https://doi.org/10.37817/ikraith-informatika.v10i2
P-ISSN : 2580-4316
E-ISSN : 2654-8054

TRADE-OFF PERFORMA DAN EFISIENSI KOMPUTASI TIGA ARSITEKTUR DEEP LEARNING UNTUK KLASIFIKASI PENYAKIT DAUN TEBU SECARA OTOMATIS

Kevin Januari[*1], Nizirwan Anwar[2], Diah Aryani[3], Azifa Habiba[4]

[1,2,3,4] Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Esa Unggul, Indonesia

Email: kevinjanuwary@student.esaunggul.ac.id, nizirwan.anwar@esaunggul.ac.id, diah.aryani@esaunggul.ac.id, azifa.habiba@esaunggul.ac.id

Abstrak

Tanaman tebu merupakan komoditas perkebunan strategis yang produktivitasnya kerap terancam oleh serangan penyakit daun. Identifikasi penyakit secara visual manual masih banyak digunakan di lapangan, namun pendekatan ini memakan waktu lama, bersifat subjektif, dan rentan terhadap kesalahan. *Deep learning* menawarkan solusi otomatisasi klasifikasi penyakit tanaman, tetapi studi yang membandingkan secara komprehensif *trade-off* antara akurasi dan efisiensi komputasi antararsitektur masih terbatas jumlahnya. Penelitian ini bertujuan menganalisis secara komparatif kinerja DenseNet-121, ConvNeXt-Tiny, dan YOLOv11s-CLS dalam mengklasifikasikan delapan kelas kondisi daun tanaman tebu. Metode penelitian menggunakan dataset gabungan sebanyak 4.075 citra yang dibagi secara *stratified* dengan rasio 70:15:15, serta pendekatan *transfer learning* berbasis ImageNet pada ketiga arsitektur. Evaluasi dilakukan pada tiga dimensi, yaitu kinerja klasifikasi, kemampuan generalisasi, dan efisiensi komputasi. Hasil pengujian menunjukkan DenseNet-121 memperoleh *accuracy* tertinggi sebesar 93,14% dan *F1-score macro* 91,09%, ConvNeXt-Tiny unggul pada *precision macro* sebesar 92,59%, sedangkan YOLOv11s-CLS mencatatkan *accuracy* 92,81% dengan jumlah parameter, ukuran model, *FLOPs*, dan waktu inferensi terendah yaitu 4,70 milidetik serta *generalization gap* terkecil sebesar 0,0027. Penelitian ini menyimpulkan bahwa DenseNet-121 sesuai untuk skenario yang memprioritaskan akurasi, sedangkan YOLOv11s-CLS direkomendasikan sebagai model dengan keseimbangan performa dan efisiensi terbaik untuk implementasi pada perangkat berdaya komputasi terbatas.

Kata kunci: convnext-tiny, *deep learning*, densenet-121, klasifikasi citra, penyakit daun tebu, yolov11s-cls

Abstract

Sugarcane is a strategic plantation commodity whose productivity is frequently threatened by leaf disease infections. Manual visual identification is still widely practiced in the field, yet this approach is time-consuming, subjective, and prone to human error. Deep learning offers a solution for automating plant disease classification, but comprehensive studies comparing the accuracy-efficiency trade-off across architectures remain limited. This study aims to comparatively analyze the performance of DenseNet-121, ConvNeXt-Tiny, and YOLOv11s-CLS in classifying eight classes of sugarcane leaf conditions. The method employed a combined dataset of 4,075 images divided using a stratified split with a 70:15:15 ratio, along with an ImageNet-based transfer learning approach applied to all three architectures. Evaluation covered three dimensions: classification performance, generalization ability, and computational efficiency. The results show that DenseNet-121 achieved the highest accuracy of 93.14% and macro F1-score of 91.09%, ConvNeXt-Tiny obtained the highest macro precision of 92.59%, while YOLOv11s-CLS recorded an accuracy of 92.81% with the lowest parameter count, model size, FLOPs, and inference time of 4.70 ms, as well as the smallest generalization gap of 0.0027. This study concludes that DenseNet-121 is suitable for accuracy-priority scenarios, while YOLOv11s-CLS is recommended as the model with the best performance-efficiency balance for implementation on resource-constrained devices.

Keywords: *convnext-tiny, deep learning, densenet-121, image classification, sugarcane leaf disease, yolov11s-cls*

P-ISSN : 2580-4316
E-ISSN : 2654-8054 https://doi.org/10.37817/ikraith-informatika.v10i2

1 Pendahuluan

Tanaman tebu (Saccharum officinarum) merupakan komoditas perkebunan strategis yang berperan penting dalam pemenuhan kebutuhan industri gula serta ketahanan pangan nasional [1]. Meskipun demikian, produktivitas budi daya tebu kerap terancam oleh serangan berbagai penyakit daun yang mengganggu proses fisiologis tanaman [2], sehingga dapat menurunkan hasil panen dan menimbulkan kerugian finansial yang signifikan bagi petani maupun pengelola perkebunan [3]. Identifikasi penyakit daun tebu di lapangan selama ini masih banyak mengandalkan pengamatan visual manual oleh petani atau tenaga ahli [4]. Pendekatan konvensional tersebut memiliki sejumlah keterbatasan karena memakan banyak waktu, bersifat subjektif, dan rentan terhadap kesalahan manusia [2]. Keterlambatan diagnosis pada tahap awal berisiko mempercepat penyebaran patogen, sehingga diperlukan metode otomatis berbasis citra digital yang lebih cepat dan akurat [4].

Perkembangan *deep learning* menghadirkan solusi untuk otomatisasi deteksi dan klasifikasi penyakit tanaman, karena teknologi ini mampu mempelajari pola visual dari citra daun secara otomatis tanpa memerlukan ekstraksi fitur secara manual [5]. Penerapan *deep learning* pada sektor pertanian juga terbukti mampu mempercepat proses diagnosis sehingga tindakan penanganan dapat diambil dengan lebih tepat sasaran [6]. Sejumlah penelitian terdahulu telah membuktikan keberhasilan *deep learning* dalam mengklasifikasikan penyakit pada berbagai spesies tanaman [7]. Namun demikian, mayoritas penelitian tersebut masih berfokus pada evaluasi satu arsitektur model tertentu atau sekadar mengejar nilai akurasi tertinggi tanpa mempertimbangkan efisiensi komputasi model [8]. Hingga saat ini, studi yang menyajikan perbandingan komprehensif antararsitektur untuk menemukan keseimbangan antara akurasi klasifikasi dan efisiensi komputasi masih terbatas jumlahnya [7].

Untuk memperoleh analisis yang lebih komprehensif, penelitian ini memilih tiga arsitektur *deep learning* dengan paradigma berbeda. DenseNet-121 dipilih sebagai representasi arsitektur *Convolutional Neural Network (CNN)* klasik yang telah luas diakui efisiensi dan keandalannya pada berbagai tugas klasifikasi citra. ConvNeXt-Tiny mewakili generasi *CNN modern* yang dirancang untuk menyaingi performa arsitektur mutakhir dengan ukuran model yang lebih ringkas [9]. Sementara itu, YOLOv11s-CLS digunakan sebagai representasi kerangka kerja klasifikasi generasi terbaru yang menonjolkan kecepatan tinggi dan potensi efisiensi komputasi yang unggul [8]. Ketiga arsitektur tersebut belum banyak dibandingkan secara langsung, khususnya pada kasus klasifikasi penyakit daun tebu [10], sehingga penelitian ini disusun untuk mengisi celah tersebut. Berdasarkan uraian tersebut, penelitian ini bertujuan untuk menganalisis secara komparatif kinerja DenseNet-121, ConvNeXt-Tiny, dan YOLOv11s-CLS dalam mengklasifikasikan delapan kelas kondisi daun tanaman tebu. Evaluasi dilakukan berdasarkan metrik klasifikasi (*accuracy*, *precision*, *recall*, *F1-score*, dan *confusion matrix*), efisiensi komputasi (jumlah parameter, ukuran model, *FLOPs*, dan *inference time*), serta kemampuan generalisasi model berdasarkan nilai *training* dan *validation loss*. Hasil penelitian ini diharapkan dapat menjadi acuan dalam pemilihan arsitektur *deep learning* yang optimal untuk mendukung pengembangan sistem identifikasi penyakit daun tanaman tebu berbasis kecerdasan buatan.

2 Kerangka Studi Kajian dan Logika Pemodelan

Kajian berangkat dari masalah identifikasi penyakit daun tebu yang masih bergantung pada observasi visual manual. Studi terdahulu menunjukkan bahwa transfer learning mampu mempercepat pembentukan model klasifikasi penyakit tanaman, tetapi evaluasi sering kali berhenti pada akurasi tanpa menguji biaya komputasi dan kemampuan generalisasi secara seimbang [5], [7]. Berdasarkan celah tersebut, penelitian dirancang sebagai eksperimen komparatif terkontrol terhadap tiga arsitektur yang mewakili paradigma berbeda, yaitu DenseNet-121 sebagai CNN klasik dengan konektivitas padat, ConvNeXt-Tiny sebagai CNN modern, dan YOLOv11s-CLS sebagai model klasifikasi berorientasi kecepatan. Perbedaan paradigma ini memungkinkan kajian tidak hanya menjawab model mana yang paling akurat, tetapi juga model mana yang paling layak digunakan ketika sumber daya komputasi terbatas.

Kerangka analisis menggunakan tiga dimensi yang saling melengkapi. Dimensi pertama adalah kinerja klasifikasi yang diukur melalui accuracy, precision, recall, F1-score, dan confusion matrix. Dimensi kedua adalah kemampuan generalisasi yang ditelaah melalui hubungan training loss dan validation loss. Dimensi ketiga adalah efisiensi komputasi yang direpresentasikan oleh jumlah parameter, FLOPs, ukuran model, latensi, dan throughput. Dengan kerangka tersebut, hasil akhir tidak ditentukan oleh satu metrik tunggal, melainkan oleh analisis trade-off antara ketepatan prediksi dan kebutuhan sumber daya [20]-[22].

3 Metode Penelitian

3.1 Dataset Penelitian

Penelitian ini menggunakan dataset gabungan dari dua repositori publik Mendeley Data, yaitu *Sugarcane Leaf Dataset* [11] sebagai sumber data utama dan *Sugarcane Leaf Disease Dataset* [12] sebagai pelengkap kelas *Red Rot* yang tidak tersedia pada dataset utama. Setelah proses seleksi kelas yang relevan dengan kondisi penyakit tebu di Indonesia, diperoleh total 4.075 citra RGB yang terbagi ke dalam delapan kelas kondisi daun, yaitu *Healthy, Dried Leaf, Mosaic, Rust, Yellow Leaf, Red Rot, Pokkah Boeng,* dan *Smut*. Seluruh citra kemudian dibagi menggunakan teknik *stratified split* dengan rasio 70:15:15 untuk menjaga proporsi setiap kelas tetap konsisten pada seluruh subset data, mengingat dataset memiliki karakteristik *class imbalance* yang signifikan [13]. Pembagian data dieksekusi sebelum tahap augmentasi untuk mencegah kebocoran data (*data leakage*) antarsubset [7]. Rincian hasil pembagian dataset disajikan pada Tabel 1.

Tabel 1. Distribusi Dataset Hasil *Stratified Split*

Kelas	Train (70%)	Validation (15%)	Test (15%)	Total
Dried Leaf	240	51	52	343
Healthy	301	64	65	430
Mosaic	464	99	100	663
Pokkah Boeng	208	45	44	297
Red Rot	362	78	78	518

Proporsi 15% untuk data validasi dan data uji dipilih agar kelas minoritas (*Pokkah Boeng, Rust, Smut*) tetap memiliki jumlah sampel evaluasi yang representatif dibandingkan rasio 80:10:10 [14]. Tahap awal menggabungkan Sugarcane Leaf Dataset sebagai sumber utama [11] dan Sugarcane Leaf Disease Dataset untuk melengkapi kelas Red Rot [12]. Setelah penyeragaman label, nama folder, dan struktur direktori, dilakukan pemeriksaan file rusak, kosong, tidak terbaca, dan duplikat. Seluruh 4.075 citra RGB dinyatakan valid dan mencakup delapan kelas, yaitu Healthy, Dried Leaf, Mosaic, Rust, Yellow Leaf, Red Rot, Pokkah Boeng, dan Smut. Data kemudian dibagi secara stratified dengan rasio 70:15:15 sebelum augmentasi menggunakan random seed 42 agar proporsi kelas terjaga, kelas minoritas tetap terwakili, dan kebocoran data dapat dicegah. Tabel 1 diperoleh dengan menghitung jumlah citra pada setiap kelas dan subset. Validasi dilakukan dengan memastikan jumlah per kelas sesuai total kelas serta jumlah seluruh kolom mencapai 2.852 citra train, 611 validation, dan 612 test, atau 4.075 citra. Dengan demikian, tabel ini menjadi jejak audit pembagian data. Namun, versi naskah yang ditelaah baru memuat lima kelas; distribusi Rust, Smut, dan Yellow Leaf tidak tersedia dan tidak direkonstruksi.

3.2 Prapemrosesan Data

Seluruh citra melalui tahapan penataan struktur folder per kelas, pembersihan data (*data cleaning*) untuk menghapus citra rusak atau duplikat, penyesuaian ukuran (*resize*) menjadi 224×224 piksel agar seragam dengan input ketiga arsitektur, serta normalisasi nilai piksel menggunakan *mean* dan standar deviasi ImageNet ([0,485; 0,456; 0,406] dan [0,229; 0,224; 0,225]). Augmentasi data — meliputi *horizontal flip* (p=0,5), rotasi acak (±20°),

P-ISSN : 2580-4316
E-ISSN : 2654-8054 https://doi.org/10.37817/ikraith-informatika.v10i2

random resized crop (skala 0,8–1,0), dan *color jitter* (kecerahan dan kontras 0,2) — hanya diterapkan pada data latih untuk memperluas variasi visual dan mengurangi risiko overfitting, sementara data validasi dan uji tetap tidak dimodifikasi agar evaluasi tetap konsisten.

3.3 Perancangan Model

Penelitian ini menerapkan pendekatan *transfer learning* pada ketiga arsitektur menggunakan bobot pra-latih dari ImageNet [5], sehingga model tidak dilatih dari awal yang berisiko tinggi terhadap *overfitting* pada dataset berukuran menengah. Pada DenseNet-121, lapisan klasifikasi bawaan (1.000 kelas ImageNet) digantikan dengan *Global Average Pooling* diikuti lapisan *fully connected* baru dan fungsi aktivasi *Softmax* yang dikonfigurasi untuk delapan kelas target. Pada ConvNeXt-Tiny, penyesuaian dilakukan pada lapisan klasifikasi akhir tanpa mengubah modul depthwise convolution 7×7 pada setiap ConvNeXt block, sehingga representasi fitur fine-grained yang telah terbentuk tetap dipertahankan [9]. Pada YOLOv11s-CLS, varian Small dari YOLOv11 dioperasikan pada mode klasifikasi murni (image-level classification), sehingga lapisan deteksi bounding box beserta fungsi kerugian regresinya ditiadakan. Model memanfaatkan backbone (blok C3k2) dan neck (modul atensi C2PSA) sebagai ekstraktor fitur, dengan head klasifikasi yang dikonfigurasi ulang untuk delapan kelas keluaran [15], [16].

3.4 Konfigurasi Pelatihan

Pelatihan dilakukan hingga maksimal 100 epoch dengan batch size 32 [17], serta mekanisme early stopping untuk menghentikan iterasi apabila *validation loss* tidak menunjukkan perbaikan signifikan [18]. Optimasi bobot menggunakan AdamW dengan learning rate awal 0,001 dan weight decay 0,0001, dipadukan dengan penjadwal Cosine Annealing [8] yang dirumuskan sebagai:

$$\eta_t = \eta_{min} + \frac{1}{2}(\eta_{max} - \eta_{min})\left(1 + \cos\left(\frac{t}{T_{max}}\pi\right)\right) \quad (1)$$

dengan η_t merupakan learning rate pada epoch ke-t, $\eta_{max} = 0{,}001$ sebagai learning rate awal, $T_{max} = 100$ sebagai jumlah maksimum epoch, dan $\eta_{min} = 1 \times 10^{-6}$ sebagai learning rate minimum.

Fungsi kerugian yang digunakan adalah Categorical Cross-Entropy dengan regularisasi Label Smoothing [8], [19], dirumuskan sebagai:

$$L = -\sum_{i=1}^{K} y_i' \log(p_i), \quad y_i' = y_i(1-\varepsilon) + \frac{\varepsilon}{K} \quad (2)$$

dengan K adalah jumlah kelas, y_i adalah label ground truth dalam bentuk one-hot encoding pada kelas ke-i, p_i adalah probabilitas prediksi model untuk kelas ke-i, dan ε merupakan faktor smoothing yang mendistribusikan sebagian probabilitas label ke seluruh kelas sehingga mengurangi kecenderungan model menghasilkan prediksi yang overconfident.

3.5 Skenario Evaluasi

Evaluasi dilakukan pada tiga dimensi: kinerja klasifikasi, kemampuan generalisasi, dan efisiensi komputasi. Kinerja klasifikasi diukur melalui *Accuracy* (3), *Precision* (4), *Recall* (5), dan *F1-score* (6) [8]:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3)$$

$$Precision = \frac{TP}{TP + FP} \quad (4)$$

$$Recall = \frac{TP}{TP + FN} \quad (5)$$

$$F1\ score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (6)$$

https://doi.org/10.37817/ikraith-informatika.v10i2
P-ISSN : 2580-4316
E-ISSN : 2654-8054

Accuracy rentan bias pada data yang tidak seimbang [20], sehingga *Precision*, *Recall*, dan *F1-score* per kelas turut dianalisis melalui *confusion matrix* [8]. Kemampuan generalisasi dievaluasi dengan memantau tren kurva *training loss* dan *validation loss* selama 100 epoch; model dengan kedua kurva yang menurun stabil menuju nol dikategorikan memiliki generalisasi baik, sedangkan pelebaran jarak kedua kurva mengindikasikan *overfitting* [8]. Efisiensi komputasi diukur melalui empat parameter: jumlah parameter (parameter size) yang merepresentasikan kompleksitas model [21], ukuran penyimpanan model (model size) dalam megabita, kompleksitas operasi matematis (*FLOPs*) [22], dan waktu inferensi (*inference time*) dalam milidetik per citra. Keempat parameter efisiensi ini kemudian dianalisis bersama metrik klasifikasi dan *generalization gap* untuk menentukan model dengan keseimbangan performa dan efisiensi terbaik melalui analisis *trade-off*.

4 Hasil dan Pembahasan

4.1 Lingkungan Eksperimen dan Persiapan Data

Eksperimen dilaksanakan menggunakan GPU NVIDIA GeForce RTX 4050 Laptop (6 GB) dengan CUDA 12.4 dan cuDNN 90100. DenseNet-121 dan ConvNeXt-Tiny diimplementasikan pada framework PyTorch, sedangkan YOLOv11s-CLS menggunakan framework Ultralytics dengan cosine learning rate scheduler bawaan (cos_lr=True). Random seed 42 diterapkan untuk menjaga konsistensi eksperimen. Proses *data cleaning* pada 4.075 citra menunjukkan seluruh data valid tanpa file rusak, kosong, atau berformat salah, sehingga seluruh data digunakan pada tahap *stratified split* 70:15:15 yang menghasilkan 2.852 citra data latih, 611 citra data validasi, dan 612 citra data uji.

4.2 Hasil Pelatihan Model

Ketiga model dilatih dengan konfigurasi seragam: optimizer AdamW, scheduler Cosine Annealing, loss function CrossEntropyLoss dengan label smoothing 0,1 dan pembobotan kelas berbasis inverse class frequency, batas maksimum 100 epoch, serta early stopping dengan patience 15 epoch. Ringkasan hasil pelatihan disajikan pada Tabel 2.

Tabel 2. Ringkasan Hasil Pelatihan Model

Model	*Epoch* Dijalankan	*Best Epoch*	*Best Validation Accuracy*	*Training Time*
DenseNet-121	97/100	82	93,62%	57,54 menit
ConvNeXt-Tiny	39/100	24	94,27%	33,31 menit
YOLOv11s-CLS	48/100	33	95,09%	7,41 menit

Seluruh citra diubah menjadi 224 x 224 piksel dan dinormalisasi menggunakan statistik ImageNet. Augmentasi berupa horizontal flip, rotasi acak, random resized crop, dan color jitter hanya diterapkan pada data train, sedangkan validation dan test tidak diubah agar evaluasi tetap objektif. Ketiga model menggunakan transfer learning berbobot ImageNet [5] dengan lapisan akhir delapan kelas: DenseNet-121 mempertahankan dense connectivity, ConvNeXt-Tiny mempertahankan depthwise convolution, dan YOLOv11s-CLS digunakan dalam mode klasifikasi tanpa regresi bounding box. Konfigurasi utama meliputi AdamW, learning rate 0,001, weight decay 0,0001, Cosine Annealing, batch size 32, label smoothing 0,1, pembobotan kelas inverse class frequency, maksimum 100 epoch, dan early stopping patience 15. Tabel 2 diringkas dari log pelatihan berupa jumlah epoch, best epoch, validation accuracy terbaik, dan waktu pelatihan. DenseNet-121 berjalan 97 epoch, terbaik pada epoch 82, dengan validation accuracy 93,62% dan waktu 57,54 menit; ConvNeXt-Tiny berhenti pada 39 epoch, terbaik pada epoch 24, dengan

P-ISSN : 2580-4316
E-ISSN : 2654-8054 https://doi.org/10.37817/ikraith-informatika.v10i2

94,27% dan 33,31 menit; YOLOv11s-CLS berhenti pada 48 epoch, terbaik pada epoch 33, dengan 95,09% dan 7,41 menit. Tabel ini menunjukkan efisiensi konvergensi, bukan kinerja akhir, sehingga penetapan model terbaik tetap harus didasarkan pada data test. Perbedaan pipeline PyTorch dan Ultralytics juga perlu dipertimbangkan ketika membandingkan waktu pelatihan dan generalization gap.

4.3 Performa Klasifikasi pada Data Uji

Evaluasi pada 612 citra data uji yang tidak pernah dilihat model selama pelatihan menghasilkan perbandingan performa sebagaimana Tabel 3.

Tabel 3. Perbandingan Performa Klasifikasi (Macro Average)

Model	*Accuracy*	*Precision*	*Recall*	*F1-score*
DenseNet-121	93,14 %	90,31 %	92,27 %	91,09 %
ConvNeXt-Tiny	92,81 %	92,59 %	91,10 %	90,68 %
YOLOv11s-CLS	92,81 %	91,38 %	91,18 %	90,39 %

Evaluasi dilakukan pada model terbaik tanpa perubahan konfigurasi menggunakan 612 citra test. Probabilitas keluaran delapan kelas diubah menjadi label prediksi berdasarkan nilai tertinggi, lalu dibandingkan dengan label aktual untuk membentuk confusion matrix. Dari matriks tersebut dihitung accuracy secara global serta precision, recall, dan F1-score dengan macro average agar setiap kelas memperoleh bobot yang sama pada dataset tidak seimbang [20]. Tabel 3 menunjukkan DenseNet-121 memperoleh accuracy 93,14%, precision 90,31%, recall 92,27%, dan F1-score 91,09%; ConvNeXt-Tiny memperoleh 92,81%, 92,59%, 91,10%, dan 90,68%; sedangkan YOLOv11s-CLS memperoleh 92,81%, 91,38%, 91,18%, dan 90,39%. DenseNet-121 unggul pada accuracy dan F1-score, sedangkan ConvNeXt-Tiny unggul pada precision, dengan selisih antarmodel kurang dari satu poin persentase. Pada tingkat kelas, seluruh model mencapai F1-score 1,0000 untuk Red Rot, tetapi Pokkah Boeng menjadi kelas tersulit dengan F1-score 0,8000, 0,7143, dan 0,6849. Kesalahan terutama terjadi antara Pokkah Boeng-Smut serta Mosaic-Smut/Healthy, sehingga penambahan data kelas minoritas lebih relevan daripada hanya meningkatkan kapasitas model.

4.4 Analisis Efisiensi Komputasi

Efisiensi komputasi diukur melalui jumlah parameter, ukuran model, *FLOPs*, dan waktu inferensi. Hasil pengukuran disajikan pada Tabel 4 dan Tabel 5.

Tabel 4. Hasil Kompleksitas Model

Model	Parameter	*FLOPs (GFLOPs)*	Model Size (MB)
DenseNet-121	6.962.056	2,896	27,18
ConvNeXt-Tiny	27.826.280	4,463	106,27
YOLOv11s-CLS	5.453.256	0,743	10,53

Profiling dilakukan pada model akhir delapan kelas dengan masukan seragam 224 x 224 piksel. Tiga indikator yang diukur ialah jumlah parameter sebagai representasi bobot model, FLOPs sebagai estimasi operasi matematis per citra, dan model size sebagai kebutuhan penyimpanan. Penyamaan ukuran masukan memastikan perbandingan terutama mencerminkan perbedaan arsitektur. Tabel 4 mencatat DenseNet-121 memiliki 6.962.056 parameter, 2,896 GFLOPs, dan ukuran 27,18 MB; ConvNeXt-Tiny memiliki 27.826.280 parameter, 4,463 GFLOPs, dan 106,27 MB; sedangkan YOLOv11s-CLS

https://doi.org/10.37817/ikraith-informatika.v10i2
P-ISSN : 2580-4316
E-ISSN : 2654-8054

memiliki 5.453.256 parameter, 0,743 GFLOPs, dan 10,53 MB. ConvNeXt-Tiny mempunyai kapasitas terbesar tetapi tidak meningkatkan accuracy dibandingkan DenseNet-121, sementara YOLOv11s-CLS menjadi model paling ringan dan paling sesuai untuk perangkat mobile atau edge.

Tabel 5. Hasil Benchmark Inferensi

Model	Single Latency (ms)	Single FPS	Batch Latency (ms)	Batch Throughput (FPS)
DenseNet-121	12,15	82,29	79,16	404,22
ConvNeXt-Tiny	5,51	181,54	78,33	408,50
YOLOv11s-CLS	4,70	212,70	14,80	2.162,44

Benchmark inferensi dilakukan pada GPU NVIDIA GeForce RTX 4050 Laptop 6 GB dengan CUDA 12.4 dan cuDNN 90100. Skenario single inference mengukur latensi satu citra, sedangkan batch inference menggunakan 32 citra. Single FPS dihitung dari 1.000 dibagi latensi dalam milidetik, sementara batch throughput dihitung dari jumlah citra dibagi waktu batch dalam detik. Kedua skenario digunakan karena kecepatan satu citra tidak selalu sejalan dengan kemampuan pemrosesan paralel. Tabel 5 menunjukkan DenseNet-121 mencapai latensi 12,15 ms atau 82,29 FPS dan batch latency 79,16 ms dengan throughput 404,22 FPS; ConvNeXt-Tiny mencapai 5,51 ms atau 181,54 FPS dan 78,33 ms dengan 408,50 FPS; sedangkan YOLOv11s-CLS mencapai 4,70 ms atau 212,70 FPS dan 14,80 ms dengan 2.162,44 FPS. YOLOv11s-CLS menjadi model tercepat pada kedua skenario. Nilai absolut benchmark tetap bergantung pada perangkat keras, pustaka, mode eksekusi, dan framework, tetapi urutan relatif antarmodel tetap valid karena seluruh pengujian dilakukan pada lingkungan yang sama. ConvNeXt-Tiny memiliki kompleksitas tertinggi (parameter terbanyak, *FLOPs* terbesar, model terbesar) akibat kapasitas representasi fiturnya yang lebih tinggi, namun kompleksitas ini tidak sebanding dengan peningkatan akurasi dibandingkan DenseNet-121. Sebaliknya, YOLOv11s-CLS tampil sebagai model paling ringan pada seluruh indikator efisiensi sekaligus tercepat, dengan keunggulan throughput yang sangat signifikan pada skenario batch inference (2.162,44 FPS), menjadikannya paling potensial untuk implementasi real-time pada perangkat dengan sumber daya terbatas.

4.5 Analisis *Trade-off* dan Perbandingan dengan Penelitian Terdahulu

Dibandingkan penelitian DenseNet-121 terdahulu yang melaporkan akurasi hingga 99,81% [23] dan 99,85% [24] pada dataset penyakit tanaman lain, hasil penelitian ini (93,14%) relatif lebih rendah, kemungkinan disebabkan oleh kompleksitas visual dataset tebu yang diambil pada kondisi lapangan nyata (real-field conditions) dibandingkan dataset yang lebih terkendali seperti PlantVillage. Efisiensi DenseNet-121 pada penelitian ini (6,96 juta parameter, 27,18 MB) tetap konsisten dengan temuan bahwa arsitektur ini menawarkan keseimbangan performa-efisiensi yang baik dibandingkan CNN konvensional lain [25].

Pada ConvNeXt-Tiny, penelitian terdahulu yang menambahkan modul attention atau deformable convolution melaporkan peningkatan akurasi yang lebih tinggi [26], [27]. Karena penelitian ini menggunakan ConvNeXt-Tiny standar tanpa modifikasi tambahan, hasil yang diperoleh (92,81%) merepresentasikan kemampuan dasar arsitektur tersebut. Pada YOLOv11s-CLS, hasil penelitian ini (92,81%) berada di bawah penelitian yang menerapkan augmentasi sintetis Stable Diffusion dengan akurasi 99,44% [13], namun keunggulan utama YOLOv11s-CLS pada penelitian ini justru konsisten dengan reputasinya sebagai arsitektur yang unggul dalam efisiensi dan kecepatan pemrosesan [10].

Analisis *trade-off* menunjukkan bahwa selisih akurasi antara YOLOv11s-CLS dan

DenseNet-121 hanya sebesar 0,33%, sementara YOLOv11s-CLS menawarkan pengurangan parameter sebesar ±22%, ukuran model ±61% lebih kecil, *FLOPs* ±74% lebih rendah, latensi ±61% lebih cepat, serta *generalization gap* yang jauh lebih terkendali. Dengan demikian, DenseNet-121 direkomendasikan sebagai model dengan performa klasifikasi terbaik, sedangkan YOLOv11s-CLS direkomendasikan sebagai model dengan keseimbangan performa-efisiensi paling optimal untuk implementasi pada perangkat berdaya komputasi terbatas. Hasil ini menegaskan bahwa pemilihan arsitektur *deep learning* untuk klasifikasi penyakit daun tebu tidak dapat hanya bergantung pada nilai akurasi, melainkan harus mempertimbangkan kebutuhan implementasi secara menyeluruh.

5 Kesimpulan

Penelitian ini berhasil membandingkan kinerja tiga arsitektur *deep learning* dengan paradigma berbeda, yaitu DenseNet-121, ConvNeXt-Tiny, dan YOLOv11s-CLS, dalam mengklasifikasikan delapan kelas kondisi daun tanaman tebu berdasarkan tiga dimensi evaluasi: kinerja klasifikasi, kemampuan generalisasi, dan efisiensi komputasi. Hasil pengujian pada data uji yang independen menunjukkan bahwa ketiga arsitektur mampu mencapai tingkat akurasi yang kompetitif dengan selisih antarmodel yang relatif kecil, di mana DenseNet-121 unggul pada metrik *accuracy* dan *F1-score*, sedangkan ConvNeXt-Tiny memimpin pada *precision*. Meskipun demikian, ConvNeXt-Tiny yang memiliki jumlah parameter dan kebutuhan komputasi paling besar tidak menghasilkan peningkatan performa klasifikasi yang sebanding, sehingga menunjukkan bahwa kompleksitas arsitektur yang lebih tinggi tidak selalu berbanding lurus dengan kualitas prediksi pada kasus klasifikasi penyakit daun tebu. Sebaliknya, YOLOv11s-CLS membuktikan diri sebagai arsitektur dengan efisiensi komputasi paling unggul, ditandai oleh jumlah parameter, ukuran model, dan *FLOPs* paling rendah, waktu inferensi tercepat, serta kesenjangan generalisasi paling terkendali, dengan selisih akurasi yang hanya berbeda tipis dari DenseNet-121. Temuan ini menegaskan bahwa pemilihan arsitektur *deep learning* untuk sistem identifikasi penyakit daun tebu tidak dapat hanya didasarkan pada nilai akurasi tertinggi, melainkan harus mempertimbangkan *trade-off* antara ketangguhan performa klasifikasi dan kebutuhan sumber daya komputasi sesuai konteks implementasinya. Berdasarkan analisis *trade-off* tersebut, DenseNet-121 lebih sesuai digunakan pada skenario yang memprioritaskan akurasi klasifikasi setinggi mungkin, sedangkan YOLOv11s-CLS lebih ideal untuk diimplementasikan pada sistem diagnosis berbasis perangkat mobile atau perangkat tepi (edge devices) yang menuntut kecepatan inferensi tinggi dengan sumber daya komputasi terbatas. Penelitian selanjutnya disarankan untuk memperluas cakupan penelitian melalui penambahan varian arsitektur, modifikasi mekanisme atensi pada model dasar, serta perluasan volume data pada kelas minoritas seperti Pokkah Boeng dan Smut guna meningkatkan kemampuan model dalam mengenali kelas dengan karakteristik visual yang saling menyerupai.

Daftar Pustaka

[1] X. Li, X. Li, S. Zhang, G. Zhang, M. Zhang, and H. Shang, “SLViT: Shuffle-convolution-based lightweight Vision transformer for effective diagnosis of sugarcane leaf diseases,” *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 6, Jun. 2023, doi: 10.1016/j.jksuci.2022.09.013.

[2] M. S *et al.*, “5G-enabled interactive sugarcane protection networks for farmers and experts: Custom sugarcane dataset trained SPIDCNet for under-canopy sugarcane insect and disease detection,” *Smart Agricultural Technology*, vol. 14, Aug. 2026, doi: 10.1016/j.atech.2026.102134.

[3] L. K. Putra, A. Kristini, and W. W. Jati, “Viral Diseases of Sugarcane in Indonesia: Occurrence Notes, Pathogenic Characteristics and Management

Strategies," in *AIP Conference Proceedings*, American Institute of Physics Inc., Jan. 2023. doi: 10.1063/5.0116089.

[4] M. M. F. Ulum, M. Sholihin, and M. Mustain, "Implementation of ResNet50 Based on Transfer Learning for Sugarcane Leaf Disease Detection," Pendidikan dan Informatika, 2025. doi: https://doi.org/10.21107/edutic.v12i2.31873.

[5] D. J. Richter, M. Ilias Bappi, S. Sanjay Kolekar, and K. Kim, "A Systematic Review of the Current State of Transfer Learning Accelerated CNN-Based Plant Leaf Disease Classification," 2025, *Institute of Electrical and Electronics Engineers Inc.* doi: 10.1109/ACCESS.2025.3584404.

[6] S. Yu and L. Liu, "Lesion-Aware Enhanced Swin Transformer for Plant Disease Identification," in *Proceedings of 2025 5th International Conference on Big Data, Artificial Intelligence and Risk Management, ICBAR 2025*, Association for Computing Machinery, Inc, Apr. 2026, pp. 326–333. doi: 10.1145/3800227.3800277.

[7] N. N. Bhaliya and B. Talati, "A Comparative Variability Analysis of Pre-Trained Deep Learning Models for Sugarcane Leaf Disease Classification," *Journal of Computing and Biomedical Informatics*, vol. 10, no. 2, 2026, doi: 10.56979/1002/2026/1208.

[8] E. H. I. Eliwa and T. Abd El-Hafeez, "A novel YOLOv11-based framework with dynamic cross-scale context aggregation for high-performance blood cell classification," *Egyptian Informatics Journal*, vol. 32, Dec. 2025, doi: 10.1016/j.eij.2025.100851.

[9] M. Trigka and E. Dritsas, "A Comprehensive Survey of Deep Learning Approaches in Image Processing," 2025. doi: 10.3390/s25020531.

[10] A. Wijaya and N. Rachmat, "Klasifikasi Penyakit Daun Mangga Menggunakan YOLOv11 Berbasis Deep Learning dan Computer Vision," vol. 6, no. 8, pp. 1444–1451, 2026, doi: 10.47065/tin.v6i8.9168.

[11] S. Thite, Y. Suryawanshi, K. Patil, and P. Chumchu, "Sugarcane leaf dataset: A dataset for disease detection and classification for machine learning applications," *Data Brief*, vol. 53, Apr. 2024, doi: 10.1016/j.dib.2024.110268.

[12] "Sugarcane Leaf Disease Dataset - Mendeley Data." Accessed: Jul. 27, 2026. [Online]. Available: https://data.mendeley.com/datasets/9424skmnrk/1

[13] N. M. Opu, M. S. Islam, S. H. Abdullah, and R. Akter, "Deep Learning-Driven Leaf Disease Classification in Indoor Plants with YOLOv11: Stable Diffusion Augmentation Approach," in *2025 28th International Conference on Computer and Information Technology, ICCIT 2025*, Institute of Electrical and Electronics Engineers Inc., 2025, pp. 443–448. doi: 10.1109/ICCIT68739.2025.11491324.

[14] L. C. Ngugi, M. Abelwahab, M. Abo-zahhad, L. C. Ngugi, M. Abelwahab, and M. Abo-zahhad, "Recent Advances in Image Processing Techniques for Automated Leaf Pest and Disease Recognition - A Review," *Information Processing in Agriculture*, 2020, doi: 10.1016/j.inpa.2020.04.004.

[15] E. H. Alkhammash, "Multi-Classification Using YOLOv11 and Hybrid YOLO11n-MobileNet Models: A Fire Classes Case Study," *Fire*, vol. 8, no. 1, Jan. 2025, doi: 10.3390/fire8010017.

[16] T. B. Nyawose, R. C. Maswanganyi, and P. Khumalo, "Multi-Plant Real-Time Disease Detection and Classification Using YOLOv11," *IEEE Access*, 2026, doi: 10.1109/ACCESS.2026.3702738.

[17] A. S. Alzaharani and A. Iqbal, "A Multi-Stage Pipeline for Date Fruit Processing: Integrating YOLOv11 Detection, Classification, and Automated Counting," *Computers, Materials and Continua*, vol. 86, no. 1, 2026, doi: 10.32604/cmc.2025.070410.

[18] D. Dai, P. Xia, Z. Zhu, and H. Che, "MTDL-EPDCLD: A Multi-Task Deep-Learning-Based System for Enhanced Precision Detection and Diagnosis of Corn Leaf Diseases," *Plants*, vol. 12, no. 13, Jul. 2023, doi: 10.3390/plants12132433.

[19] A. Ismail, W. Hamdy, A. H. Ibrahim, and W. A. Awad, "Classification of rice plant diseases using efficient DenseNet121," pp. 1–13, 2026.

[20] G. M. Foody, "Challenges in the real world use of classification accuracy metrics: From recall and precision to the Matthews correlation coefficient," pp. 1–27, 2023, doi: 10.1371/journal.pone.0291908.

[21] I. Kunduracioglu and I. Pacal, "Deep Learning-Based Disease Detection in Sugarcane Leaves: Evaluating EfficientNet Models," *Journal of Operations Intelligence*, vol. 2, no. 1, pp. 321–335, Jan. 2024, doi: 10.31181/jopi21202423.

[22] A. R. K. P and S. Gowrishankar, "ConvNeXt - based Mango Leaf Disease Detection : Differentiating Pathogens and Pests for Improved Accuracy," vol. 14, no. 6, pp. 73–82, 2023.

[23] R. Pillai, N. Sharma, R. Gupta, and A. Sharma, *Classification of Plant Diseases using DenseNet 121 Transfer Learning Model*. 2023. doi: 10.1109/ACCAI58221.2023.10200401.

[24] A. Nag, P. R. Chanda, and S. Nandi, "Mobile app-based tomato disease identification with fine-tuned convolutional neural networks," *Computers and Electrical Engineering*, vol. 112, p. 108995, 2023, doi: https://doi.org/10.1016/j.compeleceng.2023.108995.

[25] S. J. Wei, D. F. Al Riza, and H. Nugroho, "Comparative study on the performance of deep learning implementation in the edge computing: Case study on the plant leaf disease identification," *J. Agric. Food Res.*, vol. 10, p. 100389, 2022, doi: https://doi.org/10.1016/j.jafr.2022.100389.

[26] Q. Wu, X. Ma, H. Liu, C. Bi, H. Yu, and M. Liang, "A classification method for soybean leaf diseases based on an improved ConvNeXt model," pp. 1–11, 2023, doi: 10.1038/s41598-023-46492-3.

[27] H. Li *et al.*, "Maize Disease Classification System Design Based on Improved ConvNeXt," pp. 1–16, 2023.