

Fine Tuning Model Convolutional Neural Network EfficientNet-B4 dengan Augmentasi Data untuk Klasifikasi Penyakit Kakao

by De Rosal Ignatius Moses Setiadi

Submission date: 20-Feb-2024 04:19PM (UTC+0700)

Submission ID: 2207911242

File name: JISAD_-_Template.docx (2.91M)

Word count: 4703

Character count: 30601

Fine Tuning Model Convolutional Neural Network EfficientNet-B4 dengan Augmentasi Data untuk Klasifikasi Penyakit Kakao

*Fine Tuning Convolutional Neural Network Model EfficientNet-B4 with Data
Augmentation for Cocoa Disease Classification*

Akbar C²³ang Pradana¹, De Rosal Ignatius Moses Setiadi¹, Ahmad Rofiqul Musli²

¹Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro,
Semarang, Jawa Tengah 50131 Indonesia

²Fakultas Teknologi Informasi Universitas Merdeka, Malang, Jawa Timur 65146, Indonesia

E-mail: *rofickachmad@unmer.ac.id

Abstract. Cocoa represents a significant agricultural commodity in Indonesia, contributing 15.68% to the economy. Diseases such as Black Pod Rot and Pod Borer pose substantial losses to farmers, necessitating the development of a high-performance automatic classification model. However, the challenge of an unbalanced dataset can impair recognition performance. This study introduces a disease recognition model for cocoa images employing the EfficientNet-B4 Convolutional Neural Network (CNN) model, enhanced through fine-tuning and incorporating seven types of data augmentation. The proposed CNN model achieved a notable accuracy of 97.3%, marking an improvement of approximately 7.4% over the baseline model with relatively few epochs. Furthermore, a comparative analysis with other CNN architectures, including Xception, InceptionV3, ResNet, DenseNet, and other EfficientNet versions, was conducted using a consistent approach of fine-tuning and epochs. The findings affirm the proposed model's superiority, underscoring its efficacy in managing unbalanced data.

Keywords: Convolutional Neural Networks; EfficientNet-B4; fine tuning; Image Classification; Image Recognition

Abstrak. Kakao merupakan komoditas pertanian penting di Indonesia, yang berkontribusi sebesar 15,68% terhadap ekonomi. Penyakit seperti Busuk Buah Hitam dan Penggerek Buah menimbulkan kerugian besar bagi petani, sehingga diperlukan pengembangan model klasifikasi otomatis berkinerja tinggi. Namun, tantangan dataset yang tidak seimbang dapat merusak kinerja pengenalan. Studi ini memperkenalkan model pengenalan penyakit untuk gambar kakao dengan menggunakan model Convolutional Neural Network (CNN) EfficientNet-B4, yang ditingkatkan melalui fine-tuning dan menggabungkan tujuh jenis augmentasi data. Model CNN yang diusulkan mencapai akurasi mencolok sebesar 97,3%, menandai peningkatan sekitar 7,4% dibandingkan model dasar dengan jumlah epoch yang relatif sedikit. Selain itu, analisis komparatif dengan arsitektur CNN lainnya, termasuk Xception, InceptionV3, ResNet, DenseNet, dan versi EfficientNet lainnya, dilakukan menggunakan pendekatan fine-tuning dan epoch yang konsisten. Temuan-temuan ini mengukuhkan superioritas model yang diusulkan, menekankan efikasinya dalam mengelola data yang tidak seimbang.

Kata kunci: Convolutional Neural Network; EfficientNet-B4; Fine tuning; Klasifikasi Citra; Recognisi Citra

10
How to Cite:

14
Submitted: 00-00-0000 | Accepted: 00-00-0000 | Published: 00-00-0000



This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.
DOI:

PENDAHULUAN

Industri kakao dan cokelat secara global senilai 44 miliar AS pada tahun 2019. Penyakit Pod Borer yang disebabkan oleh *Conopomorpha cramerella* menjadi hama utama cacao di daerah Malaysia dan Indonesia. Penyakit ini telah terdeteksi sejak 1980-an. Sayangnya pengendalian saat ini tidak terlalu efisien[1]. Selain penyakit Pod borer, juga ada penyakit Black Pod Rot yang disebabkan oleh *Phytophthora megakarya*. Penyakit ini menjangkit tanaman kakao di Kamerun. *Phytophthora megakarya* adalah spesies jamur yang merupakan penyebab utama dari penyakit ini. Pada skala global, penyakit ini mengakibatkan kerugian sebesar 20-30%[2]. Penyakit Black Pod Rot dapat berkembang pesat dalam jangka waktu satu hingga dua minggu. Saat ini sudah banyak algoritma machine learning yang digunakan untuk klasifikasi dan berkembang menjadi algoritma deep learning. Selain itu, banyak model algoritma yang sudah dilatih sehingga siap digunakan seperti DenseNet, EfficientNet, InceptionV3, MobileNet, ResNet, dan Xception[3].

Penelitian yang dilakukan oleh[4] yaitu mengusulkan jaringan neural yang merupakan gabungan dari jaringan Xception dan ResNet50V2. Pada dataset yang terbagi menjadi tiga kelas pada penyakit Covid-19, Penyakit Pneumonia, dan Sehat didapatkan hasil akurasi rata-rata keseluruhan kelas terbilang baik yaitu 91,4%. Pada penelitian serupa juga dilakukan oleh [5] yaitu mengusulkan sistem deteksi otomatis untuk mencegah penyebaran Covid-19 secara cepat. Menggunakan tiga model yang berbeda yaitu DenseNet, InceptionV3, dan Inception-ResNetV4 yang akan digunakan untuk menganalisis pasien yang terinfeksi. Pada penelitian ini DenseNet mendapatkan hasil akurasi terbaik yaitu 92%, disusul model Inception-ResNetV4 dengan akurasi sebesar 85,57% dan terakhir yaitu InceptionV3 dengan akurasi sebesar 83,47%. Penelitian yang dilakukan oleh[6] menggunakan model InceptionV3 untuk mengklasifikasikan enam kategori olahraga. Model InceptionV3, berhasil mengklasifikasikan dengan rata-rata akurasi sebesar 96,64%. Identifikasi dan klasifikasi penyakit tanaman rumah kaca oleh[7] menggunakan model EfficientNet-B4 dengan optimizer Ranger. Hasil dari mengklasifikasikan empat jenis penyakit daun mentimun, dengan akurasi model EfficientNet-B4 dengan optimizer Ranger yaitu sebesar 96%. Penelitian menggunakan model EfficientNet juga dilakukan oleh[8] untuk mengklasifikasikan kanker kulit. Penelitian ini membandingkan semua model EfficientNet dari EfficientNet-B0 sampai EfficientNet-B7. Hasil tertinggi pada penelitian ini yaitu model EfficientNet-B4 dengan akurasi sebesar 87,91% dimana performa terbaik diraih oleh EfficientNet-B4 dan EfficientNet-B5.

Dari beberapa penelitian yang telah dilakukan didapatkan kesimpulan bahwa model pre-trained seperti DenseNet, ResNet, Xception, Inception, dan EfficientNet sangat baik digunakan untuk klasifikasi dan memiliki akurasi yang tinggi. Dari beberapa penelitian diatas, model EfficientNet-B4 merupakan salah satu algoritma terbaik untuk klasifikasi meskipun hasil akurasinya sebesar 87,91%, sehingga peneliti berharap bahwa model EfficientNet-B4 masih dapat ditingkatkan lagi. Banyak hal yang menyebabkan akurasi dapat menurun seperti objek yang memiliki kemiripan tinggi. Untuk meningkatkan nilai akurasinya, dapat dilakukan Fine-tuning dan penambahan objek secara semu menggunakan augmentasi data. Menurut [9] augmentasi digunakan supaya data yang ada meningkat secara artifisial. Penelitian yang dilakukan oleh [10] pada model EfficientNet-B4 secara konsisten mengurangi FLOPs dan parameter berdasarkan urutan besarnya daripada ConvNet. Hasil akurasi pada klasifikasi ImageNet dengan model EfficientNet-B4 mencapai 82,9% dengan parameter 19M dan FLOPs 4,2B. Hal yang dapat mempengaruhi tinggi rendahnya akurasi yaitu dataset yang digunakan, fine-tuning yang tidak tepat, dan penggunaan layer tambahan. Dari literature-literatur diatas maka penelitian ini dilakukan dengan tujuan memberikan kontribusi signifikan terhadap deteksi dan klasifikasi penyakit pada tanaman kakao menggunakan teknologi machine learning dan deep learning. Melalui pengembangan model Convolutional Neural Network (CNN) EfficientNet-B4 yang dioptimalkan dengan teknik fine-tuning dan augmentasi data, penelitian ini berhasil mencapai peningkatan akurasi yang signifikan dalam klasifikasi penyakit kakao. Kontribusi utama dari penelitian ini termasuk:

1. Menerapkan Teknik Fine-Tuning dan Augmentasi Data pada model EfficientNet-B4 dan menggunakan tujuh jenis augmentasi data, penelitian ini mengatasi tantangan dataset yang tidak seimbang, yang seringkali menghambat kinerja model klasifikasi.
2. Melakukan analisis komparatif dengan model-model CNN lain seperti Xception, InceptionV3, ResNet, DenseNet, dan versi EfficientNet lainnya, menegaskan superioritas model yang diusulkan dalam mengelola data yang tidak seimbang.

LITERATURE REVIEW

A. Convolutional Neural Network (CNN)

Convolutional Neural Networks (CNN) adalah salah satu kategori dari jaringan saraf yang paling populer terutama untuk data berupa gambar dan video. Perbedaan utama dari CNN dengan jaringan saraf standar yaitu terdapat pada setiap unit pada lapisan CNN adalah filter dua dimensi atau lebih tinggi yang digabungkan dengan lapisan input[11]. Arsitektur neural network CNN terinspirasi dari sistem visual manusia, khususnya cara manusia memproses dan memahami gambar. Pengembangan arsitektur CNN terus berlanjut, dan teknik terbaru seperti transfer learning juga digunakan untuk meningkatkan kinerja [12], [13]. Transfer learning yaitu sebuah metode dimana model akan diberikan masalah baru tetapi menerapkan pengalaman dari pelatihan sebelumnya yang telah dilakukan[14]. Jadi dapat didefinisikan bahwa transfer learning merupakan sebuah metode pada machine learning dengan menggunakan domain asal untuk meningkatkan kualitas model pada domain target. Dengan begitu metode transfer learning yaitu memanfaatkan pengetahuan pada model yang telah dilatih sebelumnya pada data yang besar dan dapat digunakan kembali pada data yang lebih kecil supaya lebih efisien[15]. Model transfer learning yang telah dilatih biasa disebut pre-trained model. Model ini telah dilatih pada dataset besar seperti ImageNet, sehingga hasil dari metode ini sudah sangat baik[16].

B. EfficientNet-B4

Algoritma EfficientNet diperkenalkan oleh[10], memiliki beberapa model dari B0 hingga B7 dan dianggap sebagai salah satu model deep learning yang paling efisien secara komputasi yang dilatih dengan ImageNet[17]. EfficientNet didasarkan pada penskalaan majemuk yang cepat memperluas ukuran model jaringan dasar untuk menargetkan ukuran model dengan cara efisien dan memperoleh akurasi model teratas. penskalaan majemuk memungkinkan jaringan untuk diskalakan secara seragam di seluruh lebar, kedalaman, dan resolusi. Model EfficientNet terdiri dari berbagai jenis mobile inverted bottleneck convolution block (MBConv) dengan berbagai ukuran kernel 3x3 dan 5x5[18]. Metode penskalaan gabungan yang menggunakan koefisien gabungan untuk menskalakan lebar, kedalaman, dan resolusi jaringan secara seragam dengan berikut ini[19].

$$\begin{aligned}d &= \alpha^\phi \\w &= \beta^\phi \\r &= \gamma^\phi\end{aligned}\tag{1}$$

Sehingga $\alpha \cdot \beta^\phi \cdot \gamma^\phi \approx 2$, $\alpha \geq 1, \beta \geq 1, \gamma \geq 1$

Dimana d untuk kedalaman, w untuk lebar, dan r untuk resolusi. Model EfficientNet-B0 dibangun dengan menggunakan nilai $\phi = 0, w = 1, d = 1, r = 1$ yang mewakili model baseline. EfficientNet-B0 terdiri dari lapisan MBconv1 dan MBconv6. Demikian juga, model EfficientNet-B4 dibangun dengan menggunakan nilai $\phi = 4, w = \alpha^4, d = \beta^4, r = \gamma^4$, yang menunjukkan bahwa lebih banyak sumber daya tersedia untuk memperoleh kinerja superlatif. Model EfficientNet-B4 terdiri dari jaringan yang lebih dalam dibandingkan dengan model dasar, yang memahami fitur yang rumit dan lebih kaya serta menggeneralisasi dengan baik pada tugas baru. EfficientNet-B4 terdiri dari jaringan yang lebih luas yang dapat mengekstraksi fitur dan pola optimal yang bermanfaat untuk tugas klasifikasi. Seiring dengan peningkatan akurasi, model EfficientNetB3 juga meningkatkan efisiensi dengan menurunkan FLOPS (Floating Point Operations per Second) dan parameter.

EfficientNet-B4 bisa dikatakan cukup bagus karena memiliki nilai akurasi yang tinggi dengan

Judul Artikel

Nama Penulis

ukuran model 36% lebih kecil dari EfficientNet-B5. Model ini juga sudah dievaluasi pada dataset transfer learning yang umum digunakan. Hasil dari evaluasi tersebut menunjukkan bahwa akurasi yang didapatkan lebih baik dengan pengurangan parameter rata-rata 7,4x. EfficientNet konsisten mendapatkan akurasi yang lebih baik dengan urutan parameter yang lebih sedikit daripada model lainnya seperti ResNet, DenseNet, Inception, dan NASNet [10].

C. Augmentasi Data

Pelatihan jaringan saraf dengan jumlah data yang sedikit menghasilkan akurasi model klasifikasi yang cukup buruk, sehingga banyaknya data yang digunakan sangat mempengaruhi tingkat akurasi jaringan saraf. Meskipun model pre-trained sudah dilatih pada kumpulan data yang besar seperti ImageNet dan memiliki akurasi yang cukup baik, akurasi tersebut masih dapat ditingkatkan lagi. Terdapat salah satu teknik yang dapat membantu untuk meningkatkan akurasi jaringan saraf dengan cara menambah data menjadi lebih banyak yang disebut augmentasi. Augmentasi adalah pendekatan untuk meningkatkan jumlah salinan data dengan sedikit modifikasi tanpa mengumpulkan data baru. Teknik ini adalah salah satu teknik yang paling umum digunakan untuk meningkatkan kinerja model jaringan saraf[20].

Augmentasi pada data dapat diartikan meningkatnya jumlah atau ukuran data secara artifisial/semu. Dengan menggunakan augmentasi data yang ada akan diperbanyak seperti dipotong, diputar/rotasi, dibalik/flip, memainkan warna, dan masih banyak lagi. Augmentasi dilakukan dengan cepat pada saat gambar dimuat. Pada setiap penelitian, augmentasi digunakan untuk memperbanyak data untuk pelatihan[9]. Teknik augmentasi data gambar secara kasar dapat dibagi menjadi tiga kategori, termasuk metode pemrosesan gambar dasar, metode pembelajaran mendalam dan metode optik. Metode pemrosesan gambar dasar meliputi augmentasi data geometris, transformasi warna, injeksi derau, filter kernel, pencampuran gambar, dan penghapusan acak. Tipe augmentasi yang sering digunakan yaitu Augmentasi data geometris dimana tipe ini menggunakan transformasi geometris dari gambar input, termasuk flipping, cropping, rotasi, translasi, deformasi, dan penskalaan[21]. Contoh augmentasi dapat dilihat pada gambar dibawah.

D. Fine Tuning

Fine tuning adalah teknik yang digunakan dalam machine learning untuk memperbaiki kinerja model. Fine tuning memperbaiki kinerja model dengan cara menyesuaikan kembali bobot dan bias dari model yang telah dilatih sebelumnya pada tugas baru. Teknik ini biasanya digunakan pada model yang dilatih pada dataset besar untuk kemudian digunakan pada tugas pengenalan citra atau teks yang lebih kecil[22].

Cara kerja fine-tuning dimulai dengan mengambil model yang sudah dilatih pada dataset besar sebagai model dasar atau pre-trained model. Kemudian perubahan dilakukan dengan mengganti lapisan terakhir dari model tersebut untuk diadaptasi pada tugas yang baru. Lapisan terakhir ini disebut lapisan kelas atau lapisan keluaran. Proses fine-tuning dilakukan dengan melatih ulang lapisan terakhir sambil menjaga agar lapisan lain tetap menggunakan bobot dan bias yang sudah diatur sebelumnya. Dalam proses fine-tuning ini, bobot dan bias pada lapisan terakhir disesuaikan dengan dataset target. Selain itu, fine-tuning juga dapat dilakukan pada beberapa lapisan terakhir atau seluruh lapisan terakhir dari model dasar tergantung pada kompleksitas dataset target dan model yang digunakan[22].

Dense layer adalah jenis lapisan neural network yang terhubung secara padat dengan operasi yang dilakukan oleh dense layer adalah $\text{output} = \text{activation}(\text{dot}(\text{input}, \text{kernel}) + \text{bias})$ [23]. Pada jenis Dense layer tertentu, digunakan fungsi aktivasi ReLU. Properti ReLU adalah membuang semua nilai negatif dan memberikan nol. Namun, parameter default dapat dimodifikasi selain nol[24]. Fine-tuning dengan menambahkan dense layer dilapisan akhir dapat mengatasi masalah gradien yang hilang, selain itu dense berpengaruh dalam kecepatan dan kemudahan pada pelatihan model.

Selain menambahkan dense juga terdapat pooling layer. Pooling layer memiliki fungsi untuk mengurangi ukuran agar biaya komputasi yang dibutuhkan lebih sedikit untuk pemrosesan data

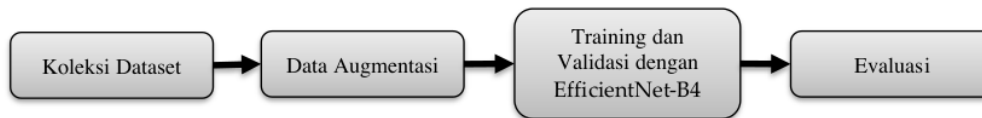
dengan cara downsampling dimana layer yang digunakan yaitu Global Average Pooling[25]. Global Average Pooling sama halnya seperti Pooling layer biasanya, perbedaanya terdapat pada penyatuan yang ditetapkan ke semua nilai saluran fitur dan digabungkan menjadi satu nilai[26]. Layer ini akan digunakan sebagai representasi akhir dari model pengenalan gambar karena hanya menghasilkan satu nilai per kanal fitur. Hal ini memungkinkan mengurangi dimensi sehingga meningkatkan kinerja model dengan mengurangi masalah over-fitting[20], [27].

Untuk melatih jaringan saraf agar lebih cepat dan mencegah terjadinya over-fitting maka perlu ditambahkan dropout. Dengan menggunakan dropout layer, menjadi lebih mudah bagi model selama tahap pengujian untuk memperkirakan hasil rata-rata dari semua prediksi jaringan yang ditipiskan ini dengan memanfaatkan jaringan yang tidak ditipiskan yang memiliki bobot lebih kecil[20]. Memilih 0,2 nilai probabilitas p dropout dalam model yang diusulkan diharapkan mencapai akurasi yang lebih tinggi dan mengurangi masalah over-fitting dari jaringan saraf yang diusulkan.

Tahap terakhir yaitu menggabungkan semua vektor fitur yang telah dibentuk dari tahap sebelumnya. Dari vektor fitur yang telah diterima dari lapisan dropout diteruskan lebih jauh kedalam jaringan untuk klasifikasi sebagai lapisan keluaran menggunakan fully connected layer. Dengan lapisan Softmax sebagai lapisan keluaran untuk pengklasifikasi model berdasarkan jumlah kelas pada dataset yang dibangun menggunakan Dense layer dan menggabungkan input dengan output model[20]. Pada pelatihan model peneliti juga menambahkan optimizer yang berfungsi untuk mengatasi kelemahan dari penurunan gradien. Optimizer yang digunakan yaitu Adam[28].

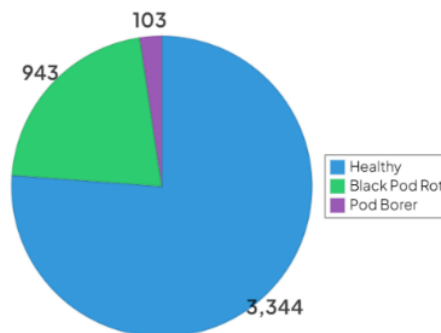
METODE YANG DIUSULKAN

Penelitian ini mengusulkan augmentasi data dan model EfficientNet-B4 dengan fine-tuning dan penambahan beberapa lapisan layer terakhir. Secara umum tahapan metode digambarkan pada Gambar 1.



Gambar 1. Tahapan Metode

Jenis data bersifat kualitatif dengan type gambar adalah JPG. Sumber didapatkan dari dataset umum Kaggle (<https://www.kaggle.com/datasets/zaldyjr/cacao-diseases>). Pada dataset terdapat tiga kelas, dimana kelas black pod rot terdapat 943 gambar, pod borer 103 gambar, dan healthy 3344 gambar sehingga keseluruhan berjumlah 4.390 gambar, lihat Gambar 2. Tiap kelas akan diambil 5% gambar untuk digunakan pada pengujian.



Gambar 2. Grafik Jumlah Kelas dan Baris pada Dataset

Model augmentasi data yang diusulkan adalah rotasi, pergeseran lebar, pergeseran tinggi, shear, zoom, flip horizontal dan vertical. Sedangkan model CNN yang diusulkan dipresentasikan pada Tabel 1.

Tabel 1. Model CNN yang diusulkan

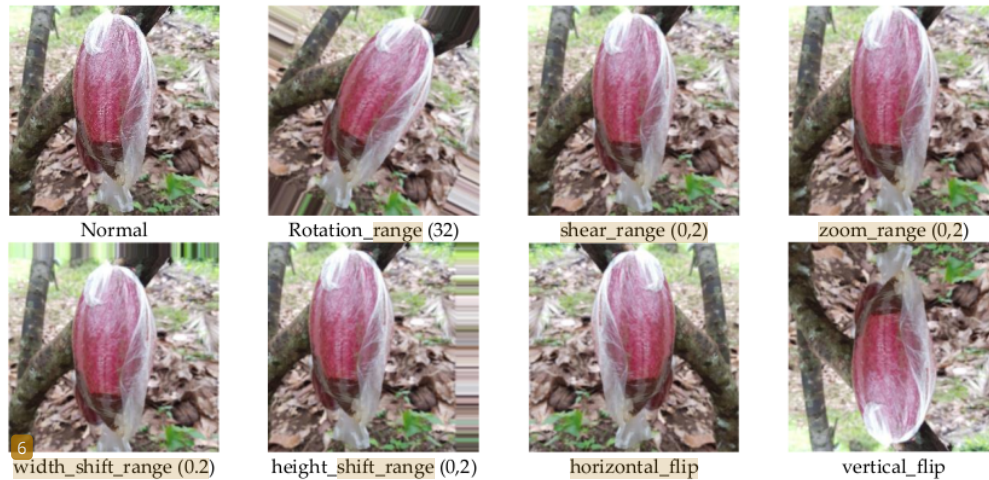
Layer	Output Shape
input_1 (InputLayer)	(380, 380, 3)
rescaling (Rescaling)	(380, 380, 3)
normalization (Normalization)	(380, 380, 3)
7f.math.truediv (TFOpLambda)	(380, 380, 3)
stem_conv_pad (ZeroPadding2D)	(381, 381, 3)
stem_conv (Conv2D)	(190, 190, 48)
stem_bn (BatchNormalization)	(190, 190, 48)
stem_activation (Activation)	(190, 190, 48)
block1a_dwconv (DepthwiseConv2D)	(190, 190, 48)
block1a_bn (BatchNormalization)	(190, 190, 48)
...	...
2 block7b_se_reduce (Conv2D)	(1, 1, 112)
block7b_se_expand (Conv2D)	(1, 1, 2688)
block7b_se_excite (Multiply)	(12, 12, 2688)
block7b_project_conv (Conv2D)	(12, 12, 448)
block7b_2project_bn (BatchNormalization)	(12, 12, 448)
block7b_dropout (Dropout)	(12, 12, 448)
block7b_add (Add)	(12, 12, 448)
top_conv (Conv2D)	(12, 12, 1792)
top_bn (BatchNormalization)	(12, 12, 1792)
17_activation (Activation)	(12, 12, 1792)
global_average_pooling2d (GlobalAveragePooling2D)	(1792)
dropout (Dropout)	(1792)
dense (Dense)	(32)
dense_1 (Dense)	(3)
Total params: 17,731,298	
Trainable params: 16,200,223	
Non-trainable params: 1,531,075	
Non-trainable params: 1,531,075	

Secara default, EfficientNet-B4 memiliki 478 layer, dimana tiga layer terakhir adalah classifier. Tetapi karena ketiga layer ini merupakan classifier yang dapat digunakan untuk mengklasifikasikan 1000 kelas maka tentunya akan mengurangi akurasi jika difokuskan untuk klasifikasi tiga kelas saja. Jumlah layer pada model yang diusulkan yaitu 479 layer termasuk empat layer tambahan. Layer tambahan ditandai dengan warna merah pada tabel tersebut. Layer tambahan yaitu tiga layer model dan satu layer fully connected. Tiga layer tambahan yang diberikan yaitu GlobalAveragePooling2D, Dropout, dan Dense layer serta layer fully connected menggunakan Dense. Alasan dari menambahkan beberapa layer tersebut karena setiap layer yang ditambahkan memiliki keunggulan yang cukup mempengaruhi pelatihan model. Pada GlobalAveragePooling2D digunakan untuk mengambil nilai rata-rata dari citra, Dense dengan activation ReLU untuk membuat pembatas pada bilangan nol sehingga dapat mempercepat proses yang dilakukan, dan Dropout untuk mencegah terjadinya overfitting. Lalu fully connected layer yang ditambahkan yaitu Dense dengan activation Softmax untuk mengubah jumlah kelas awal menjadi jumlah kelas yang sesuai dengan dataset. Selain itu, pada model ini hanya sebagian dari layer yang ditraining dimana layer ke-1 hingga ke-238 trainable=False dan layer ke-239 hingga terakhir trainable=True. Sehingga parameter yang dilatih berjumlah 16,200,223 dari total parameter 17,731,298.

HASIL DAN PEMBAHASAN

A. Augmentasi Data

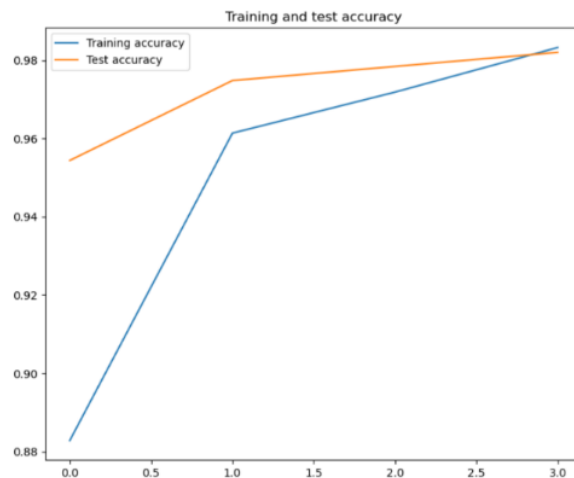
Augmentasi data pada klasifikasi citra bertujuan untuk meningkatkan kinerja model dengan menciptakan variasi baru dari dataset, memperluas jumlah sampel pelatihan, mengatasi variasi domain, dan mengurangi risiko overfitting. Hal ini membantu model mengenali objek dengan lebih baik, meningkatkan keandalan pada data baru, dan mengoptimalkan hasil klasifikasi citra[29], [30]. Dimana contoh hasil augmentasi disajikan pada Gambar 3.



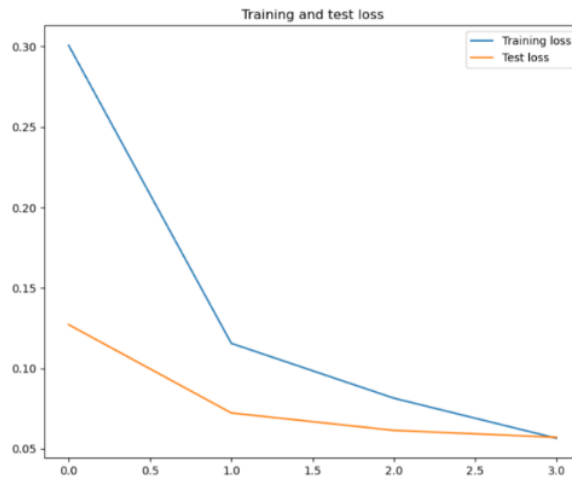
Gambar 3. Contoh Hasil Augmentasi Data

B. Tes Pelatihan

Sebelum melakukan pelatihan model peneliti menetapkan callback sebesar 0.98 atau 98% yang artinya proses pelatihan akan berhenti jika akurasi training dan validation keduanya sama diangka 98%. sehingga pada proses pelatihan model didapatkan data sebagai berikut.

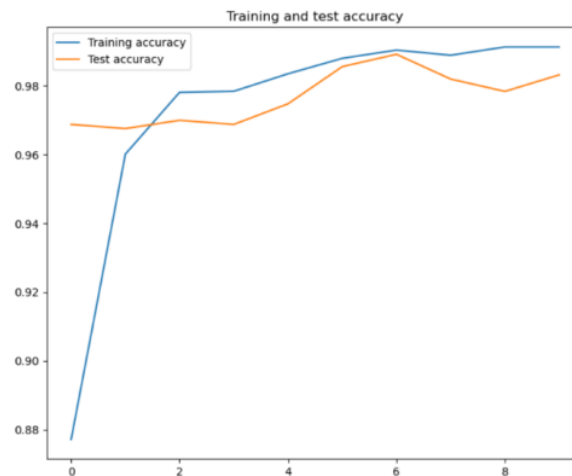


Gambar 4. Grafik Akurasi dari Model yang Diusulkan pada minimal epoch

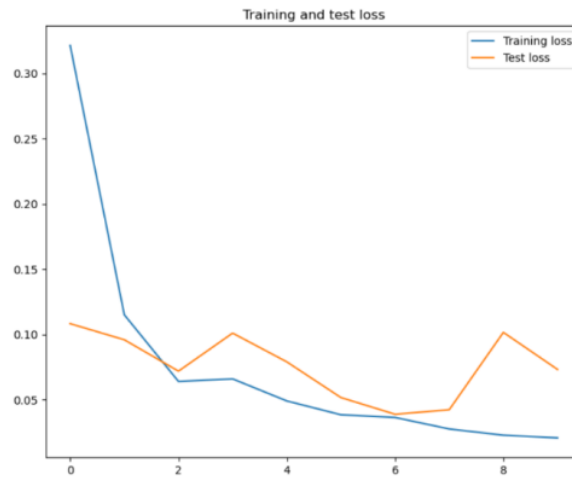


Gambar 5. Grafik Loss dari Model yang Diusulkan pada Minimal Epoch

Melihat dari grafik pada Gambar 4 dan Gambar 5, diketahui bahwa akurasi pada epoch pertama memiliki selisih yang cukup jauh yaitu hampir 7%. Tetapi pada epoch kedua meningkat drastis, hingga pada epoch keempat antara training dan validation memiliki nilai diatas 98% sehingga pelatihan model secara otomatis berhenti. Pada grafik loss juga mirip dengan garafik akurasi dimana epoch pertama terdapat selisih loss yang cukup tinggi. Tetapi pada epoch kedua nilai loss pada training dapat turun secara drastis mendekati nilai loss validation pada epoch pertama. Hingga pada epoch ke empat nilainya cukup kecil yaitu 0.0565 untuk loss training dan 0.0571 untuk loss validation. Hasil ini sebenarnya telah menunjukkan bahwa performa model CNN yang dihasilkan sangat memuaskan bahkan dengan jumlah epoch yang sangat minim, yaitu empat. Tetapi ketika dibandingkan model-model lain pembeding yang dibahas pada bagian VI. E tampak adanya gap yang sangat tinggi ketika dibandingkan dengan jumlah epoch yang sama, maka dicoba pengujian menggunakan 10 epoch tanpa pembatasan, dan ternyata hasilnya juga sudah relative stabil, seperti yang ditunjukkan pada Gambar 6 dan Gambar 7.



Gambar 6. Grafik Akurasi dari Model yang Diusulkan pada minimal epoch



Gambar 7. Grafik Loss dari Model yang Diusulkan pada Minimal Epoch

C. Pengujian

Metode Confusion Matrix juga digunakan untuk menghitung akurasi pada konsep data mining. Confusion Matrix didefinisikan melalui tabel yang menunjukkan jumlah data uji yang diklasifikasikan dengan benar dan jumlah data uji yang diklasifikasikan dengan salah. Model yang diuji akan dihitung berdasarkan recall, F1-score, precision dan accuracy. Metrik dihitung pada kasus True Positive (TP), False Positive (FP), True Negative (TN), dan False Negative (FN) [19], [31].

True Positive (TP) label benar positif dan label diprediksi positif.

False Positive (FP) label benar negatif dan label diprediksi positif.

True Negative (TN) label benar negatif dan label diprediksi negatif.

False Negative (FN) label benar positif dan label diprediksi negatif.

Akurasi merupakan metrik evaluasi yang menemukan kinerja model di semua kelas. Ini adalah sebagian kecil dari jumlah prediksi yang benar dengan jumlah total prediksi, yang dapat dihitung dengan Formula (2).

$$Accuracy = \frac{(TN + TP)}{(TN + TP + FN + FP)} \times 100\% \quad (2)$$

Presisi adalah metrik evaluasi yang menghitung fraksi dari total sampel positif terhadap jumlah total sampel positif baik yang diklasifikasikan secara tepat maupun tidak tepat, yang dapat dihitung dengan Formula (3).

$$Precision = \frac{TP}{(TP + FP)} \times 100\% \quad (3)$$

Recall: metrik evaluasi yang menghitung fraksi dari total sampel positif terhadap jumlah total sampel input positif yang diklasifikasikan dengan benar, yang dapat dihitung dengan Formula (4).

$$Recall = \frac{TP}{(TP + FN)} \times 100\% \quad (4)$$

F1-score: metrik evaluasi yang menemukan akurasi mode dengan menggabungkan presisi dan daya ingat dengan memberi lebih banyak bobot pada positif palsu dan negatif palsu, yang dapat dihitung dengan Formula (5).

$$F1\ Score = \frac{2 \times (Recall \times Precision)}{(Recall + Precision)} \quad (5)$$

Dengan menggunakan confusion matrix diatas model diuji dengan 5% gambar yang telah

Judul Artikel

Nama Penulis

dipisahkan diawal. 5% data tersebut meliputi pod borer (5), black pod rot (47), dan healthy (167). Hasil pengujian dinyatakan dalam bentuk tabel confusion matrix. Setelah hasil pengujian sudah dalam bentuk tabel confusion matrix, dan akan dilakukan pengukuran dengan rumus Accuracy, Precision, Recall, dan F1-Score untuk dijadikan sebuah evaluasi. Gambar 8 adalah hasil pengujian dari model fine-tuning EfficientNet-B4.

Class	BPR	Healthy	Pod Borer
BPR	44	1	0
Healthy	3	166	1
Pod Borer	0	0	4

Gambar 8. Confusion Matrix Hasil Testing

Jika hasil pada tabel diatas dilakukan pengukuran dengan rumus, didapatkan hasil akurasi 97.72, presisi 98.47, recall 91% dan F1-skor 94,59%.

D. Komparasi

Sebagai bahan perbandingan, peneliti juga melatih model EfficientNet-B4 original untuk dibandingkan. Selain itu peneliti juga melatih beberapa model yang sering digunakan dengan pengaturan fine-tuning yang sama uuntuk model Xception, InceptionV3, DenseNet, dan RestNet dengan jumlah epoch yang sama, yaitu sepuluh epoch. Hasil komparasi disajikan.

Tabel 2. Komparasi Model CNN yang Diusulkan Dibandingkan dengan Model Original dan Model-Model Lain berdasarkan Confusion Matrix

Model	Acc	Prec	Rec
EfficientNet-B4 (Original)	89.9%	77.1%	61.6%
Xception	76.3%	25.4%	33%
InceptionV3	76.3%	25.4%	33%
DenseNet201	76.3%	25.4%	33%
ResNet50V2	76.3%	25.4%	33%
EfficientNet-B4 (Fine-tuning)	97.3%	97.3%	97.3%

SIMPULAN DAN SARAN

Dengan melihat dari hasil penelitian yang telah dilakukan, maka dapat ditarik kesimpulan yaitu akurasi untuk training dan validation untuk model EfficientNet-B4 yang sudah difine-tuning dapat mencapai 98%, sedangkan untuk model original hanya dapat mencapai 87% untuk training dan validation. Melakukan fine-tuning dapat meningkatkan akurasi pelatihan hingga 10%. Model EfficientNet-B4 dengan fine-tuning dapat berhasil untuk mengklasifikasikan dengan baik. Melakukan fine-tuning dan menambah dataset dengan augmentasi pada model lain tidak dapat mengklasifikasikan dengan baik. Memperbanyak data dengan cara augmentasi tidak membuat data menjadi seimbang, tetapi model EfficientNet-B4 dapat belajar dengan baik pada data tersebut. Memberikan tambahan layer untuk model EfficientNet-B4 memiliki peran yang cukup penting. Dense relu ditambahkan untuk mempercepat proses sehingga proses pelatihan tidak terlalu lama. Lalu ada layer dropout untuk mencegahnya terjadi overfitting. Selain itu penggunaan pooling layer digunakan untuk mereduksi data dimana ukuran dari data akan dikecilkan (downsampling). GlobalAveragePooling2D akan mengambil nilai rata-rata dari semua matrix datanya. Selain itu dapat terkonfirmasi dan terbukti bahwa model CNN yang diusulkan mampu bekerja pada data tidak seimbang. Penelitian mendatang dapat mengeksplorasi penerapan model deep learning lainnya, seperti Generative Adversarial Networks (GANs), untuk meningkatkan kemampuan augmentasi data. Selain itu, studi lebih lanjut dianjurkan untuk mengintegrasikan teknologi IoT (Internet of Things) dalam sistem deteksi penyakit tanaman kakao, memungkinkan pengumpulan data secara real-time dan aplikasi model klasifikasi secara langsung di lapangan, untuk intervensi penyakit yang lebih cepat dan

efisien.

DAFTAR PUSTAKA

- [1] N. A. Nor Muhammad *et al.*, "Data on RNA-seq analysis of the cocoa pod borer pest *Conopomorpha cramerella* (Snellen) (Lepidoptera: Gracillariidae)," *Data Br.*, vol. 34, p. 106638, Feb. 2021, doi: 10.1016/j.dib.2020.106638.
- [2] J. B. Mbarga *et al.*, "Field testing an oil-based *Trichoderma asperellum* formulation for the biological control of cacao black pod disease, caused by *Phytophthora megakarya*," *Crop Prot.*, vol. 132, p. 105134, Jun. 2020, doi: 10.1016/j.cropro.2020.105134.
- [3] S. B. Imanulloh, A. R. Muslikh, and D. R. I. M. Setiadi, "Plant Diseases Classification based Leaves Image using Convolutional Neural Network," *J. Comput. Theor. Appl.*, vol. 1, no. 1, pp. 1–10, Aug. 2023, doi: 10.33633/jcta.v1i1.8877.
- [4] M. Rahimzadeh and A. Attar, "A modified deep convolutional neural network for detecting COVID-19 and pneumonia from chest X-ray images based on the concatenation of Xception and ResNet50V2," *Informatics Med. Unlocked*, vol. 19, p. 100360, 2020, doi: 10.1016/j.imu.2020.100360.
- [5] S. Albahli, N. Ayub, and M. Shiraz, "Coronavirus disease (COVID-19) detection using X-ray images and enhanced DenseNet," *Appl. Soft Comput.*, vol. 110, p. 107645, Oct. 2021, doi: 10.1016/j.asoc.2021.107645.
- [6] K. Joshi, V. Tripathi, C. Bose, and C. Bhardwaj, "Robust Sports Image Classification Using InceptionV3 and Neural Networks," *Procedia Comput. Sci.*, vol. 167, pp. 2374–2381, Jan. 2020, doi: 10.1016/j.procs.2020.03.290.
- [7] P. Zhang, L. Yang, and D. Li, "EfficientNet-B4-Ranger: A novel method for greenhouse cucumber disease recognition under natural complex environment," *Comput. Electron. Agric.*, vol. 176, p. 105652, Sep. 2020, doi: 10.1016/j.compag.2020.105652.
- [8] K. Ali, Z. A. Shaikh, A. A. Khan, and A. A. Laghari, "Multiclass skin cancer classification using EfficientNets – a first step towards preventing skin cancer," *Neurosci. Informatics*, vol. 2, no. 4, p. 100034, Dec. 2022, doi: 10.1016/j.neuri.2021.100034.
- [9] F. Pérez-García, R. Sparks, and S. Ourselin, "TorchIO: A Python library for efficient loading, preprocessing, augmentation and patch-based sampling of medical images in deep learning," *Comput. Methods Programs Biomed.*, vol. 208, p. 106236, Sep. 2021, doi: 10.1016/j.cmpb.2021.106236.
- [10] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," May 2019, [Online]. Available: <http://arxiv.org/abs/1905.11946>
- [11] A. P. Wibowo and D. R. I. M. Setiadi, "Performance Analysis of Deep Learning Models for Sweet Potato Image Recognition," in *2022 International Seminar on Application for Technology of Information and Communication (iSemantic)*, Sep. 2022, pp. 101–106. doi: 10.1109/iSemantic55962.2022.9920423.
- [12] S. Khan, H. Rahmani, S. A. A. Shah, and M. Bennamoun, "A Guide to Convolutional Neural Networks for Computer Vision," *Synth. Lect. Comput. Vis.*, vol. 8, no. 1, pp. 1–207, Feb. 2018, doi: 10.2200/S00822ED1V01Y201712COV015.
- [13] S. Ali, A. Hashmi, A. Hamza, U. Hayat, and H. Younis, "Dynamic and Static Handwriting Assessment in Parkinson's Disease: A Synergistic Approach with C-Bi-GRU and VGG19," *J. Comput. Theor. Appl.*, vol. 1, no. 2, pp. 151–162, 2023, doi: 10.33633/jcta.v1i2.9469.
- [14] V. R. Kamath, V. S, and V. M, "Transferred Fusion Learning using Skipped Networks," Nov. 2020, [Online]. Available: <http://arxiv.org/abs/2011.05895>
- [15] Y. Li, K. He, D. Xu, and D. Luo, "A transfer learning method using speech data as the source domain for micro-Doppler classification tasks," *Knowledge-Based Syst.*, vol. 209, p. 106449, Dec. 2020, doi: 10.1016/j.knosys.2020.106449.
- [16] J. Yang, Y. Xu, H. Cao, H. Zou, and L. Xie, "Deep learning and transfer learning for device-free human activity recognition: A survey," *J. Autom. Intell.*, vol. 1, no. 1, p. 100007, Dec. 2022, doi: 10.1016/j.jai.2022.100007.
- [17] M. Canayaz, "C+EffxNet: A novel hybrid approach for COVID-19 diagnosis on CT images based on CBAM and EfficientNet," *Chaos, Solitons & Fractals*, vol. 151, p. 111310, Oct. 2021, doi: 10.1016/j.chaos.2021.111310.
- [18] A. Shamila Ebenezer, S. Deepa Kanmani, M. Sivakumar, and S. Jeba Priya, "Effect of image transformation on EfficientNet model for COVID-19 CT image classification," *Mater. Today Proc.*, vol. 51, pp. 2512–2519, 2022, doi: 10.1016/j.matpr.2021.12.121.
- [19] L. Yang *et al.*, "A dual attention network based on efficientNet-B2 for short-term fish school feeding behavior analysis in aquaculture," *Comput. Electron. Agric.*, vol. 187, p. 106316, Aug. 2021, doi:

Judul Artikel

Nama Penulis

- 10.1016/j.compag.2021.106316.
- [20] S. Salian and D. Sawarkar, "MELANOMA SKIN LESION CLASSIFICATION USING IMPROVED EFFICIENTNETB3," *Jordanian J. Comput. Inf. Technol.*, vol. 08, no. 01, p. 1, 2022, doi: 10.5455/jjcit.71-1636005929.
- [21] B. Li, O. K. Ersoy, C. Ma, Z. Pan, W. Wen, and Z. Song, "A 4F optical diffuser system with spatial light modulators for image data augmentation," *Opt. Commun.*, vol. 488, p. 126859, Jun. 2021, doi: 10.1016/j.optcom.2021.126859.
- [22] G. Jang, J. Lee, J.-G. Lee, and Y. Liu, "Distributed fine-tuning of CNNs for image retrieval on multiple mobile devices," *Pervasive Mob. Comput.*, vol. 64, p. 101134, Apr. 2020, doi: 10.1016/j.pmcj.2020.101134.
- [23] "Dense layer." https://keras.io/api/layers/core_layers/dense/ (accessed Dec. 12, 2023).
- [24] M. Agarwal, S. Gupta, and K. K. Biswas, "A new Conv2D model with modified ReLU activation function for identification of disease type and severity in cucumber plant," *Sustain. Comput. Informatics Syst.*, vol. 30, p. 100473, Jun. 2021, doi: 10.1016/j.suscom.2020.100473.
- [25] P. Singh, P. Raj, and V. P. Namboodiri, "EDS pooling layer," *Image Vis. Comput.*, vol. 98, p. 103923, Jun. 2020, doi: 10.1016/j.imavis.2020.103923.
- [26] I. Rodriguez-Martinez, J. Lafuente, R. H. N. Santiago, G. P. Dimuro, F. Herrera, and H. Bustince, "Replacing pooling functions in Convolutional Neural Networks by linear combinations of increasing functions," *Neural Networks*, vol. 152, pp. 380–393, Aug. 2022, doi: 10.1016/j.neunet.2022.04.028.
- [27] M. S. Sunarjo and H. Gan, "High-Performance Convolutional Neural Network Model to Identify COVID-19 in Medical Images," *J. Comput. Theor. Appl.*, vol. 1, no. 1, pp. 19–30, Aug. 2023, doi: 10.33633/jcta.v1i1.8936.
- [28] K. R. Prilianti, T. H. P. Brotosudarmo, S. Anam, and A. Suryanto, "Performance comparison of the convolutional neural network optimizer for photosynthetic pigments prediction on plant digital image," 2019, p. 020020. doi: 10.1063/1.5094284.
- [29] A. Susanto, I. U. Wahyu Mulyono, C. Atika Sari, E. Hari Rachmawanto, D. R. I. M. Setiadi, and M. K. Sarker, "Handwritten Javanese script recognition method based 12-layers deep convolutional neural network and data augmentation," *IAES Int. J. Artif. Intell.*, vol. 12, no. 3, p. 1448, Sep. 2023, doi: 10.11591/ijai.v12.i3.pp1448-1458.
- [30] H. T. Adityawan, O. Farroq, S. Santosa, H. M. M. Islam, M. K. Sarker, and D. R. I. M. Setiadi, "Butterflies Recognition using Enhanced Transfer Learning and Data Augmentation," *J. Comput. Theor. Appl.*, vol. 1, no. 2, pp. 115–128, Nov. 2023, doi: 10.33633/jcta.v1i2.9443.
- [31] F. Mustofa, A. N. Safriandono, and A. R. Muslikh, "Dataset and Feature Analysis for Diabetes Mellitus Classification using Random Forest," *J. Comput. Theor. Appl.*, vol. 1, no. 1, pp. 41–48, Sep. 2023, doi: 10.33633/jcta.v1i1.9190.

Fine Tuning Model Convolutional Neural Network EfficientNet-B4 dengan Augmentasi Data untuk Klasifikasi Penyakit Kakao

ORIGINALITY REPORT

10%

SIMILARITY INDEX

8%

INTERNET SOURCES

4%

PUBLICATIONS

4%

STUDENT PAPERS

PRIMARY SOURCES

1	jurnal.unmer.ac.id Internet Source	2%
2	Submitted to Indiana University Student Paper	1%
3	Submitted to Udayana University Student Paper	1%
4	dspace.uii.ac.id Internet Source	1%
5	ichi.pro Internet Source	<1%
6	stackoverflow.com Internet Source	<1%
7	Submitted to uwe Student Paper	<1%
8	Submitted to STT PLN Student Paper	<1%

repository.its.ac.id

9	Internet Source	<1 %
10	e-journal.lp2m.uinjambi.ac.id Internet Source	<1 %
11	Miftahuz Zuhan, Yosi Kristian. "Detection of Porang Plant Diseases and Pests (Amorphophallus Muelleri) Based on Leaf Imagery Utilizing DCNN Transfer Learning", JTECS : Jurnal Sistem Telekomunikasi Elektronika Sistem Kontrol Power Sistem dan Komputer, 2023 Publication	<1 %
12	publikasi.dinus.ac.id Internet Source	<1 %
13	repository.uin-suska.ac.id Internet Source	<1 %
14	suparmi.staff.unri.ac.id Internet Source	<1 %
15	docplayer.info Internet Source	<1 %
16	jurnal.usu.ac.id Internet Source	<1 %
17	"Intelligent Computing and Optimization", Springer Science and Business Media LLC, 2023 Publication	<1 %

18	developers-id.googleblog.com Internet Source	<1 %
19	student-activity.binus.ac.id Internet Source	<1 %
20	docobook.com Internet Source	<1 %
21	Alda Cendekia Siregar. "Identifikasi Telinga Berdasarkan Fitur Geometrik dan KNN", CYBERNETICS, 2017 Publication	<1 %
22	core.ac.uk Internet Source	<1 %
23	inass.org Internet Source	<1 %
24	pdfs.semanticscholar.org Internet Source	<1 %
25	prosiding.konik.id Internet Source	<1 %
26	doku.pub Internet Source	<1 %

Exclude quotes Off
Exclude bibliography On

Exclude matches Off