

Journal of Comprehensive Science
p-ISSN: 2962-4738 e-ISSN: 2962-4584
Vol. 3. No. 12 Desember 2024

Penerapan Transfer Learning Dengan Inception-V3 Dan Efficientnet-B4 Pada Studi Kasus Klasifikasi Penyakit Pada Daun Singkong

Tri Anton, Arief Setyanto, Dhani Ariatmanto

Universitas Amikom Yogyakarta, Indonesia

Email: trianton@students.amikom.ac.id, arief_s@amikom.ac.id, dhaniari@amikom.ac.id

Abstrak

Singkong adalah tanaman yang memiliki permintaan yang tinggi di Indonesia, ditandai dengan peningkatan tingkat produksi seiring waktu. Selain kuantitas, kualitas tanaman harus dijaga, salah satunya dengan memperhatikan gejala penyakit. Gejala penyakit yang timbul pada daun singkong dapat dideteksi dengan inspeksi visual. Tetapi, diperlukan pengetahuan lebih untuk membedakan gejala suatu penyakit dengan penyakit lainnya. Salah satu solusi dari masalah ini adalah pemanfaatan convolutional neural networks (CNN) untuk klasifikasi penyakit. Penulis menggunakan model CNN untuk masalah ini. Parameter penilai kinerja model CNN yang digunakan adalah akurasi, presisi, recall, dan F1-score. penelitian ini akan menggunakan dua arsitektur pada transfer learning, yaitu EfficientNet-B4 dan Inception-V3. Kedua arsitektur ini masih jarang digunakan pada studi kasus terkait. Tujuan peningkatan jumlah parameter ini adalah untuk menemukan konfigurasi optimal pada optimizer dan learning rate yang dapat memaksimalkan performa model. Dengan peningkatan jumlah parameter dan pemanfaatan dua arsitektur pada transfer learning, diharapkan kemampuan model dalam menangani kompleksitas masalah klasifikasi gambar daun singkong penyakit dapat ditingkatkan. Fokus penelitian ini juga akan difokuskan pada penerapan arsitektur EfficientNet-B4 dan Inception-V3 dengan skema hyperparameter tuning untuk meningkatkan performa model. Oleh karena itu, penelitian ini diharapkan dapat memberikan kontribusi yang lebih unggul dalam pengembangan CNN untuk klasifikasi penyakit pada daun singkong, dengan performa yang lebih baik dan lebih akurat.

Kata kunci: penyakit singkong, convolutional neural network, computer vision, image classification, densenet

Abstract

Cassava is a crop that has high demand in Indonesia, marked by increasing production levels over time. In addition to quantity, crop quality must be maintained, one of which is by paying attention to disease symptoms. Disease symptoms that appear on cassava leaves can be detected by visual inspection. However, more knowledge is needed to distinguish the symptoms of one disease from another. One solution to this problem is the use of convolutional neural networks (CNN) for disease classification. The author uses a CNN model for this problem. The performance assessment parameters of the CNN model used are accuracy, precision, recall, and F1-score. This study will use two architectures in transfer learning, namely EfficientNet-B4 and Inception-V3. Both of these architectures are still rarely used in related case studies. The purpose of increasing the number of parameters is to find the optimal configuration of the optimizer and learning rate that can maximize model performance. By increasing the number of parameters and utilizing two architectures in transfer learning, it is hoped that the model's

ability to handle the complexity of the problem of classifying images of cassava leaves with disease can be improved. The focus of this study will also be focused on the application of the EfficientNet-B4 and Inception-V3 architectures with a hyperparameter tuning scheme to improve model performance. Therefore, this research is expected to provide a superior contribution in the development of CNN for disease classification in cassava leaves, with better and more accurate performance.

Keyword: *cassava diseases, convolutional neural networks, computer vision, image classification, densenet*

PENDAHULUAN

Tanaman singkong merupakan salah satu komoditas pertanian yang memiliki peran strategis dalam memenuhi kebutuhan pangan masyarakat Indonesia (Salim, 2024). Selain sebagai sumber karbohidrat yang signifikan, singkong juga memiliki potensi untuk diolah menjadi berbagai produk olahan dengan nilai tambah tinggi, seperti tepung tapioka, keripik singkong, dan makanan ringan lainnya (Dewiani et al., 2024).

Meski demikian, di tengah potensi besar yang dimiliki, tanaman singkong juga menghadapi tantangan serius, terutama terkait dengan penyakit yang dapat memengaruhi produktivitas dan kualitas hasil. Penyakit pada tanaman singkong, seperti antraknosa, bercak daun, dan virus daun kering, dapat merugikan petani dan berpotensi menghambat pertumbuhan industri singkong (Saparinto & Susiana, 2024). Pemahaman mendalam tentang penyakit-penyakit ini dan implementasi strategi pengelolaan penyakit yang efektif menjadi krusial untuk menjaga keberlanjutan produksi singkong. Dalam upaya menghadapi tantangan ini, penerapan teknologi modern, seperti kecerdasan buatan, menjadi semakin relevan. Pemanfaatan algoritma kecerdasan buatan dapat memberikan deteksi dini terhadap gejala penyakit, memungkinkan petani untuk merespons lebih cepat dan menerapkan langkah-langkah pengendalian yang efektif (Hartati, 2021; Putri et al., 2024).

Studi terdahulu oleh Masita, (2023) menunjukkan bahwa penyakit bercak daun merupakan salah satu penyebab utama penurunan kualitas singkong di Lampung Barat, sementara penelitian oleh Ramadhan, (2020) mengidentifikasi virus daun kering sebagai ancaman yang terus meningkat di sentra produksi singkong. Meski berbagai metode konvensional telah diterapkan, deteksi dini dan akurasi identifikasi penyakit masih menjadi kendala utama dalam pengelolaan penyakit tanaman ini.

Urgensi penelitian ini terletak pada kebutuhan akan solusi yang lebih efektif dan efisien dalam mendeteksi penyakit singkong sejak dini, mengingat peningkatan insiden penyakit yang berdampak pada ketahanan pangan nasional. Novelty dari penelitian ini terletak pada penerapan dua arsitektur deep learning, yakni EfficientNet-B4 dan Inception-V3, yang dikombinasikan dengan skema hyperparameter tuning untuk meningkatkan akurasi klasifikasi gambar daun singkong yang terinfeksi. Pendekatan ini belum banyak diadopsi dalam studi serupa, sehingga diharapkan mampu memberikan kontribusi baru dalam pengembangan model deteksi penyakit tanaman berbasis kecerdasan buatan (AI).

Penelitian ini bertujuan untuk mengembangkan model CNN yang mampu mengklasifikasikan penyakit pada daun singkong dengan akurasi yang lebih tinggi dibandingkan metode konvensional. Manfaat dari penelitian ini mencakup peningkatan produktivitas petani melalui deteksi dini dan pengelolaan penyakit yang lebih efektif, serta kontribusi dalam pengembangan teknologi pertanian berbasis AI di Indonesia. Dengan peningkatan jumlah parameter dan pemanfaatan dua arsitektur pada transfer learning, diharapkan kemampuan model dalam menangani kompleksitas masalah klasifikasi gambar daun singkong penyakit dapat ditingkatkan. Fokus penelitian ini juga akan difokuskan pada penerapan arsitektur EfficientNet-B4 dan Inception-V3 dengan skema *hyperparameter tuning* untuk meningkatkan performa model. Oleh karena itu, penelitian ini diharapkan dapat

memberikan kontribusi yang lebih unggul dalam pengembangan CNN untuk klasifikasi penyakit pada daun singkong, dengan performa yang lebih baik dan lebih akurat.

METODE PENELITIAN

Penelitian ini menggunakan desain penelitian eksperimen (*experimental research*). Desain penelitian eksperimen adalah metode penelitian yang melibatkan manipulasi variabel independen untuk mengamati efeknya pada variabel dependen, dengan tujuan menentukan hubungan sebab-akibat (Santoso & Madiistriyatno, 2021). Penelitian eksperimen sering dianggap sebagai metode penelitian yang paling kuat dalam mengidentifikasi hubungan kausal karena kontrol yang ketat terhadap variabel-variabel yang terlibat (Ibrahim et al., 2023).

Adapun tahapan yang dilakukan dalam penelitian ini adalah sebagai berikut.

1. Pengumpulan Data Sekunder

Dataset yang digunakan berasal dari platform Kaggle.

2. Augmentasi Gambar

Augmentasi gambar adalah teknik yang digunakan dalam pemrosesan citra untuk memperbanyak dan memperkaya dataset gambar tanpa harus mengumpulkan gambar tambahan (Yusuf et al., 2024). Teknik ini sangat penting dalam penelitian yang melibatkan model pembelajaran mesin dan pembelajaran mendalam, terutama ketika dataset asli tidak cukup besar atau beragam untuk melatih model dengan baik. Augmentasi gambar membantu dalam meningkatkan performa dan generalisasi model dengan membuatnya lebih tahan terhadap variasi dalam data.

3. Pemrosesan Gambar

Pemrosesan gambar adalah serangkaian teknik yang digunakan untuk memanipulasi dan menganalisis gambar digital dengan tujuan meningkatkan kualitas gambar, mengekstrak informasi penting, atau mempersiapkan gambar untuk analisis lebih lanjut dalam sistem berbasis kecerdasan buatan (Gunawan et al., 2023). Pemrosesan gambar merupakan langkah penting dalam banyak aplikasi, termasuk pengenalan pola, visi komputer, dan pembelajaran mesin.

4. Pembagian Data

Dataset dibagi menjadi data pelatihan dan data pengujian dengan proporsi 80:20%

5. Pembangunan Model

Setelah pembagian data, langkah selanjutnya adalah pembangunan model menggunakan arsitektur InceptionV3 dan EfficientNetB4. Kedua arsitektur ini digunakan untuk membangun dan melatih model pembelajaran mesin yang dapat mengenali pola dan fitur dari dataset yang telah dibagi. Dalam proses pembangunan model ini, berbagai kombinasi optimizer dan learning rate akan diuji untuk menemukan konfigurasi terbaik yang menghasilkan kinerja optimal.

6. Evaluasi Model

Evaluasi model merupakan langkah penting dalam pengembangan sistem pembelajaran mesin. Ini memungkinkan kita untuk mengukur kinerja model yang telah dibangun dan memastikan bahwa model tersebut dapat digunakan secara efektif dalam kasus penggunaan yang sesuai. Evaluasi model melibatkan penggunaan metrik evaluasi yang relevan untuk mengevaluasi kinerja model terhadap data pengujian yang belum pernah dilihat sebelumnya. Dalam evaluasi model, akan menggunakan metrik akurasi, recall, presisi, dan F1-Score.

HASIL DAN PEMBAHASAN

Dataset

Data yang digunakan adalah dataset *casava leaf disease* yang tersedia di Kaggle <https://www.kaggle.com/datasets/gauravduttakiit/cassava-leaf-disease-classification>.

Dataset berupa gambar berwarna dengan format .jpg yang berjumlah 21.308 dengan label *Cassava Bacterial Blight (CBB)*, *Cassava Brown Streak Disease (CBSD)*, *Cassava Green Mottle (CGM)*, *Cassava Mosaic Disease (CMD)*, dan *Healthy*. Berikut ini merupakan contoh gambar dari setiap label:



Gambar 1. Healthy



Gambar 2. CMD



Gambar 3. CGM



Gambar 4. CBSD



Gambar 5. CBB

Gambar 4. 1 CBSD

Prapemrosesan Gambar

Pemrosesan data dilakukan melalui dua tahapan yaitu dengan mengubah ukuran gambar menjadi 128x128 *piksel* dan untuk mengubah skala intensitas warna pada gambar yang semula 0 – 255 menjadi hanya 0 – 1.

Augmentasi Gambar

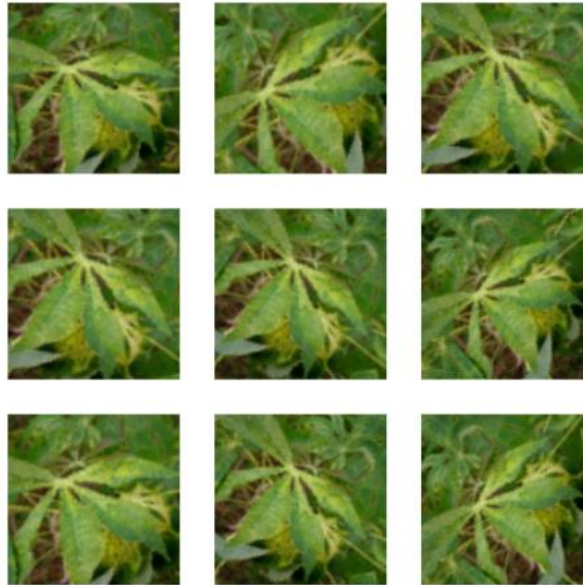
Dalam penelitian ini, teknik augmentasi gambar diterapkan untuk memperkaya dataset dan meningkatkan variasi data pelatihan (Abel et al., 2024). Dua teknik augmentasi utama yang digunakan *Random Rotation* dan *Random Zoom*. Adapun *sourcecode* yang digunakan yaitu sebagai berikut:

Tabel 1. *Sourcecode Augmentasi Gambar*

```
def data_augmentar():
    data_augmentation = Sequential()
    data_augmentation.add(RandomRotation(factor=(-0.15,0.15)))
    data_augmentation.add(RandomZoom((-0.3, -0.1)))

    return data_augmentation
```

Pada tabel 1. Fungsi `data_augmentar()` diimplementasikan untuk melakukan augmentasi pada gambar daun singkong. Fungsi ini membuat sebuah pipeline augmentasi data menggunakan model `Sequential` dari Keras. Selanjutnya baru diterapkan dua teknik augmentasi yaitu merotasi gambar secara acak dengan sudut antara -0.15 hingga 0.15 radian untuk membantu model mengenali gejala penyakit pada daun singkong dari berbagai sudut orientasi dan memperbesar gambar secara acak dengan faktor antara -0.3 hingga -0.1 tujuannya yaitu untuk meningkatkan kemampuan model dalam mendeteksi gejala penyakit pada skala yang berbeda. Adapun hasil dari proses augmentasi ini yaitu sebagai berikut :



Gambar 2. Hasil Proses *Augmentasi*

Pengujian dengan *Struktur Inception-V3*

Pengujian ini dilakukan menggunakan Arsitektur *Convolutional Neural Network* (CNN) menggunakan struktur *Inception-V3*. Berikut adalah *source code* untuk arsitektur model yang digunakan:

Tabel 2. Sourcode arsitektur CNN *Inception-V3*

```
def alzheimer_classifier(image_shape=IMG_SIZE,
data_augmentation=data_augmentar()):

    IMG_SHAPE = image_shape + (3,)
    base_model = applications.InceptionV3(input_shape=IMG_SHAPE,
include_top=False,
weights='imagenet')
    base_model.trainable = True
    for layer in base_model.layers[0:218]:
        layer.trainable = False

    inputs = Input(shape=IMG_SHAPE)
    x = data_augmentation(inputs)
    x = preprocess_input(inputs)
    x = base_model(x)
    x = GlobalAveragePooling2D()(x)
    outputs = Dense(units=5, activation = "softmax")(x)

    model = Model(inputs, outputs)
    return model
```

Pada tabel 2. merupakan *source code* arsitektur CNN yang menggunakan struktur *Inception-V3*. Model ini menggunakan pendekatan transfer learning dengan memanfaatkan bobot pre-trained dari ImageNet pada 218 lapisan awal yang dibekukan, sementara lapisan-lapisan akhir dapat di *fine-tune* untuk menyesuaikan dengan tugas spesifik. Untuk meningkatkan variasi data dan ketahanan model, augmentasi real-time diterapkan

menggunakan rotasi dan *zoom* acak, sebagaimana penelitian yang dilakukan oleh Predianto & Sutomo, (2024). Setelah feature ekstraksi, *Global Average Pooling* digunakan untuk mengurangi dimensi dan mencegah *overfitting*. Terakhir, terdapat *Dense Layer* dengan 5 *neuron* dan *aktivasi softmax* berfungsi sebagai lapisan *output* untuk klasifikasi *multi-kelas* penyakit daun singkong. Adapun detail dari arsitektur model yang digunakan yaitu sebagai berikut:

Layer (type)	Output Shape	Param #
input_21 (InputLayer)	[(None, 128, 128, 3)]	0
tf.math.truediv (TFOpLambda)	(None, 128, 128, 3)	0
tf.math.subtract (TFOpLambda)	(None, 128, 128, 3)	0
inception_v3 (Functional)	(None, 2, 2, 2048)	21802784
global_average_pooling2d_3 (GlobalAveragePooling2D)	(None, 2048)	0
dense_3 (Dense)	(None, 5)	10245

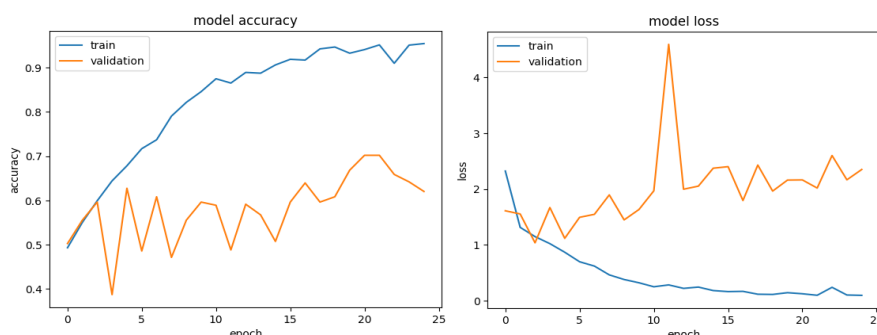
=====
 Total params: 21813029 (83.21 MB)
 Trainable params: 13228421 (50.46 MB)
 Non-trainable params: 8584608 (32.75 MB)

Gambar 3. arsitektur *CNN Inception-V3*

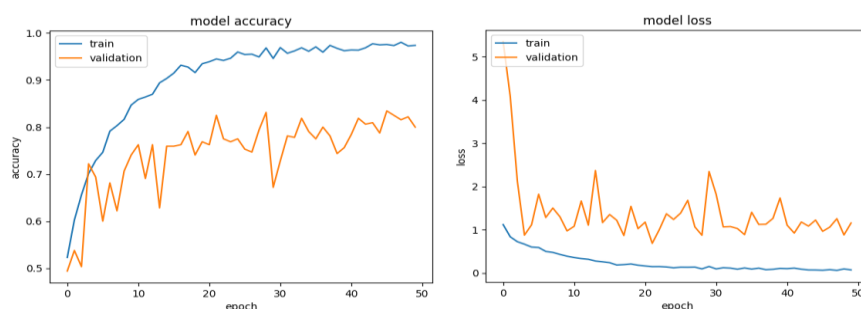
Gambar 7. merupakan arsitektur yang terdiri dari lapisan *Inception-V3* yang diikuti oleh lapisan Input yang menerima gambar berukuran 128x128 *piksel* dengan 3 saluran warna (RGB) Selanjutnya, terdapat dua operasi normalisasi yaitu *tf.math.truediv* dan *tf.math.subtract* yang bertujuan untuk menormalisasi data input. Setelah itu, lapisan utama adalah *Inception-V3* yang digunakan sebagai *backbone* untuk ekstraksi fitur, menghasilkan *output* dengan ukuran [None, 2, 2, 2048]. Setelah ekstraksi fitur, lapisan *Global Average Pooling* diterapkan untuk mengubah ukuran *output*. Lapisan terakhir adalah *Dense Layer* dengan 5 neuron yang digunakan untuk klasifikasi akhir. Secara keseluruhan, model ini memiliki total 21.813.029 parameter, dengan 13.228.421 di antaranya adalah parameter yang dapat dilatih dan 8.584.608 adalah parameter yang tidak dapat dilatih.

Hasil Pengujian Menggunakan *Optimizer Adam*

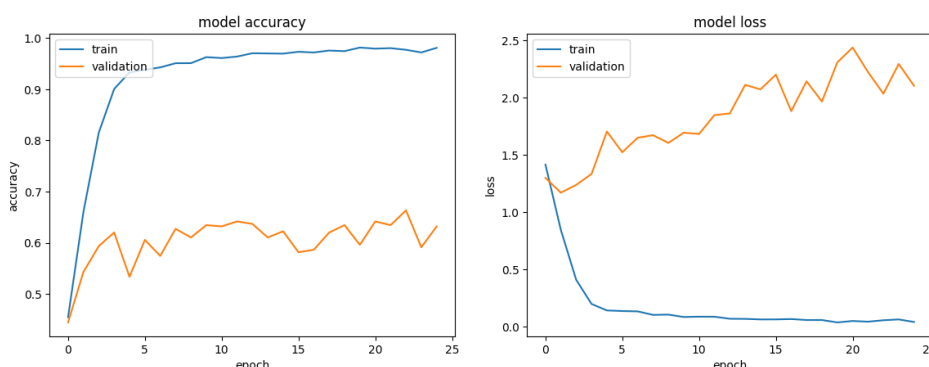
Pengujian ini dilakukan dengan menggunakan *optimizer Adam* dengan *Learning Rate* sebesar 0.01, 0.001, dan 0.001. Hasilnya adalah sebagai berikut:



Gambar 8. Grafik akurasi *loss* menggunakan *optimizer adam* dengan *learning rate 0.01*



Gambar 4. Grafik akurasi loss menggunakan *optimizer adam* dengan *learning rate* 0.001



Gambar 5. Grafik akurasi loss menggunakan *optimizer adam* dengan *learning rate* 0.0001

Dapat dilihat pada ke 3 gambar tersebut, model dengan *learning rate* 0.001 menunjukkan performa terbaik meskipun masih ada indikasi *overfitting*. Namun fluktuasi pada kurva *validation* dan *training* tidak seburuk model dengan *learning rate* 0.01 dan 0.0001. Selain itu, model dengan *learning rate* 0.001 memiliki keseimbangan yang lebih baik antara penurunan *loss* pelatihan dan akurasi validasi yang meningkat secara signifikan dibandingkan dengan dua model lainnya. Adapun hasil perbandingannya berdasarkan hasil *training* setiap model adalah sebagai berikut:

Tabel 3. Hasil *Training* Struktur *Inception-V3* Menggunakan *Optimizer Adam*

<i>Learning Rate</i>	<i>Accuracy (%)</i>	<i>Val Accuracy (%)</i>	<i>Loss</i>	<i>Val Loss</i>
0.01	95.44	62.02	0.0944	2.3508
0.001	94.11	46.15	0.1661	3.0455
0.0001	98.07	63.22	0.0419	2.1040

Selain itu, hasil perbandingan metrik performa pada *data train* dan *data test* untuk setiap model dengan *learning rate* yang berbeda adalah sebagai berikut:

Tabel 4. *Metrik Performa* Pada Struktur *Inception-V3* Menggunakan *Optimizer Adam*

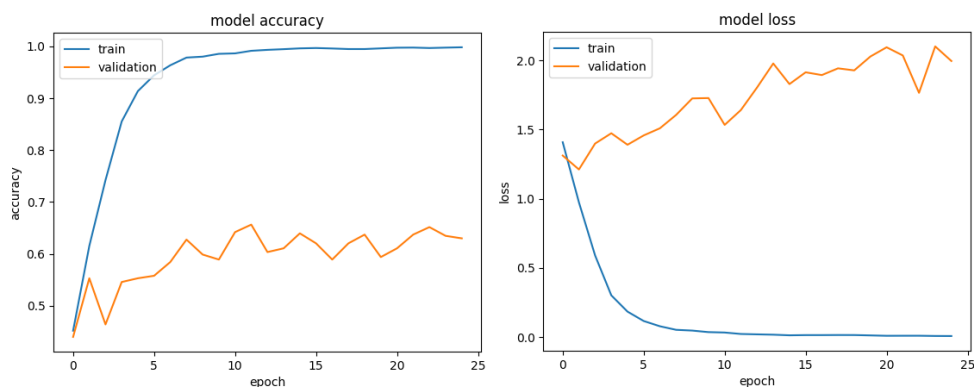
Lr	<i>Data Train</i>				<i>Data Test</i>			
	Akurasi	<i>Presisi</i>	<i>Recall</i>	<i>F1-Score</i>	Akurasi	<i>Presisi</i>	<i>Recall</i>	<i>F1-Score</i>
0.01	0.977	0.954	0.987	0.970	0.630	0.463	0.472	0.458

0.001	0.987	0.987	0.987	0.987	0.778	0.813	0.781	0.782
0.0001	0.991	0.983	0.996	0.989	0.650	0.466	0.490	0.475

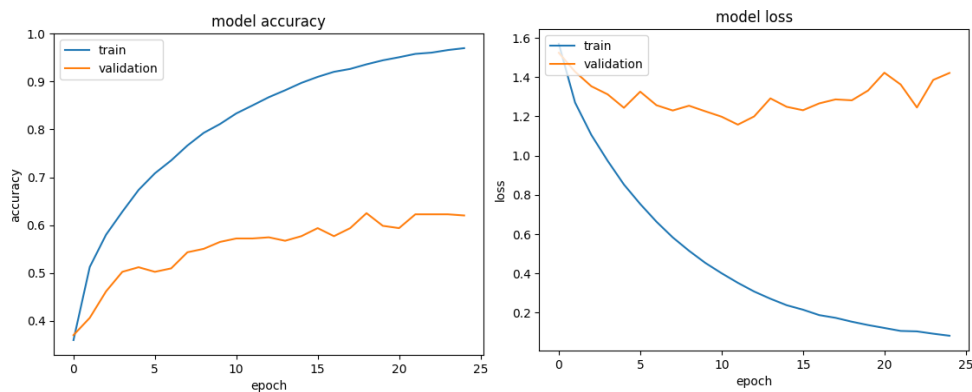
Dari tabel 4. di atas, dapat dilihat bahwa model dengan *learning rate* 0.001 memiliki keseimbangan yang lebih baik antara akurasi pelatihan dan akurasi pengujian dibandingkan dengan model dengan *learning rate* 0.01 dan 0.0001. Meskipun model dengan *learning rate* 0.0001 memiliki akurasi pelatihan tertinggi, performanya pada data pengujian tidak sebaik model dengan *learning rate* 0.001. Model dengan *learning rate* 0.01 menunjukkan performa yang kurang baik pada data pengujian dengan akurasi yang rendah dan perbedaan yang signifikan antara akurasi pelatihan dan pengujian, mengindikasikan adanya *overfitting*.

Hasil Pengujian Menggunakan Optimizer SGD

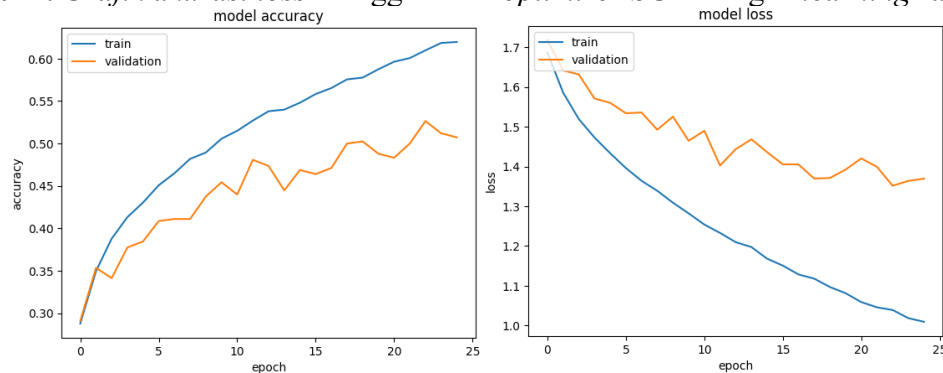
Pengujian ini dilakukan dengan menggunakan *optimizer* SGD dengan *Learning Rate* sebesar 0.01, 0.001, dan 0.001. Hasilnya adalah sebagai berikut:



Gambar 6. Grafik akurasi *loss* menggunakan *optimizer* SGD dengan *learning rate* 0.01



Gambar 7. Grafik akurasi *loss* menggunakan *optimizer* SGD dengan *learning rate* 0.001



Gambar 8. Grafik akurasi *loss* menggunakan *optimizer* SGD dengan *learning rate* 0.0001

Pada ketiga gambar grafik diatas terlihat bahwa *learning rate* 0.01 menunjukkan *overfitting*, di mana akurasi pelatihan cepat mencapai hampir 1, tetapi akurasi validasi berfluktuasi di sekitar 0.6 dan *loss* validasi meningkat setelah beberapa epoch awal. Lalu untuk grafik dengan *learning rate* 0.001 menunjukkan keseimbangan terbaik, dengan akurasi pelatihan dan validasi yang meningkat secara bertahap dan stabil, serta penurunan *loss* yang konsisten, meskipun ada sedikit indikasi *overfitting*. Selanjutnya Grafik dengan *learning rate* 0.0001 menunjukkan pelatihan yang sangat lambat dan kurang optimal, dengan akurasi pelatihan hanya mencapai sekitar 0.6 dan akurasi validasi di sekitar 0.5, tetapi model menunjukkan stabilitas yang lebih baik dan tidak ada tanda-tanda *overfitting* yang signifikan. Adapun hasil perbandingannya berdasarkan hasil *training* setiap model adalah sebagai berikut:

Table 5. Hasil Perbandingan Setiap Model

<i>Learning Rate</i>	<i>Accuracy (%)</i>	<i>Val Accuracy (%)</i>	<i>Loss</i>	<i>Val Loss</i>
0.01	99.80	62.98	0.0063	1.9958
0.001	96.98	62.02	0.0824	1.4205
0.0001	61.97	5-.72	1.0091	1.3695

Pengujian dengan struktur *EfficientNet-B4*

Pengujian ini dilakukan menggunakan Arsitektur *Convolutional Neural Network* (CNN) menggunakan struktur *EfficientNet-B4*. Berikut adalah *source code* untuk arsitektur model yang digunakan:

Tabel 6. Sourcode arsitektur CNN *EfficientNet-B4*

```
def 5461amper5461er_classifier(image_shape=IMG_SIZE,
data_augmentation=data_augmentar()):

    IMG_SHAPE = image_shape + (3,)
    base_model = applications.EfficientNetB4 (input_shape=IMG_SHAPE,
        include_top=False,
        weights='imagenet')
    base_model.trainable = True
    for layer in base_model.layers[0:218]:
        layer.trainable = False

    inputs = Input(shape=IMG_SHAPE)
    x = data_augmentation(inputs)
    x = preprocess_input(inputs)
    x = base_model(x)
    x = GlobalAveragePooling2D()(x)
    outputs = Dense(units=5, activation = "softmax")(x)

    model = Model(inputs, outputs)
    return model
```

Pada tabel 6. merupakan *source code* arsitektur CNN yang menggunakan struktur *EfficientNet-B4*. Model ini menggunakan pendekatan transfer learning dengan memanfaatkan

bobot pre-trained dari ImageNet pada 218 lapisan awal yang dibekukan, sementara lapisan-lapisan akhir dapat di-fine-tune untuk menyesuaikan dengan tugas spesifik. Untuk meningkatkan variasi data dan ketahanan model, augmentasi real-time diterapkan menggunakan rotasi dan *zoom* acak, sejalan penelitian yang dilakukan oleh M. Farhan, (2024). Setelah feature extraction, *Global Average Pooling* digunakan untuk mengurangi dimensi dan mencegah *overfitting*. Terakhir, terdapat *Dense Layer* dengan 5 neuron dan aktivasi *softmax* berfungsi sebagai lapisan *output* untuk klasifikasi multi-kelas penyakit daun singkong. Adapun detail dari arsitektur model yang digunakan yaitu sebagai berikut:

```
Model: "model"
```

Layer (type)	Output Shape	Param #
input_3 (InputLayer)	[(None, 128, 128, 3)]	0
efficientnetb4 (Functional)	(None, 4, 4, 1792)	17673823
global_average_pooling2d (GlobalAveragePooling2D)	(None, 1792)	0
dense (Dense)	(None, 5)	8965

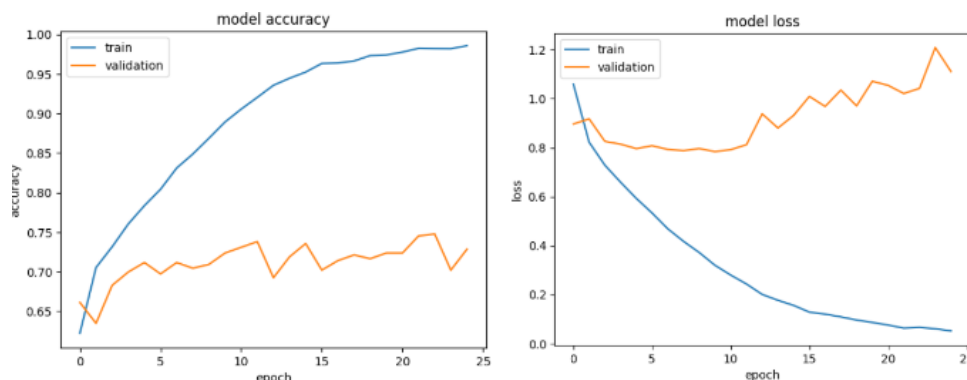
```
Total params: 17682788 (67.45 MB)
Trainable params: 16501629 (62.95 MB)
Non-trainable params: 1181159 (4.51 MB)
```

Gambar 9. arsitektur CNN *EfficientNet-B4*

Gambar 14. merupakan arsitektur yang terdiri dari lapisan *EfficientNet-B4* yang diikuti oleh lapisan Input yang menerima gambar berukuran 128x128 piksel dengan 3 saluran warna (RGB). Lapisan utama adalah *EfficientNet-B4* yang digunakan sebagai *backbone* untuk ekstraksi fitur, menghasilkan *output* dengan ukuran [None, 4, 4, 1792]. Setelah ekstraksi fitur, lapisan *Global Average Pooling* diterapkan untuk mengubah ukuran *output*. Lapisan terakhir adalah *Dense Layer* dengan 5 neuron yang digunakan untuk klasifikasi akhir. Secara keseluruhan, model ini memiliki total 17.682.788 parameter, dengan 16.501.629 di antaranya adalah parameter yang dapat dilatih dan 1.181.159 adalah parameter yang tidak dapat dilatih.

Optimizer SGD

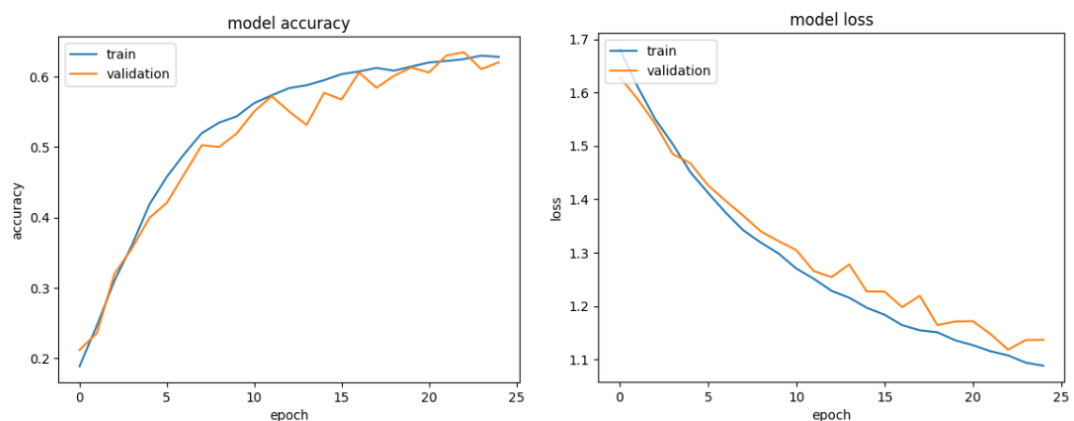
Pengujian ini dilakukan dengan menggunakan *optimizer* SGD dengan *Learning Rate* sebesar 0.01, 0.001, dan 0.0001. Hasilnya adalah sebagai berikut:



Gambar 15. Grafik akurasi *loss* menggunakan *optimizer adam* dengan *learning rate* 0.01



Gambar 16. Grafik akurasi *loss* optimizer SGD dengan *learning rate* 0.001



Gambar 17. Grafik akurasi *loss* menggunakan optimizer adam dengan *learning rate* 0.0001

Pada ketiga gambar grafik diatas terlihat bahwa *learning rate* 0.01 menunjukkan akurasi pelatihan yang sangat cepat meningkat dan mencapai hampir 100%, namun akurasi validasi tidak mengikuti dan bahkan menurun, menunjukkan *overfitting*. Lalu untuk grafik dengan *learning rate* 0.001 model memiliki akurasi pelatihan yang meningkat stabil, namun akurasi validasi tidak mengalami peningkatan signifikan setelah sekitar 15 epoch, menunjukkan kemungkinan *overfitting* ringan. Selanjutnya Grafik dengan *learning rate* 0.0001, baik akurasi pelatihan maupun validasi meningkat secara paralel dengan baik, dan tidak ada indikasi *overfitting*, menunjukkan bahwa lr 0.0001 memberikan hasil yang paling seimbang dan stabil.. Adapun hasil perbandingannya berdasarkan hasil *training* setiap model adalah sebagai berikut:

Table 7. Hasil Perbandingan Setiap Model

<i>Learning Rate</i>	<i>Accuracy (%)</i>	<i>Val Accuracy (%)</i>	<i>Loss</i>	<i>Val Loss</i>
0.01	62.79	62.02	1.0881	1.1368
0.001	74.24	67.55	0.6950	0.8540
0.0001	98.57	72.84	0.0515	1.1107

Selain itu, hasil perbandingan metrik performa pada *data train* dan *data test* untuk setiap model dengan *learning rate* yang berbeda adalah sebagai berikut:

Table 8. Hasil Perbandingan Metrik

Lr	<i>Data Train</i>				<i>Data Test</i>			
	Akurasi	<i>Presisi</i>	<i>Recall</i>	<i>F1-Score</i>	Akurasi	<i>Presisi</i>	<i>Recall</i>	<i>F1-Score</i>
0.01	0.999	0.999	0.999	0.988	0.716	0.518	0.492	0.492
0.001	0.787	0.663	0.603	0.619	0.716	0.518	0.492	0.492
0.0001	0.612	0.345	0.326	0.328	0.625	0.380	0.369	0.349

Dari tabel 8. di atas, dapat dilihat bahwa model dengan *learning rate* 0.01 menunjukkan peningkatan akurasi pelatihan yang sangat cepat dan 5464 amper mencapai 100%, namun akurasi validasi tidak mengikuti dan bahkan menurun, menunjukkan *overfitting* yang parah. Model ini juga memiliki akurasi validasi 62.02% dan *loss* validasi 1.1368, sementara pada *data test* memiliki performa yang buruk dengan akurasi 71.6%, *presisi* 51.8%, *recall* 49.2%, dan *F1-score* 49.2%. Selanjutnya model dengan *learning rate* 0.001 memiliki akurasi pelatihan yang meningkat stabil, namun akurasi validasi tidak mengalami peningkatan signifikan setelah sekitar 15 epoch, menunjukkan kemungkinan *overfitting* ringan. Model ini memiliki nilai akurasi validasi 67.55% dan *loss* validasi 0.8540. sedangkan performa model pada *data test* sedikit lebih baik daripada *learning rate* 0.01 dengan akurasi 71.6%, *presisi* 51.8%, *recall* 49.2%, dan *F1-score* 49.2%.

Sementara itu, model dengan *learning rate* 0.0001 menunjukkan peningkatan yang seimbang antara akurasi pelatihan dan validasi, tanpa indikasi *overfitting*. Model ini memiliki akurasi validasi tertinggi sebesar 72.84% dengan *loss* validasi 1.1107. Meskipun pada *data train*, metrik performanya lebih rendah dibandingkan model lain yaitu dengan akurasi 61.2%, *presisi* 34.5%, *recall* 32.6%, dan *F1-score* 32.8%, namun tetap menunjukkan performa yang lebih seimbang dan stabil antara data pelatihan dan pengujian dengan akurasi test 62.5%, *presisi* 38.0%, *recall* 36.9%, dan *F1-score* 34.9%. Sehingga dapat disimpulkan bahwa model dengan *learning rate* 0.0001 memberikan hasil yang paling seimbang dan stabil di antara ketiga model tersebut, meskipun metrik performanya lebih rendah dibandingkan dengan *learning rate* 0.001, namun tetap lebih baik dalam hal keseimbangan antara data pelatihan dan pengujian (Allaam & Wibowo, 2021).

KESIMPULAN

Implementasi klasifikasi gejala penyakit pada daun singkong berhasil dilakukan menggunakan model *convolutional neural network* (CNN). Model dengan *learning rate* 0.0001 menunjukkan peningkatan yang seimbang antara akurasi pelatihan dan validasi, tanpa indikasi *overfitting*. Model ini memiliki akurasi validasi tertinggi sebesar 72.84% dengan *loss* validasi 1.1107. Meskipun pada *data train*, metrik performanya lebih rendah dibandingkan model lain yaitu dengan akurasi 61.2%, *presisi* 34.5%, *recall* 32.6%, dan *F1-score* 32.8%, namun tetap menunjukkan performa yang lebih seimbang dan stabil antara data pelatihan dan pengujian dengan akurasi test 62.5%, *presisi* 38.0%, *recall* 36.9%, dan *F1-score* 34.9%. Sehingga dapat disimpulkan bahwa model dengan *learning rate* 0.0001 memberikan hasil yang paling seimbang dan stabil di antara ketiga model tersebut, meskipun metrik performanya lebih rendah dibandingkan dengan *learning rate* 0.001, namun tetap lebih baik dalam hal keseimbangan antara data pelatihan dan pengujian

DAFTAR PUSTAKA

- Abel, S., Pranidana, A. M., Qasos, L., & Ula, M. (2024). Perbandingan Akurasi Metode Convolutional Neural Network (CNN) dan Sobel untuk Klasifikasi Buah Rambutan melalui Pengolahan Citra. *Prosiding Seminar Nasional Teknologi Dan Teknik Informatika (SENASTIKA)*, 1(1).
- Allaam, M. R. R., & Wibowo, A. T. (2021). Klasifikasi Genus Tanaman Anggrek Menggunakan Metode Convolutional Neural Network (CNN). *EProceedings of Engineering*, 8(2).
- Dewiani, D., Hutabarat, M. Y., Marwan, M., Miranda, D., Solin, S. R. B., Muzamil, T., Zuriansyah, Z., Fuqara, F. A., Akmal, A. K., & Pamungkas, I. (2024). Sosialisasi Nilai Tambah Singkong di Desa Keras. *Teknodimas: Teknologi Pengabdian Masyarakat*, 2(1), 70–76.
- Farhan, M. (2024). *Analisis perbandingan pengaruh variasi data augmentasi terhadap kinerja mobilenetv2 dalam klasifikasi penyakit daun teh*. Fakultas Sains dan Teknologi UIN Syarif Hidayatullah Jakarta.
- Gunawan, I. P., Gunawan, A. N. S., Ratnadewi, R., Safrizal, S., Sembiring, A. A., Husain, N. P., & Kuntarto, G. P. (2023). *Pemrosesan citra*. PT Mafy Media Literasi Indonesia.
- Hartati, S. (2021). *Kecerdasan Buatan Berbasis Pengetahuan*. Ugm Press.
- Ibrahim, M. B., Sari, F. P., Kharisma, L. P. I., Kertati, I., Artawan, P., Sudipa, I. G. I., Simanihuruk, P., Rusmayadi, G., Nursanty, E., & Lolang, E. (2023). *Metode Penelitian Berbagai Bidang Keilmuan (Panduan & Referensi)*. PT. Sonpedia Publishing Indonesia.
- Masita, F. (2023). *Pengaruh Kombinasi Populasi Dari Tumpangsari Singkong-Kedelai Pada Hasil Dan Vigor Awal Benih Kedelai (Glycine max L. Merrill)*.
- Predianto, E., & Sutomo, B. (2024). Klasifikasi Jenis Bunga Dengan Algoritma Convolutional Neural Network (CNN) Menggunakan Metode Region-Based Convolutional Neural Network (R-CNN). *Cyberspace: Jurnal Pendidikan Teknologi Informasi*, 8(2), 1–15.
- Putri, N. A., Siregar, M., Perwitasari, I. D., & Mayasari, S. (2024). *Aplikasi Diagnosa Penyakit Dan Hama Pohon Aren (Arenga Pinnata) Dengan Metode Certainty Factor*. Serasi Media Teknologi.
- Ramadhan, M. A. (2020). Penerapan Farm From Home Melalui Kegiatan Vertikultur Sebagai Solusi Antisipatif Terhadap Krisis Ketahanan Pangan Akibat Pandemi Covid-19. *Minda Mahasiswa Indonesia: Antisipasi Resesi Dan Krisis Pangan Akibat Pandemi*, 55.
- Salim, E. (2024). *Mengolah Singkong Menjadi Tepung Mocaf, Bisnis Produk Alternatif Pengganti Terigu*. Penerbit Andi.
- Santoso, I., & Madiistriyatno, H. (2021). *Metodologi penelitian kuantitatif*. Indigo Media.
- Saparinto, C., & Susiana, R. (2024). *Grow Your Own Kitchen Spice—Panduan Praktis Menanam 28 Tanaman Bumbu Dapur Populer di Pekarangan*. Penerbit Andi.
- Yusuf, M., Kurniawan, D., & Agustin, T. (2024). Klasifikasi Penyakit Tanaman Jagung dengan Kecerdasan Buatan Berbasis CNN. *Prosiding Seminar Nasional Amikom Surakarta*, 2, 355–368.



This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.