

Implementasi Metode *Convolutional Neural Network* Dalam Pengembangan Klasifikasi Jamur Layak Konsumsi Menggunakan Arsitektur *EfficientNetV2*

Mujahidin Syamil Kaffah
Department of Computer Engineering
Telkom University
Surabaya, Indonesia
syamilkaffa11@gmail.com

Isa Hafidz
Department of Computer Engineering
Telkom University
Surabaya, Indonesia
isahafidz@telkomuniversity.ac.id

Pangestu Widodo
Department of Computer Engineering
Telkom University
Surabaya, Indonesia
pangestuw@telkomuniversity.ac.id

Abstrak — Indonesia memiliki keanekaragaman hayati termasuk berbagai jenis jamur. Beberapa di antaranya aman dikonsumsi sementara lainnya beracun. Identifikasi yang salah dapat menyebabkan keracunan serius sehingga dibutuhkan sistem yang mampu mengklasifikasikan jenis jamur secara otomatis. Penelitian ini bertujuan mengembangkan sistem klasifikasi jamur layak konsumsi menggunakan metode *Convolutional Neural Network* (CNN) dengan arsitektur *EfficientNetV2*. Dataset terdiri dari 3000 citra jamur layak konsumsi dan 3000 citra jamur tidak layak konsumsi yang diperoleh dari *Kaggle*. Tahapan penelitian meliputi *preprocessing* citra, *augmentasi* data, *tuning hyperparameter*, dan evaluasi menggunakan *confusion matrix*. Hasil penelitian menunjukkan bahwa kombinasi *hyperparameter learning rate* 0.0001, *batch size* 32 dan 50 *epoch* memberikan akurasi tertinggi sebesar 95,75%. Sistem ini diharapkan mampu membantu masyarakat dalam mengidentifikasi jamur dengan lebih aman dan akurat.

Kata kunci— *Convolutional Neural Network*, *EfficientNetV2*, Klasifikasi, *Hyperparameter*.

I. PENDAHULUAN

Indonesia memiliki keanekaragaman hayati yang luar biasa termasuk puluhan ribu jenis jamur yang tumbuh di negara ini [1]. Dari 70.000 spesies jamur di dunia, sekitar 2.000 aman dikonsumsi, sementara 10% dari 30 spesies jamur beracun dianggap mematikan [2]. Beberapa contoh jamur yang dapat dimakan (*edible mushrooms*) antara lain *Pleurotus ostreatus* (jamur tiram) dan *Auricularia auricula* (jamur kuping) [3][4]. Di sisi lain, ada juga jamur yang sangat beracun seperti *Amanita phalloides* dan *Russula emetica* [5][6]. Serta beberapa jenis yang hanya boleh dimakan sebagai obat [7][8].

Jamur dapat menjadi sumber nutrisi yang baik. Namun kesalahan dalam mengidentifikasinya dapat menyebabkan keracunan serius karena beberapa jenis jamur mengandung senyawa berbahaya bagi tubuh manusia [9]. Bentuk, warna, dan tekstur payung adalah ciri morfologi jamur yang dapat dilihat untuk membedakan jamur yang dapat dan tidak dapat dikonsumsi [10][11]. Namun pada beberapa jamur beracun memiliki bentuk yang sangat mirip dengan jamur yang dapat

dikonsumsi sehingga sulit untuk membedakannya dengan mata telanjang [12].

Kemajuan teknologi khususnya dalam bidang pengolahan citra digital memungkinkan pengembangan sistem klasifikasi jamur secara otomatis [13]. Penerapan metode *Convolutional Neural Network* (CNN) dalam klasifikasi citra telah menjadi topik penelitian yang menarik dalam beberapa tahun terakhir [14]. *Convolutional Neural Network* (CNN) telah terbukti efektif dalam menganalisis data citra dengan keunggulan mengenali pola visual yang kompleks melalui lapisan *convolutional* [15][16]. Dengan memanfaatkan CNN komputer dapat dilatih untuk mengenali pola visual karakteristik jamur konsumsi atau beracun, sehingga dapat memungkinkan proses klasifikasi jamur secara otomatis [17].

Pada penelitian [18] dilakukan identifikasi jamur yang layak dan tidak layak konsumsi menggunakan *Convolutional Neural Network* (CNN) dengan arsitektur *EfficientNet*. Dataset yang digunakan terdiri dari 3000 citra jamur beracun dan 3000 citra jamur tidak beracun. Sistem yang dirancang mampu mencapai akurasi sebesar 89% dan nilai *loss* 0,7963.

Pada penelitian [19] menggunakan model *EfficientNetV2* dalam memprediksi jenis kelamin pada wajah pengguna masker. Data terdiri dari 72.318 data *training* dan 16.813 data *testing*. Hasil penelitian menunjukkan akurasi sebesar 94.5% untuk data *training* dan 94.7% untuk data *testing*.

Penelitian [20] menggunakan *Naïve Bayes* untuk klasifikasi jamur dengan akurasi terbaik 100%, namun menggunakan metode berbeda dari penelitian ini. Penelitian [21] mengklasifikasikan jamur berdasarkan genus dengan CNN, menghasilkan akurasi pengujian 76%.

Penelitian [22] menunjukkan bahwa arsitektur *MobileNetV2* memberikan akurasi 92,19% pada klasifikasi jamur menggunakan metode *Transfer Learning*.

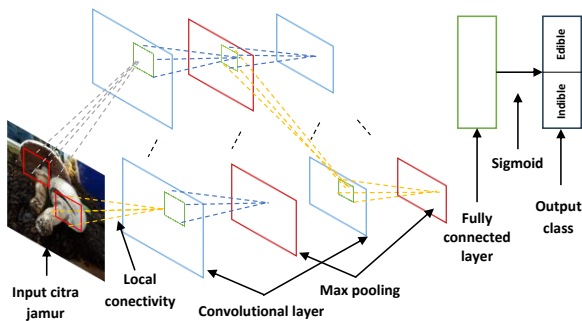
EfficientNetV2 memiliki kecepatan pelatihan yang lebih cepat dan efisiensi parameter yang lebih baik daripada model sebelumnya [23]. Dari beberapa penelitian, arsitektur *EfficientNetV2* terbukti efisien dalam memprediksi citra dengan akurasi yang tinggi. Berdasarkan hal tersebut penulis

melakukan penelitian tentang “Implementasi metode *Convolutional Neural Network* dalam pengembangan klasifikasi jamur layak konsumsi menggunakan arsitektur *EfficientNetV2*”.

II. KAJIAN TEORI

A. Convolutional Neural Network

Convolutional Neural Network (CNN) adalah arsitektur deep learning yang digunakan dalam klasifikasi gambar otomatis. CNN mengekstraksi fitur penting dari gambar melalui beberapa convolutional layer, memungkinkan model untuk mengenali pola dan membuatnya lebih unggul dalam klasifikasi citra dibandingkan dengan metode berbasis fitur manual. Struktur utama CNN meliputi convolutional layer, pooling layer, dan fully connected layer [24].



GAMBAR 1
(ARSITERKTUR CNN)

1. Convolutional Layer

Convolutional layer adalah komponen utama dalam CNN yang melakukan ekstraksi fitur menggunakan kernel berukuran 3x3 atau 5x5. Kernel ini melakukan operasi konvolusi dengan memindai setiap piksel gambar dan menghasilkan feature map yang diteruskan ke lapisan berikutnya. Proses konvolusi melibatkan operasi perkalian antara elemen kernel dan data input, yang hasilnya dijumlahkan [13].

2. Pooling Layer

Pooling layer bertujuan untuk melakukan down sampling dan pengurangan dimensi gambar, mengurangi beban komputasi jaringan. Teknik *pooling* yang umum digunakan adalah *max pooling* dan *average pooling*, yang bertujuan untuk mengurangi ukuran citra tanpa mengurangi informasi penting. Beberapa teknik lain seperti *Lp pooling* dan *mixed pooling* juga digunakan untuk mengurangi *overfitting* [25].

3. Fully Connected Layer

Fully Connected (FC) layer terletak di akhir jaringan CNN dan menghubungkan neuron di lapisan sebelumnya dengan neuron di lapisan berikutnya. *FC layer* digunakan untuk mengklasifikasikan fitur yang diekstraksi oleh convolutional dan pooling layers, dengan setiap neuron terhubung ke seluruh neuron di lapisan sebelumnya dan berikutnya [16].

B. EfficientNetV2

EfficientNetV2 merupakan arsitektur *convolutional network* yang lebih efisien dibandingkan dengan model pendahulunya, *EfficientNet* [26]. *EfficientNetV2* memungkinkan pelatihan model 3-9x lebih cepat dengan

ukuran parameter yang lebih kecil, hingga 6.8x lebih kecil, yang membuatnya lebih efisien dalam penggunaan sumber daya komputasi.

Peningkatan utama pada *EfficientNetV2* termasuk penambahan Fused-MBConv untuk mengatasi kelemahan lapisan MBConv pada *EfficientNet*, terutama pada tahap awal pelatihan. Selain itu, penggabungan MBConv dan Fused-MBConv mempercepat pelatihan tanpa menambah jumlah parameter. Arsitektur *EfficientNetV2* terdiri dari beberapa blok baru yang meningkatkan kinerja dan efisiensi, termasuk MBConv, Fused-MBConv, dan Squeeze-and-Excitation (SE), yang menunjukkan struktur model yang lebih optimal dan efisien dibandingkan dengan *EfficientNet* [23].

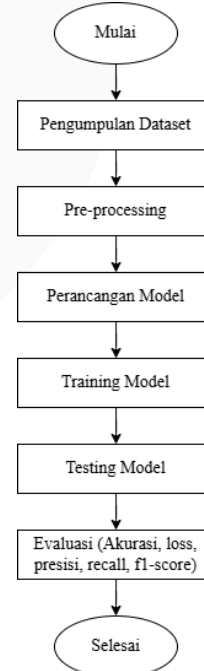
TABEL 1

Stage c	Operator $\hat{F}_c$	Stride	# Channels $\hat{C}_c$	# Layers $\hat{L}_c$
0	Conv3x3	2	24	1
1	Fused-MBConv1, k3x3	1	24	2
2	Fused-MBConv4, k3x3	2	48	4
3	Fused-MBConv4, k3x3	2	64	4
4	MBConv4, k3x3, SE0.25	2	128	6
5	MBConv6, k3x3, SE0.25	1	160	9
6	MBConv6, k3x3, SE0.25	2	256	15
7	Conv1x1 & Pooling & FC	-	1280	1

III. METODE

A. Desain Sistem

Pada tugas akhir ini dibuat sistem untuk melakukan klasifikasi jamur yang bisa dan tidak bisa dimakan menggunakan *Convolutional Neural Network* dengan arsitektur *EfficientNetV2*.



GAMBAR 2

(DIAGRAM ALUR KERJA SISTEM)

Flowchart pada Gambar 1 menjelaskan alur kerja sistem.

Dimulai dari *input* dataset yang berisi dua kelas jamur (bisa dimakan dan tidak bisa dimakan). Dataset ini diproses melalui tahap *preprocessing* untuk persiapan pelatihan. Model dilatih dan diuji menggunakan CNN dengan arsitektur *EfficientNetV2*, lalu melakukan klasifikasi jamur. Tahap akhir adalah evaluasi performa menggunakan parameter seperti akurasi, *loss*, presisi, *recall*, dan *f1-score* untuk menilai kinerja sistem.

B. Dataset

Penelitian ini menggunakan dataset citra jamur dari *Kaggle* [27] dalam format JPG dan model warna RGB. Dataset terdiri dari 29.100 citra, terbagi menjadi dua kategori: jamur bisa dimakan (12.775 citra) dan tidak bisa dimakan (16.325 citra).

Untuk mengurangi beban komputasi pada arsitektur *EfficientNetV2*, digunakan 6.000 citra dengan pembagian masing-masing 3.000 citra per kategori. Dataset dibagi menjadi data *training* dan *testing*. Data *training* digunakan untuk melatih sistem mengenali pola citra, sedangkan data *testing* digunakan untuk mengevaluasi kinerja sistem. Terdapat dua skenario proses validasi: *split* dataset (80% *training*, 20% *validation*) dan *cross-validation*. Gambar di bawah ini memperlihatkan contoh dari dataset citra jamur yang akan digunakan ini.



(a)

(b)

GAMBAR 3

(CONTOH DATASET YANG AKAN DIGUNAKAN (A) *ARMILLARIA TABESCENS* (B) *HYPOMYCES LACTIFLUORUM*)

C. Pre-processing

Pre-processing bertujuan mengoptimalkan kualitas citra, menghilangkan *noise*, dan menyesuaikan data agar sesuai untuk pelatihan CNN. Pada tahap ini citra jamur akan di *resize* menjadi 224x224 piksel untuk menyesuaikan ukuran input model. Selain itu dilakukan augmentasi data untuk meningkatkan performa dan ketahanan model terhadap variasi data. Teknik augmentasi yang digunakan meliputi:

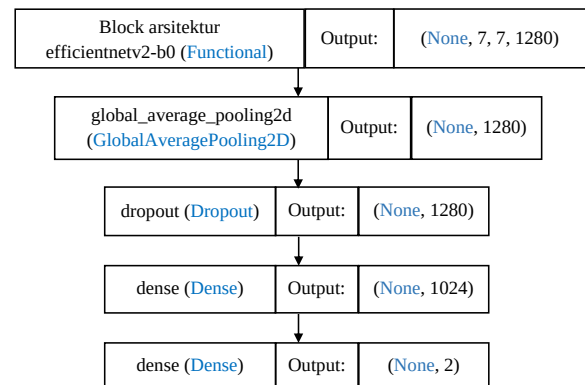
1. Rotasi hingga 30 derajat.
2. Pergeseran horizontal hingga 25% dari lebar citra.
3. Pergeseran vertikal hingga 25% dari tinggi citra.
4. Transformasi geser dengan sudut tertentu.
5. *Zoom* hingga 25%.
6. Membalikkan citra secara horizontal.

Tahap ini menghasilkan data pelatihan yang lebih bervariasi dan mempersiapkan citra untuk proses pelatihan model.

D. Arsitektur Sistem

Setelah melakukan *preprocessing* data, langkah selanjutnya adalah melatih model menggunakan arsitektur CNN. Pada tugas akhir ini akan menggunakan arsitektur *EfficientNetV2* dengan tambahan beberapa *layer* seperti *pooling*, *dropout* dan *dense*. Berikut merupakan gambar

rancangan desain arsitektur CNN yang akan digunakan dalam tugas akhir ini.



GAMBAR 4

(ARSITEKTUR MODEL SISTEM)

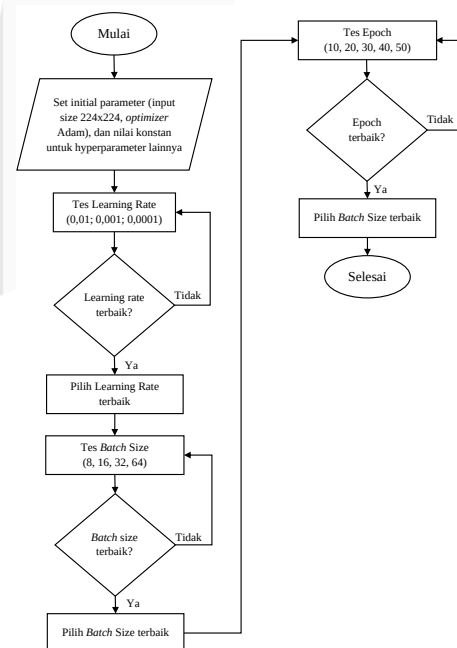
E. Parameter Pengujian Sistem

Parameter pengujian merupakan faktor-faktor yang divariasikan atau dimodifikasi dalam rangka menguji sistem yang telah dirancang. Pada penelitian ini terdapat beberapa variasi parameter yang digunakan adalah:

1. Parameter *learning rate* dengan nilai 0,01; 0,001 dan 0,0001.
2. Parameter *batch size* dengan ukuran 8, 16, 32 dan 64.
3. Parameter *epoch* 10, 20, 30, 40 dan 50.

F. Pengujian Sistem

Pengujian sistem bertujuan untuk mengukur sejauh mana sistem mampu berfungsi secara optimal dalam berbagai kondisi. Proses pengujian dilakukan dengan menganalisis pengaruh perubahan *hyperparameter* terhadap performa dataset dan arsitektur yang digunakan untuk mengidentifikasi kombinasi parameter terbaik bagi sistem ini.



GAMBAR 5

(FLOWCHART ALUR PENGUJIAN SISTEM)

Proses pengujian sistem dilakukan dengan menganalisis pengaruh perubahan *hyperparameter* terhadap performa dataset dan arsitektur yang digunakan, untuk mengidentifikasi kombinasi parameter terbaik. *Hyperparameter* yang diuji meliputi *optimizer*, *learning rate*, *batch size*, dan *epoch*.

Adam *optimizer* dipilih karena kecepatan, kestabilan, serta kemampuannya dalam menangani *noise*, *outlier*, atau gradien yang tidak merata. *Learning rate* diuji untuk menentukan nilai terbaik yang mengoptimalkan langkah pembaruan bobot selama pelatihan. Variasi *batch size* (8, 16, 32, 64) diuji untuk menemukan jumlah sampel ideal yang diproses dalam satu iterasi. Sementara itu pengujian *epoch* dilakukan untuk menentukan jumlah siklus pelatihan optimal.

Semua pengujian menggunakan ukuran input citra 224x224 piksel sebagai parameter konstan dengan Adam *optimizer*, serta kombinasi *learning rate* dan *batch size* terbaik dari pengujian sebelumnya.

IV. HASIL DAN PEMBAHASAN

A. Tuning Hyperparameter

Tuning hyperparameter merupakan langkah penting untuk memastikan model *EfficientNetV2* dapat mencapai performa optimal dalam tugas klasifikasi. *Hyperparameter* yang diuji dalam penelitian ini meliputi *learning rate*, *batch size* dan jumlah *epoch*. Setiap parameter diuji secara terpisah dengan pendekatan bertahap untuk menilai pengaruhnya terhadap performa model.

Proses *tuning* mengikuti skenario pengujian yang telah dijelaskan di sub-bab sebelumnya dengan pengaturan dataset, parameter awal model, dan metrik evaluasi untuk menilai performa model. *Tuning* dilakukan untuk menemukan kombinasi *hyperparameter* terbaik yang mengoptimalkan akurasi, *loss*, dan efisiensi pelatihan.

1. Tuning Hyperparameter Learning rate

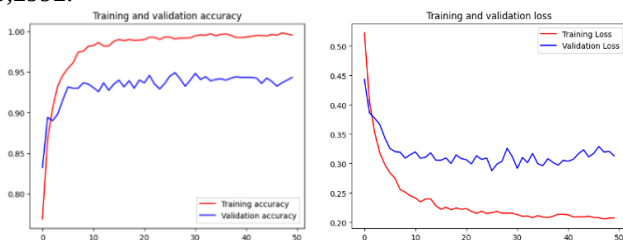
Pengujian sistem diawali dengan melakukan *tuning hyperparameter* pada *learning rate*. Variasi *learning rate* yang digunakan meliputi 0,01; 0,001; dan 0,0001. Pengujian dilakukan dengan parameter tetap yaitu *optimizer* Adam, 50 *epoch*, dan *batch size* 32.

TABEL 2

(HASIL PENGUJIAN LEARNING RATE)

Learning rate	Akurasi	Loss	Presisi	Recall	F1-score
0,01	78,91%	0,5268	78,67%	79,33%	79%
0,001	92,91%	0,3325	92,28%	93,66%	92,97%
0,0001	94,91%	0,2992	94,54%	95,33%	94,93%

Performa terbaik dicapai pada *learning rate* 0,0001 dengan akurasi tertinggi sebesar 94,91% dan *loss* terendah sebesar 0,2992.



GAMBAR 6

(GRAFIK PENGUJIAN LEARNING RATE TERBAIK)

Grafik *training* pada nilai ini menunjukkan kurva yang baik serta tidak menunjukkan grafik *overfitting* maupun *underfitting*. Langkah *tuning* parameter selanjutnya akan menggunakan parameter tetap dengan *learning rate* 0,0001

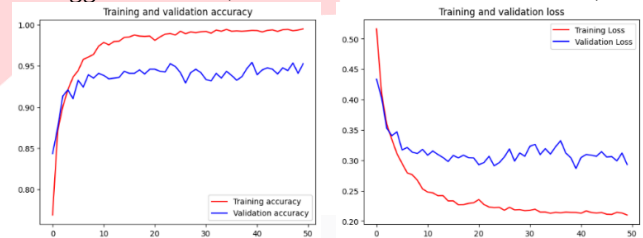
2. Tuning Hyperparameter Learning rate

Pada pengujian sistem yang kedua adalah *tuning hyperparameter batch size*. Variasi nilai *batch size* yang digunakan ada 4 yaitu 8, 16, 32 dan 64. Parameter tetap yang digunakan dalam pengujian adalah *optimizer* Adam, *epoch* 50 dan *learning rate* 0,0001.

TABEL 3
(HASIL PENGUJIAN BATCH SIZE)

Batch size	Akurasi	Loss	Presisi	Recall	F1-score	Waktu training (s)
8	94,76%	0,3024	94,20	94,5%	94,35%	5605,15
16	94,83%	0,3007	94,09%	95,66%	94,68%	4328,56
32	95,42%	0,2866	95,95%	94,83%	95,39%	4216,78
64	94,92%	0,2992	94,54%	95,33%	94,94%	4192,5

Performa terbaik dicapai pada *batch size* 32 dengan akurasi tertinggi sebesar 95,42% dan *loss* terendah sebesar 0,2866.



GAMBAR 7

(GRAFIK PENGUJIAN BATCH SIZE TERBAIK)

Grafik *training* pada nilai *batch size* 32 menunjukkan kurva yang baik dan tidak menunjukkan grafik *overfitting* maupun *underfitting*. Langkah *tuning* parameter selanjutnya akan menggunakan parameter tetap dengan nilai *batch size* 32.

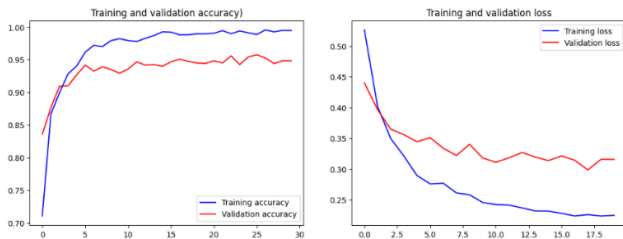
1. Tuning hyperparameter Epoch

Pada pengujian sistem yang ketiga dilakukan *tuning hyperparameter epoch*. Variasi nilai *epoch* yang digunakan yaitu 10, 20, 30, 40 dan 50. Parameter tetap yang digunakan dalam pengujian adalah *optimizer* Adam, *learning rate* 0.0001 dan *batch size* 32.

TABEL 4
(HASIL PENGUJIAN EPOCH)

Epoch	Akurasi	Loss	Presisi	Recall	F1-score	Waktu training (s)
10	92,91%	0,3149	94,20%	96,10%	95,15%	884,2
20	95,08%	0,2926	94,50%	96,00%	95,25%	1742,9
30	95,75%	0,2825	93,95%	95,83%	94,88%	2529,5
40	95,75%	0,2825	94,78%	96,83%	95,79%	3234,1
50	95,75%	0,2825	94,78%	96,83%	95,79%	4196,9

Hasil pengujian menunjukkan peningkatan performa model seiring dengan bertambahnya jumlah *epoch*. Akurasi tertinggi tercapai pada *epoch* 30. Namun pada *epoch* 40 dan 50 tidak ada peningkatan performa yang signifikan karena nilai maksimum akurasi dan *loss* tetap sama seperti pada *epoch* 30.



GAMBAR 8

(GRAFIK TERBAIK PENGUJIAN EPOCH)

Epoch 30 merupakan pilihan yang optimal untuk model ini karena dapat memberikan performa terbaik dengan waktu komputasi lebih efisien dibandingkan epoch yang lebih tinggi.

2. Kombinasi Hyperparameter EfficientNetV2

Hasil tuning hyperparameter menunjukkan kombinasi terbaik untuk EfficientNetV2 adalah learning rate 0,0001, batch size 32 dan epoch 30.

TABEL 5

(KOMBINASI TERBAIK EFFICIENTNETV2)

Hyperparameter	Akurasi	Loss	Pesisi	Recall	F1-score	Waktu training (s)
Learning rate 0,0001	95,75%	0,2825	94,78%	96,83%	95,79%	2529,5
Batch size 32						
Epoch 30						

B. Cross Validation

Cross validation digunakan untuk mengevaluasi performa model dengan membagi data menjadi beberapa fold dan memastikan model tidak hanya menghafal dataset yang sama. Nilai k-fold yang digunakan dalam penelitian ini adalah k = 5.

Pengujian dilakukan pada kedua model menggunakan dataset yang sama dengan hyperparameter terbaik dari hasil tuning sebelumnya. Akan tetapi untuk parameter epoch akan dilakukan penyesuaian karena keterbatasan komputasi. Tiap fold dilatih menggunakan 20 epoch, dengan total 100 epoch untuk seluruh proses.

TABEL 6

(HASIL PENGUJIAN CROSS VALIDATION EFFICIENTNETV2)

Fold	Akurasi	Loss	Pesisi	Recall	F1-score	Waktu training (s)
1	93,67%	0,1994	98,67%	98,8%	98,73%	8438,86
2	93,58%	0,2663				
3	93,67%	0,2020				
4	93,5%	0,2799				
5	94,33%	0,2018				
Rata-rata	93,75%	0,2299				

Model EfficientNetV2 mencapai rata-rata akurasi 93,75%, dengan akurasi tertinggi 94,33% (fold ke-5) dan terendah 93,50% (fold ke-4), menunjukkan performa yang stabil. Nilai loss rata-rata sebesar 0,2299 dengan rentang 0,1994 hingga 0,2799, mencerminkan kemampuan model meminimalkan kesalahan. Total waktu pelatihan untuk semua fold adalah 8438,86 detik.

C. Perbandingan dengan Penelitian Sebelumnya

Penelitian ini membandingkan implementasi metode Convolutional Neural Network (CNN) dengan arsitektur EfficientNetV2 dalam klasifikasi jamur menggunakan dataset

yang sama dengan penelitian terdahulu, yang terdiri dari 6.000 citra jamur dengan dua kelas: *edible* dan *inedible*. Pembagian data terdiri dari 80% untuk data latih dan 20% untuk data uji, sama seperti penelitian sebelumnya.

TABEL 7

(PERBANDINGAN EFFICIENTNETV2 DENGAN MODEL SEBELUMNYA)

Model	Akurasi	Loss	Pesisi	Recall	F1-score
EfficientNetV2 (Penelitian Ini)	95,75%	0,2825	94,67%	96,76%	95,71%
EfficientNet (Penelitian Sebelumnya)	89%	0,7963	89%	89%	89%

Penelitian ini menggunakan EfficientNetV2 dan memperoleh hasil terbaik dengan akurasi 95,75%, loss 0,2825, presisi 94,67%, recall 96,76%, dan f1-score 95,71%. Waktu pelatihan yang dibutuhkan adalah 2529,5 detik dengan pengaturan learning rate 0,0001, batch size 32, dan epoch 30. Sebagai perbandingan, penelitian sebelumnya dengan model EfficientNet mencapai akurasi 89%, loss 0,7963, presisi 89%, recall 89%, dan f1-score 89%, dengan pengaturan parameter berbeda.

Peningkatan akurasi sekitar 6% menunjukkan bahwa EfficientNetV2 lebih efisien dalam memproses citra dengan lebih sedikit parameter, sehingga mengurangi loss dan meningkatkan performa. Penelitian ini juga menerapkan augmentasi data, seperti rotasi dan zoom, yang tidak digunakan dalam penelitian sebelumnya, untuk memperkaya variasi citra dan mengurangi overfitting. Selain itu, pengujian tuning hyperparameter yang lebih mendalam menghasilkan performa model yang lebih baik, dibandingkan dengan parameter standar yang digunakan sebelumnya.

Secara keseluruhan, penerapan EfficientNetV2, augmentasi data, dan tuning hyperparameter yang lebih cermat berhasil meningkatkan akurasi dan performa model dalam klasifikasi citra jamur.

V. KESIMPULAN

Pada penelitian ini, implementasi CNN dengan arsitektur EfficientNetV2 menghasilkan akurasi terbaik sebesar 95,75% dan nilai loss yang rendah yaitu 0,2825, dengan metrik evaluasi presisi 94,78%, recall 96,83%, dan f1-score 95,79%. Hasil cross validation menunjukkan rata-rata akurasi 93,72% dan loss rata-rata 0,3081, dengan waktu total 8483,29 detik, yang mengindikasikan stabilitas performa pada berbagai subset data. Pengujian tuning hyperparameter menghasilkan akurasi tertinggi dengan kombinasi parameter batch size 32, epoch 30, dan learning rate 0,0001. Secara keseluruhan, EfficientNetV2 mampu menghasilkan akurasi tinggi dengan waktu pelatihan yang efisien, menjadikannya solusi andal untuk klasifikasi jamur layak konsumsi, dengan potensi pengembangan lebih lanjut melalui teknik augmentasi data dan optimasi hyperparameter.

DAFTAR PUSTAKA

- [1] A. Retnowati, Rugayah, J. S. Rahajoe, and D. Arifiani, *Status Keanekaragaman Hayati Indonesia : Kekayaan Jenis Tumbuhan Indonesia*. 2019.
- [2] T. Temesgen, "Application of Mushroom as Food and Medicine," *Adv. Biotechnol. Microbiol.*, vol. 11,

- no. 4, 2018, doi: 10.19080/aibm.2018.11.555817.
- [3] F. Wu *et al.*, "Global Diversity and Updated Phylogeny of Auricularia (Auriculariales, Basidiomycota)," *J. Fungi*, vol. 7, no. 11, p. 933, Nov. 2021, doi: 10.3390/jof7110933.
 - [4] T. Nakazawa *et al.*, "Pleurotus ostreatus as a model mushroom in genetics, cell biology, and material sciences," *Appl. Microbiol. Biotechnol.*, vol. 108, no. 1, p. 217, Dec. 2024, doi: 10.1007/s00253-024-13034-4.
 - [5] P. Alvarado *et al.*, "Amanita Section Phalloideae Species in the Mediterranean Basin: Destroying Angels Reviewed," *Biology (Basel)*, vol. 11, no. 5, p. 770, May 2022, doi: 10.3390/biology11050770.
 - [6] H. Lee, K. Wissitrassameewong, M. S. Park, A. Verbeken, J. Eimes, and Y. W. Lim, "Taxonomic revision of the genus *Lactarius* (Russulales, Basidiomycota) in Korea," *Fungal Divers.*, vol. 95, no. 1, pp. 275–335, Mar. 2019, doi: 10.1007/s13225-019-00425-6.
 - [7] Y. Suryani, *Mikologi*. 2020.
 - [8] J. E. I. Codjia *et al.*, "Morphology, Multilocus Phylogeny, and Toxin Analysis Reveal *Amanita* albolimbata, the First Lethal *Amanita* Species From Benin, West Africa," *Front. Microbiol.*, vol. 11, p. 599047, Nov. 2020, doi: 10.3389/fmicb.2020.599047.
 - [9] A. Zubair and A. R. Muslikh, "Identifikasi Jamur Menggunakan Metode K-Nearest Neighbor Dengan Ekstraksi Ciri Morfologi," *J. Kegiat. Pertan. Desa Palapan*, no. September, pp. 965–972, 2018.
 - [10] D. Liu, "Classification of medically important fungi," in *Molecular Medical Microbiology*, Elsevier, 2024, pp. 2763–2777.
 - [11] A. Beniah Nugroho, "Pendeteksian Edibilitas Jamur Konsumsi Berdasarkan Citra Digital dengan Convolutional Neural Network," Universitas Multimedia Nusantara, 2022. [Online]. Available: <https://kc.umn.ac.id/id/eprint/22499>
 - [12] X. Liu, K. H. Ghazali, F. Han, and I. I. Mohamed, "Review of CNN in aerial image processing," *Imaging Sci. J.*, vol. 71, no. 1, pp. 1–13, 2023, doi: 10.1080/13682199.2023.2174651.
 - [13] M. A. Hossain and M. S. Alam Sajib, "Classification of Image using Convolutional Neural Network (CNN)," *Glob. J. Comput. Sci. Technol.*, vol. 19, no. May, pp. 13–18, 2019, doi: 10.34257/gjcstdvol19is2pg13.
 - [14] A. Nawrocka, M. Nawrocki, and A. Kot, "Research study of image classification algorithms based on Convolutional Neural Networks," *Proc. 2023 24th Int. Carpathian Control Conf. ICC3 2023*, pp. 299–302, 2023, doi: 10.1109/ICCC57093.2023.10178933.
 - [15] G. Gu, J.; Wang, Z.; Kuen, J.; Ma, L.; Shahroudy, A.; Shuai, B.; Liu, T.; Wang, X.; Wang, "Recent Advances in Convolutional Neural Networks," *Pattern Recognit*, vol. 77, pp. 354–377, 2018.
 - [16] L. Alzubaidi *et al.*, *Review of deep learning: concepts, CNN architectures, challenges, applications, future directions*, vol. 8, no. 1. Springer International Publishing, 2021. doi: 10.1186/s40537-021-00444-8.
 - [17] S. ORAL, İ. ÖKTEN, and U. YÜZGEÇ, "Fungus Classification Based on CNN Deep Learning Model," *Bitlis Eren Üniversitesi Fen Bilim. Derg.*, vol. 12, no. 1, pp. 226–241, 2023, doi: 10.17798/bitlisfen.1225375.
 - [18] A. J. Panahatan, "Identifikasi Jamur yang Layak dan Tidak Layak Konsumsi melalui Convolutional Neural Network (CNN) dengan Arsitektur EfficientNet," pp. 4–9, 2023.
 - [19] N. Adisaputra Sinaga and R. Rosnelly, "Analisis Penggunaan Model EfficientNetV2 Dalam Memprediksi Jenis Kelamin Pada Wajah Pengguna Masker," *J. Tek. Inform. dan Sist. Inf.*, vol. 9, no. 3, 2022, [Online]. Available: <http://jurnal.mdp.ac.id>
 - [20] R. Hayami, Soni, and I. Gunawan, "Klasifikasi Jamur Menggunakan Algoritma Naïve Bayes," *J. CoSciTech (Computer Sci. Inf. Technol.)*, vol. 3, no. 1, pp. 28–33, 2022, doi: 10.37859/coscitech.v3i1.3685.
 - [21] U. S. Rahmadhani and N. L. Marpaung, "Klasifikasi Jamur Berdasarkan Genus Dengan Menggunakan Metode CNN," *J. Inform. J. Pengemb. IT*, vol. 8, no. 2, pp. 169–173, 2023, doi: 10.30591/jpit.v8i2.5229.
 - [22] E. I. Haksoro and A. Setiawan, "Pengenalan Jamur Yang Dapat Dikonsumsi Menggunakan Metode Transfer Learning Pada Convolutional Neural Network," *J. ELTIKOM*, vol. 5, no. 2, pp. 81–91, 2021, doi: 10.31961/eltikom.v5i2.428.
 - [23] B. Z. Demiray, M. Sit, and I. Demir, "DEM Super-Resolution with EfficientNetV2," 2021, [Online]. Available: <http://arxiv.org/abs/2109.09661>
 - [24] A. A. Elngar *et al.*, "Image Classification Based On CNN: A Survey," *J. Cybersecurity Inf. Manag.*, vol. 6, no. 1, p. PP. 18-50, 2021, doi: 10.54216/jcim.060102.
 - [25] M. M. K. Sarker, F. Akram, M. Alsharid, V. K. Singh, R. Yasrab, and E. Elyan, "Efficient Breast Cancer Classification Network with Dual Squeeze and Excitation in Histopathological Images," *Diagnostics*, vol. 13, no. 1, pp. 1–15, 2023, doi: 10.3390/diagnostics13010103.
 - [26] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," *36th Int. Conf. Mach. Learn. ICML 2019*, vol. 2019-June, pp. 10691–10700, 2019.
 - [27] "mushroom imgs." [Online]. Available: <https://www.kaggle.com/datasets/melissaaukland/mushroom-imgs>