lapisan ini juga dapat mengatasi overfitting dan mengambil nilai maksimum di area filter.

3. Dropout Layer

Fungsing Dropout yaitu mengurangi kompleksitas model yang telah dibangun dengan membangun jaringan



saraf yang tidak terpakai. Ukuran model akan tetap sama, tidak ada perubahan.

4. Flatten Layer

Bentuk ulang peta fitur menjadi vektor untuk digunakan sebagai input dari lapisan layer yang terhubung sepenuhnya. Lapisan Terhubung sepenuhnya.

5. Lapisan Layer terhubung Sepenuhnya

Lapisan ini terhubung sepenuhnya. Peta fitur yang dihasilkan harus diratakan agar dapat digunakan sebagai input dari lapisan yang terhubung sepenuhnya, karena masih berupa matriks filter. Setiap neuron berada di lapisan ini, dan mereka terhubung ke semua angka dan volume.

F. Evaluasi Model

Hasil pemodelan dengan optimasi terbaik digunakan untuk mengklasifikasikan data pengujian, yang mencakup citra daun jagung yang mengalami penyakit blight, common rust, gray leaf spot, serta daun yang sehat (healthy). Akurasi yang diperoleh dari penerapan metode klasifikasi dengan algoritma Convolutional Neural Network (CNN) dapat dilihat melalui confusion matrix, yang menunjukkan hasil klasifikasi ke dalam empat jenis kelas daun jagung tersebut.

Confusion matrix merupakan matriks yang menunjukkan hasil klasifikasi dalam suatu himpunan data. Matriks ini biasanya disajikan dalam bentuk tabel dan sering digunakan untuk mengevaluasi kinerja model klasifikasi. Tabel ini terdiri dari baris dan kolom sebanyak jumlah kelas, serta menampilkan nilai False Positive (FP), yaitu data negatif yang terdeteksi sebagai positif; False Negative (FN), yaitu data positif yang terdeteksi sebagai negatif; True Positive (TP), yaitu data positif yang terdeteksi dengan benar; dan True Negative (TN), yaitu data negatif yang terdeteksi dengan benar. Persamaan (1)

3. Hasil dan Pembahasan

Pada bagian hasil dan pembahasan, penelitian ini menguji sistem deteksi penyakit jagung pada daun jagung dengan menerapkan metode klasifikasi berbasis algoritma CNN, yang dilakukan melalui serangkaian tahapan sebagai berikut.

A. Data pemrosesan Awal (Processing Data)

Pada tahap preprocessing, seluruh data citra yang telah dikumpulkan akan diproses lebih lanjut. Dataset ini terdiri dari 576.452 citra daun jagung yang terinfeksi penyakit dan 1.162 citra daun sehat, sehingga total data mencapai 577.614 citra. Selanjutnya, data tersebut dibagi ke dalam tiga subset, yaitu data pelatihan, validasi, dan pengujian. Proses pelatihan dan validasi model dilakukan selama fase training, sedangkan subset pengujian digunakan untuk evaluasi akhir. Pembagian data untuk setiap eksperimen disajikan secara rinci pada Tabel I di bawah ini.

menyajikan rumus perhitungan akurasi berdasarkan confusion matrix tersebut.

$$Accuracy = \frac{TP + TNP}{TP + TN + FP + FN} \times 100\% \quad (1)$$

Kemudian gunakan persamaan berikut untuk mengukur nilai presisi. Kemampuan pengklasifikasi untuk tidak memberi label positif pada sampel negatif disebut presisi intuitif. Persamaan untuk perhitungan presisi Confusion matrix dapat ditemukan dalam persamaan 2.

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

langkah berikutnya gunakan Persamaan (3) untuk menghitung nilai recall. Recall secara intuitif menunjukkan kemampuan model klasifikasi untuk menemukan seluruh sampel yang termasuk dalam kelas positif. Perhitungan recall didasarkan pada nilai-nilai pada matriks confusion.

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

Setelah itu untuk mengukur nilai F1-Score gunakan persamaan 4 berikut ini untuk menentukan hasilnya.

$$F1-Score = 2 \times \frac{precision \times recall}{precision + recall} \quad (4)$$

G. Penerapan Model

Pada penelitian ini, penerapan model di lakukan setelah melakukan pelatihan pada data dan telah mendapatkan convolutional matrix, jika hasil data belum memungkinkan maka akan di lakukan pengujian baru sampai menemukan akurasi terbaik. Untuk di uji menggunakan data external. Dengan cara memanggil model dan di import ke dalam jupyter notebook.

Tabel 2 Pembagian Dataset

No	Sampel Data	Traning	Validasi	testing
1	Percobaan 1 CNN	40 %	30%	30%
2	Percobaan 2 CNN	80%	10%	10%
3	Percobaan 3 CNN	90%	5%	5%
4	Percobaan 4 VGG 16	80%	10%	10%
5	Percobaan 5 Xception	70%	15%	15%

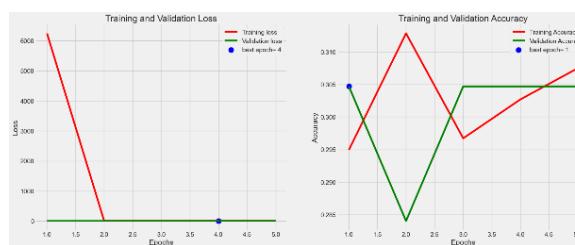
Berdasarkan Tabel X, masing-masing percobaan menerapkan strategi partisi data yang bervariasi dalam membagi dataset ke dalam tiga subset utama, yaitu data pelatihan, validasi, dan pengujian. Pada Percobaan 1 dengan model CNN, proporsi distribusi data dilakukan secara relatif seimbang, yakni 40% untuk pelatihan, serta



masing-masing 30% untuk validasi dan pengujian. Meskipun demikian, proporsi data pelatihan tergolong rendah dibandingkan percobaan lainnya. Adapun pada Percobaan 2 dan 4, alokasi data lebih difokuskan pada pelatihan sebesar 80%, sedangkan validasi dan pengujian masing-masing hanya memperoleh 10%, dengan tujuan meningkatkan efektivitas pembelajaran model. Sebaliknya, Percobaan 3 memberikan porsi data latih yang lebih dominan, yaitu 90%, sementara validasi dan pengujian hanya menyisakan 5% masing-masing, yang berpotensi menimbulkan risiko overfitting akibat minimnya data evaluasi. Di sisi lain, pada Percobaan 5 yang menggunakan arsitektur Xception, pembagian data dilakukan secara lebih proporsional dengan komposisi 70% untuk pelatihan dan masing-masing 15% untuk validasi serta pengujian, sehingga dinilai lebih ideal dalam mendukung kebutuhan pelatihan model deep learning yang memerlukan jumlah data pelatihan yang besar, namun tetap mempertahankan validitas evaluasi model.

B. Hasil Pemodelan Convolutional Neural Network (CNN)

1. Percobaan 1 CNN



Gambar 3. Kurva loss dan akurasi CNN percobaan 1

Percobaan ini dilakukan dengan menggunakan parameter batch size sebesar 20 dan ukuran citra masukan (input image size) sebesar $128 \times 128 \times 3$ piksel, yang mewakili format citra berwarna dengan tiga kanal RGB. Arsitektur jaringan yang digunakan ditunjukkan pada Gambar 3, di mana model *Convolutional Neural Network* (CNN) dirancang dengan empat lapisan konvolusi (convolutional layers) serta satu lapisan fully connected sebagai tahap akhir pemrosesan fitur. Pada setiap lapisan konvolusi diterapkan kombinasi jumlah filter yang bervariasi, yaitu 64, 4, 9, dan 81 filter, yang berfungsi untuk mengekstraksi representasi pola dari citra masukan dengan tingkat kompleksitas yang berbeda-beda. Proses pelatihan model dilaksanakan menggunakan algoritma optimasi RMSprop dengan nilai learning rate yang relatif tinggi, yaitu 0,1, dan dijalankan selama lima epoch.

Hasil grafik training loss dan validation loss terlihat nilai loss pada data pelatihan menurun secara signifikan dari epoch pertama hingga akhir, yang menunjukkan kemampuan model untuk menyesuaikan bobot parameter terhadap data pelatihan. nilai validation loss justru cenderung stagnan dan tidak mengalami penurunan sehingga dapat diinterpretasikan bahwa model mengalami gejala overfitting. Ini diperkuat melalui analisis grafik training accuracy dan validation accuracy. Grafik menunjukkan akurasi pada data pelatihan meningkat tajam seiring dengan bertambahnya epoch, sedangkan

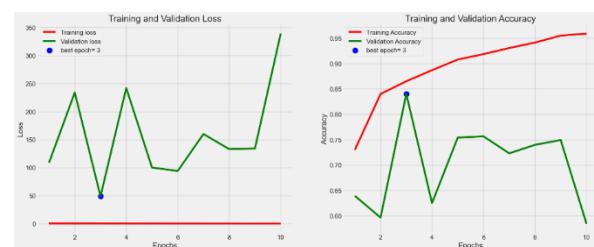
akurasi pada data validasi tidak mengalami peningkatan signifikan dan relatif stabil pada nilai yang rendah. Mengindikasikan model mampu mempelajari pola pada data pelatihan dengan baik, namun model gagal melakukan generalisasi terhadap data baru, sehingga performanya tidak optimal pada data validasi



Gambar 4. Confusion Matrix Model CNN percobaan 1

Hasil evaluasi menggunakan *confusion matrix* pada Gambar 4 menunjukkan bahwa model mengalami kesulitan dalam menggeneralisasi data validasi. Kondisi tersebut menunjukkan akurasi keseluruhan yang relatif rendah sebesar 30,25%. Temuan ini mengindikasikan adanya *underfitting*, di mana model belum mampu mengenali pola secara optimal dari data uji yang diberikan.

2. Percobaan 2 CNN



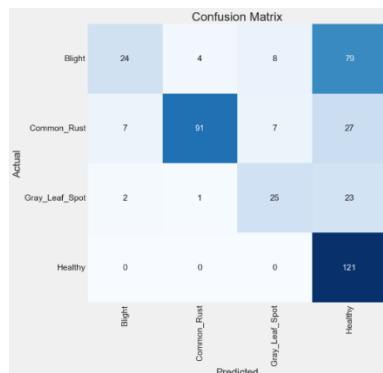
Gambar 5. Kurva loss dan akurasi CNN percobaan 2

Percobaan yang ditampilkan pada Gambar 4 menggunakan batch size sebesar 32 dengan ukuran citra masukan (input image size) $128 \times 128 \times 3$ piksel. Arsitektur jaringan yang digunakan terdiri dari 2 lapisan konvolusi, 2 lapisan pooling, dan diakhiri dengan 1 lapisan fully connected. Pada setiap lapisan konvolusi, digunakan dua jenis jumlah filter, yaitu 32 dan 64. Selain itu, padding diterapkan agar dimensi citra tidak berubah selama proses pelatihan. Penggunaan stride juga diterapkan untuk

memperhalus proses ekstraksi fitur sehingga dapat mempelajari informasi lebih rinci dari setiap piksel gambar. Dalam proses kompilasi model, digunakan algoritma optimasi Adam dengan learning rate sebesar 0,001. Pemilihan parameter ini bertujuan untuk menyesuaikan pembaruan bobot secara adaptif selama proses pelatihan. Proses pelatihan dijalankan selama 10 epoch. Berdasarkan grafik training and validation loss, terlihat fluktuasi yang cukup tinggi, terutama pada



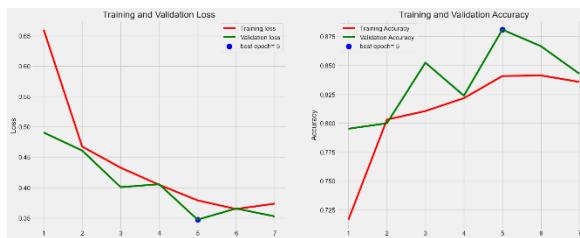
validation loss, yang menunjukkan bahwa model belum sepenuhnya stabil dalam mempelajari pola dari data validasi. Sementara itu, grafik training and validation accuracy memperlihatkan bahwa akurasi data pelatihan meningkat secara konsisten, tetapi akurasi validasi cenderung fluktuatif, yang menandakan potensi overfitting ringan.



Gambar 6. Confusion Matrix Model CNN percobaan 2

Evaluasi menggunakan confusion matrix menunjukkan bahwa model memiliki kinerja yang cukup baik dalam mengklasifikasikan kelas Healthy dan Common Rust, namun masih terjadi kesalahan klasifikasi pada kelas Blight dan Gray Leaf Spot. Hal ini menunjukkan bahwa model masih perlu ditingkatkan dalam hal generalisasi terhadap beberapa kelas tertentu. Secara keseluruhan, model pada percobaan ini berhasil mencapai akurasi klasifikasi sebesar 62,29%, yang menunjukkan adanya perbaikan dibandingkan percobaan sebelumnya, meskipun masih memerlukan penyempurnaan lebih lanjut untuk meningkatkan performa klasifikasi lintas kelas secara merata.

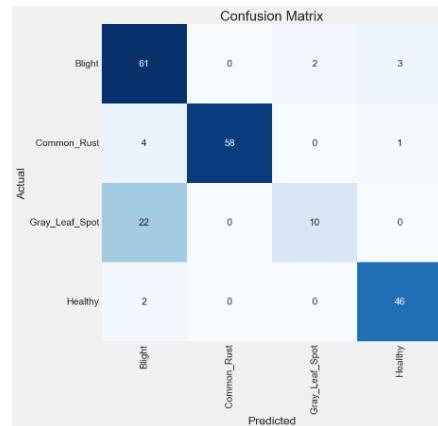
3. Percobaan 3 CNN



Gambar 7. Kurva loss dan akurasi CNN percobaan 3

Gambar 5 menunjukkan arsitektur model pada tahap uji ke-3, yang terdiri dari satu lapisan konvolusi, satu lapisan pooling, dan lapisan dropout untuk mengurangi risiko overfitting. Model diakhiri dengan lapisan fully connected yang terhubung ke output. Model dilatih menggunakan optimizer Adam dengan learning rate sebesar 0,0001, yang membantu proses pelatihan berlangsung stabil dan adaptif. Selain itu, digunakan dua callback penting, yaitu Early Stopping untuk menghentikan pelatihan saat performa validasi tidak lagi meningkat, serta Model Checkpoint untuk menyimpan bobot model terbaik. Pelatihan dilakukan dengan batch size 32, ukuran citra 224×224×3 piksel, dan maksimal 50

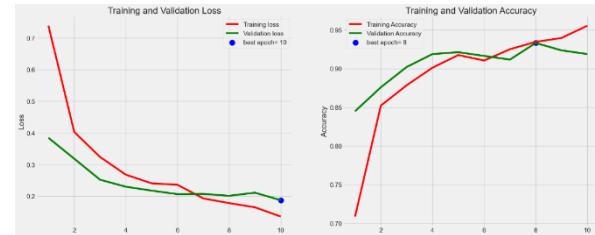
epoch. Namun, pelatihan berhenti otomatis pada epoch ke-12 karena Early Stopping aktif. Hasil pelatihan menunjukkan grafik loss yang menurun stabil dan grafik akurasi yang meningkat pada data pelatihan maupun validasi.



Gambar 8. Confusion Matrix Model CNN percobaan 3

Pada tahap percobaan ke-3, model berhasil mencapai akurasi validasi sebesar 83,73%. Nilai ini menunjukkan bahwa performa klasifikasi yang dicapai tergolong cukup baik. Dengan demikian, model mampu mengenali pola data secara lebih optimal dibandingkan percobaan sebelumnya.

4. Percobaan 4 menggunakan VGG16



Gambar 9. Kurva loss dan akurasi VGG16

Percobaan ke-4 dilakukan dengan memanfaatkan arsitektur VGG16 untuk mengevaluasi performa model dalam mengenali citra berukuran 224×224×3 piksel, dengan menggunakan batch size sebesar 64. Gambar 9 memperlihatkan struktur model yang terdiri dari satu lapisan konvolusi, satu lapisan dropout, dan satu lapisan fully connected. Hiperparameter utama yang digunakan dalam percobaan ini adalah satu jenis filter berukuran 32, dengan penambahan padding untuk mempertahankan dimensi citra input dan penggunaan strides guna memungkinkan pembelajaran fitur secara lebih rinci pada tiap piksel. Lapisan dropout disisipkan untuk menonaktifkan sejumlah neuron secara acak selama pelatihan, dengan tujuan mengurangi overfitting dan meningkatkan kemampuan generalisasi model. Proses kompilasi model dilakukan menggunakan optimizer Adam dengan learning rate sebesar 0,001, untuk memastikan proses pembaruan bobot berlangsung secara adaptif dan stabil. Pelatihan dilakukan selama 15 epoch, dan hasilnya divisualisasikan melalui grafik loss dan akurasi. Grafik training and validation loss menunjukkan



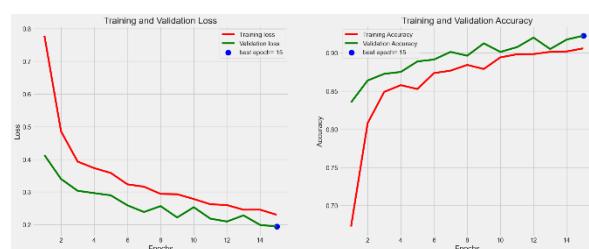
tren penurunan yang konsisten pada data pelatihan (garis merah) maupun validasi (garis biru), mengindikasikan bahwa model belajar secara stabil tanpa overfitting. Sementara itu, grafik training and validation accuracy menunjukkan peningkatan akurasi secara bertahap pada kedua dataset, yang menandakan bahwa model mampu mempelajari fitur penting dari citra masukan serta melakukan generalisasi dengan baik terhadap data uji. Model pada percobaan ini berhasil mencapai akurasi validasi sebesar 91,41%, yang merupakan capaian terbaik dari seluruh rangkaian percobaan.



Gambar 10. Confusion Matrix Model VGG16

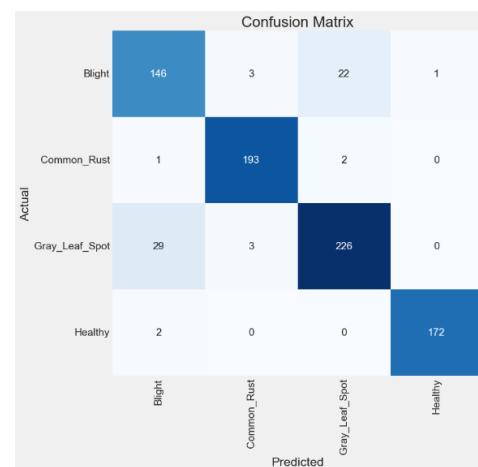
Capaian akurasi tersebut diperkuat oleh hasil *confusion matrix* yang memperlihatkan kinerja klasifikasi yang tinggi pada setiap kelas. Secara khusus, kelas *Healthy* dan *Blight* menunjukkan jumlah prediksi benar yang paling dominan dibandingkan kelas lainnya. Hal ini menegaskan bahwa model memiliki kemampuan lebih baik dalam mengenali pola visual pada kedua kelas tersebut.

5. Percobaan ke 5 menggunakan Xception



Gambar 11. Kurva loss dan akurasi Xception

Pada percobaan ke-5, model dikembangkan menggunakan arsitektur Xception dengan ukuran citra 224×224 piksel dan tiga saluran warna (RGB). Dataset disusun dari direktori berdasarkan path dan label kelas, lalu diseimbangkan melalui oversampling pada kelas *Gray_Leaf_Spot*. Data dibagi menjadi subset training, validasi, dan pengujian menggunakan metode stratified split. Selama pelatihan, augmentasi citra dilakukan untuk meningkatkan variasi data, sedangkan data validasi dan uji hanya dilakukan normalisasi. Model terdiri atas lapisan Conv2D, MaxPooling, Dropout, Flatten, dan Dense (softmax), serta dikompilasi menggunakan optimizer Adam dengan learning rate 0,0001. Hasil pelatihan selama 15 epoch menunjukkan tren peningkatan akurasi dan penurunan loss yang stabil (Gambar 1), dengan akurasi validasi tertinggi sebesar 92,12%.



Gambar 12. Confusion Matrix Model Xception

Evaluasi menggunakan *confusion matrix* pada Gambar 12 menunjukkan bahwa model memiliki kinerja klasifikasi yang baik dan seimbang. Hal ini tercermin dari capaian akurasi keseluruhan sebesar 92%. Selain itu, nilai *precision*, *recall*, dan *f1-score* rata-rata yang mencapai 0,92 semakin menegaskan konsistensi performa model pada seluruh kelas.

C. Evaluasi Model

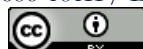
Pengujian model dilakukan dengan menampilkan *confusion matrix* sebagai alat evaluasi kinerja klasifikasi. Melalui *confusion matrix*, distribusi prediksi model terhadap setiap kelas dapat diamati secara menyeluruh. Rangkuman hasil percobaan ini ditunjukkan pada tabel berikut.

Tabel 3 Evaluasi Convusion Matrix

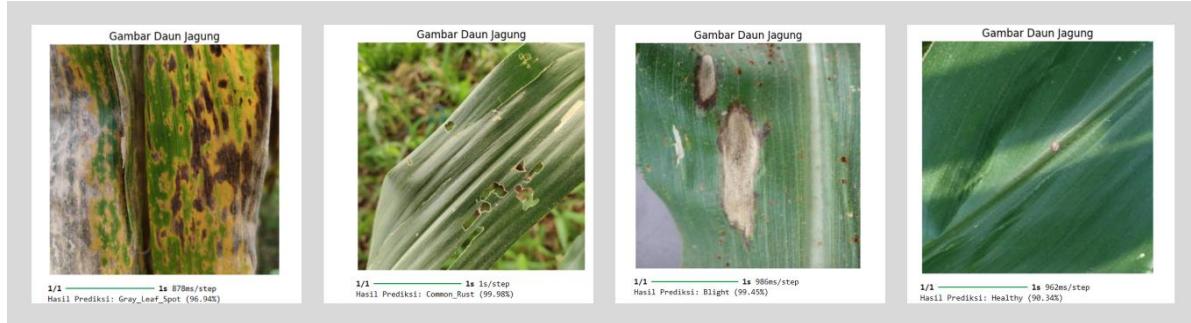
No	Percobaan	akurasi	precision	recall	F1_score	result
1	Percobaan 1 CNN	30.25 %	0.09	0.30	0.14	<i>underfitting</i>
2	Percobaan 2 CNN	62.29 %	0.71	0.62	0.60	<i>underfitting</i>
3	Percobaan 3 CNN	83.73 %	0.86	0.78	0.82	<i>Good</i>
4	Percobaan 4 VGG16	91.41 %	0.92	0.91	0.91	<i>Very good</i>
5	Percobaan 5 Xception	92.12 %	0.92	0.92	0.92	<i>Very good</i>

Perhitungan pada tabel di atas dilakukan berdasarkan metrik evaluasi klasifikasi, di mana akurasi dihitung

berdasarkan persamaan 1, presisi pada persamaan 2, recall pada persamaan 3, dan F1-score mengacu pada



persamaan 4. Dari kelima percobaan, terlihat bahwa performa model meningkat secara bertahap dari percobaan pertama hingga kelima. Akurasi terendah sebesar 30.25% terdapat pada percobaan pertama, yang menunjukkan gejala underfitting. Percobaan kedua juga menunjukkan underfitting, meskipun nilai metriknya sedikit lebih tinggi. Hasil yang baik mulai tampak pada percobaan ketiga dengan akurasi mencapai 83.73% dan F1-score sebesar 0.82, yang mengindikasikan bahwa model mulai mampu mengenali pola data dengan baik.



Gambar 13. Hasil Penerapan Model

Model diimplementasikan dengan mengimpor citra daun jagung ke dalam lingkungan Jupyter Notebook untuk menguji kemampuan klasifikasi ke dalam empat kategori. Data uji berasal dari empat citra eksternal daun jagung yang diambil langsung melalui pemotretan di area perkebunan. Tujuannya adalah untuk mengevaluasi apakah model mampu mengidentifikasi kategori penyakit dengan akurat. Gambar berikut menyajikan hasil klasifikasi dari masing-masing citra daun jagung tersebut.

4. Kesimpulan

Berdasarkan hasil yang ditampilkan pada Tabel 2, dapat disimpulkan bahwa performa model klasifikasi mengalami peningkatan signifikan dari percobaan pertama hingga percobaan kelima. Percobaan pertama dan kedua menunjukkan hasil akurasi rendah, masing-masing sebesar 30.25% dan 62.29%, dengan nilai precision, recall, dan F1-score yang juga relatif rendah. Hal ini menunjukkan bahwa model masih mengalami underfitting, di mana model belum mampu mengenali pola data dengan baik. Peningkatan mulai terjadi pada percobaan ketiga, dengan akurasi mencapai 83.73% serta F1-score sebesar 0.82. Hal ini mengindikasikan bahwa arsitektur dan konfigurasi model pada percobaan ketiga

sementara itu, percobaan keempat dan kelima menunjukkan performa terbaik, dengan akurasi di atas 91% serta nilai precision, recall, dan F1-score yang seimbang dan tinggi. Kedua percobaan ini dikategorikan sebagai very good, yang menunjukkan bahwa arsitektur model dan parameter yang digunakan telah optimal dalam mendeteksi jenis hama pada tanaman caisim.

D. Hasil Penerapan Model

sudah cukup baik dalam melakukan klasifikasi terhadap beberapa kelas. Hasil terbaik diperoleh pada percobaan keempat dan kelima, masing-masing dengan akurasi sebesar 91.41% dan 92.12%. Kedua percobaan ini memiliki nilai precision, recall, dan F1-score yang seimbang dan tinggi, yaitu mencapai angka 0.91 dan 0.92. Dengan demikian, model pada percobaan keempat dan kelima dikategorikan sebagai very good, serta memiliki kemampuan klasifikasi yang stabil dan andal. Keberhasilan tersebut dipengaruhi oleh beberapa faktor, antara lain komposisi data latih yang cukup besar, pemilihan arsitektur yang sesuai (seperti penggunaan model VGG16 dan Xception), serta penerapan strategi training seperti callback (early stopping dan model checkpoint), pengaturan hyperparameter yang tepat, dan penggunaan optimizer Adam dengan learning rate rendah. Dengan demikian, dapat disimpulkan bahwa pengembangan model CNN yang baik sangat bergantung pada pengolahan data, desain arsitektur yang tepat, serta penyesuaian hyperparameter yang sesuai. Untuk pengembangan lebih lanjut, disarankan untuk mencoba arsitektur lain atau memperluas dataset guna menghindari risiko overfitting dan meningkatkan generalisasi model.

5. Daftar Pustaka

- [1] X. Sun and H. Huo, "Corn leaf disease recognition based on improved EfficientNet," *IET Image Process.*, vol. 19, no. 1, pp. 1–12, 2025, doi: 10.1049/ipt2.13288
- [2] M. Tariq *et al.*, "Corn leaf disease: insightful diagnosis using VGG16 empowered by explainable AI," *Front. Plant Sci.*, vol. 15, no. June, pp. 1–12, 2024, doi: 10.3389/fpls.2024.1402835
- [3] X. Zhang, Y. Qiao, F. Meng, C. Fan, and M. Zhang, "Identification of maize leaf diseases using improved deep convolutional neural networks," *IEEE Access*, vol. 6, pp. 30370–30377, 2018, doi: 10.1109/ACCESS.2018.2844405
- [4] N. Ermias, B. Mesfin, D. Chaka, A. Gezahegn, S. Bikila, and W. Teshome, *Transforming agriculture with Machine Learning, Deep Learning, and IoT: perspectives from Ethiopia — challenges and opportunities*. Springer International Publishing, 2024. doi:



- 10.1007/s44279-024-00066-7
- [5] F. Rajeena P. P, A. S. U, M. A. Moustafa, and M. A. S. Ali, "Detecting Plant Disease in Corn Leaf Using EfficientNet Architecture—An Analytical Approach," *Electron.*, vol. 12, no. 8, 2023, doi: 10.3390/electronics12081938
- [6] R. Ahila Priyadharshini, S. Arivazhagan, M. Arun, and A. Mirnalini, "Maize leaf disease classification using deep convolutional neural networks," *Neural Comput. Appl.*, vol. 31, no. 12, pp. 8887–8895, 2019, doi: 10.1007/s00521-019-04228-3
- [7] J. Kusuma, Rubianto, R. Rosnelly, Hartono, and B. H. Hayadi, "Klasifikasi Penyakit Daun Pada Tanaman Jagung Menggunakan Algoritma Support Vector Machine, K-Nearest Neighbors dan Multilayer Perceptron," *J. Appl. Comput. Sci. Technol.*, vol. 4, no. 1, pp. 1–6, 2023, doi: 10.52158/jacost.v4i1.484
- [8] A. Pfordt and S. Paulus, "A review on detection and differentiation of maize diseases and pests by imaging sensors," *J. Plant Dis. Prot.*, vol. 132, no. 1, pp. 1–21, 2025, doi: 10.1007/s41348-024-01019-4
- [9] G. Mohyuddin, M. A. Khan, A. Haseeb, S. Mahpara, M. Waseem, and A. M. Saleh, "Evaluation of Machine Learning Approaches for Precision Farming in Smart Agriculture System: A Comprehensive Review," *IEEE Access*, vol. 12, no. March, pp. 60155–60184, 2024, doi: 10.1109/ACCESS.2024.3390581
- [10] H. Zhou *et al.*, "Maize Leaf Disease Recognition Based on Improved Convolutional Neural Network ShuffleNetV2," *Plants*, vol. 13, no. 12, pp. 1–18, 2024, doi: 10.3390/plants13121621
- [11] R. Archana and P. S. E. Jeevaraj, *Deep learning models for digital image processing: a review*, vol. 57, no. 1. Springer Netherlands, 2024. doi: 10.1007/s10462-023-10631-z
- [12] S. Timilsina *et al.*, "CNDD-Net: A Lightweight Attention-Based Convolutional Neural Network for Classifying Corn Nutritional Deficiencies and Leaf Diseases," *Electron.*, vol. 14, no. 7, pp. 1–21, 2025, doi: 10.3390/electronics14071482
- [13] C. Yin, T. Zeng, H. Zhang, W. Fu, L. Wang, and S. Yao, "Maize Small Leaf Spot Classification Based on Improved Deep Convolutional Neural Networks with a Multi-Scale Attention Mechanism," *Agronomy*, vol. 12, no. 4, 2022, doi: 10.3390/agronomy12040906
- [14] S. A. Shah *et al.*, "Application of Drone Surveillance for Advance Agriculture Monitoring by Android Application Using Convolution Neural Network," *Agronomy*, vol. 13, no. 7, 2023, doi: 10.3390/agronomy13071764
- [15] R. Rashid, W. Aslam, R. Aziz, and G. Aldehim, "An Early and Smart Detection of Corn Plant Leaf Diseases Using IoT and Deep Learning Multi-Models," *IEEE Access*, vol. 12, no. February, pp. 23149–23162, 2024, doi: 10.1109/ACCESS.2024.3357099
- [16] W. Shafik, A. Tufail, C. Liyanage De Silva, and R. A. Awg Haji Mohd Apong, "A novel hybrid inception-xception convolutional neural network for efficient plant disease classification and detection," *Sci. Rep.*, vol. 15, no. 1, p. 3936, 2025, doi: 10.1038/s41598-024-82857-y
- [17] T. A. Prasetyo, V. L. Desrony, H. F. Panjaitan, R. Sianipar, and Y. Pratama, "Corn plant disease classification based on leaf using residual networks-9 architecture," *Int. J. Electr. Comput. Eng.*, vol. 13, no. 3, pp. 2908–2920, 2023, doi: 10.11591/ijece.v13i3.pp2908-2920
- [18] A. Ahmad, D. Saraswat, A. El Gamal, and G. Johal, "CD&S Dataset: Handheld Imagery Dataset Acquired Under Field Conditions for Corn Disease Identification and Severity Estimation," 2021, [Online]. Available: <http://arxiv.org/abs/2110.12084>
- [19] R. Poojary, R. Raina, and A. K. Mondal, "Effect of data-augmentation on fine-tuned cnn model performance," *LAES Int. J. Artif. Intell.*, vol. 10, no. 1, pp. 84–92, 2021, doi: 10.11591/ijai.v10.i1.pp84-92
- [20] M. M. Malik *et al.*, "A novel deep CNN model with entropy coded sine cosine for corn disease classification," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 36, no. 7, p. 102126, 2024, doi: 10.1016/j.jksuci.2024.102126
- [21] R. S. Abdulsadig and E. Rodriguez-Villegas, "A comparative study in class imbalance mitigation when working with physiological signals," *Front. Digit. Heal.*, vol. 6, no. March, pp. 1–11, 2024, doi: 10.3389/fdgth.2024.1377165

