

Klasifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan Metode Convolutional Neural Network

Classification Of Plants Based On Leaf Image Using Convolutional Neural Network Method

Indra Rizki Ramadhani¹⁾, Agung Nilogiri^{2)*}, Qurrota A'yun³⁾

¹⁾Mahasiswa Fakultas Teknik, Universitas Muhammadiyah Jember

Email: indrarizki518@gmail.com

²⁾Dosen Fakultas Teknik, Universitas Muhammadiyah Jember* Koresponden Author

Email: agungnilogiri@unmuhjember.ac.id

³⁾Dosen Fakultas Teknik, Universitas Muhammadiyah Jember

Email: qurrota.ayun@unmuhjember.ac.id

Abstrak

Salah satu makhluk hidup yang dibutuhkan oleh manusia adalah tumbuhan. Sebelum memanfaatkan tumbuhan, langkah terbaik yaitu mengidentifikasinya. Hal ini karena memiliki banyak kemiripan antar tumbuhan, seperti kemiripan daun. Maka dari itu dibutuhkan teknologi yang dapat mengidentifikasi tumbuhan berdasarkan daunnya. *Convolutional Neural Network* (CNN) merupakan salah satu pengembangan algoritma *Multilayer Perceptron* (MLP) yang mampu mengidentifikasi data dua dimensi, seperti gambar dan suara. Arsitektur CNN yang digunakan yaitu VGG-16 dengan terdapat 2300 citra daun. VGG-16 yang dikembangkan oleh K. Simonyan dan A. Zisserman memiliki 13 convolution layer dengan fungsi aktivasi ReLU dan 5 pooling layer. VGG-16 mempunyai dua Fully Connected layer yang mempunyai 4096 neuron setiap layer. Pada layer akhir terdapat pengklasifikasian 23 kelas menggunakan aktivasi softmax. Akurasi yang dihasilkan berhasil mencapai 92,6%. Sedangkan akurasi pada identifikasi mampu mencapai 92% yang didapatkan dari pengujian 50 citra. Kesimpulan dari penelitian ini bahwa algoritma *Convolutional Neural Network* (CNN) yang dibangun relatif mampu mengidentifikasi jenis tumbuhan berdasarkan citra daun dan tingkat efektivitas model mengidentifikasi jenis tumbuhan berdasarkan citra daun kurang baik.

Kata Kunci : *Convolutional Neural Network*, VGG-16, Identifikasi

Abstract

One of the living things needed by humans is plants. Before mandating a plant, the best step is to identify it. This is because it has many similarities between plants, such as the resemblance of leaves. Therefore, technology is needed that can identify plants based on their leaves. Convolutional Neural Network (CNN) is one of the development of multilayer perceptron (MLP) algorithms that are able to identify two-dimensional data, such as images and sounds. CNN architecture used is VGG-16 with 2300 leaf images. The VGG-16 developed by K. Simonyan and A. Zisserman has 13 convolution layers with ReLU activation function and 5 pooling layers. VGG-16 has two Fully Connected layers that have 4096 neurons per layer. At the end layer there is a classification of 23 classes using softmax activation. The resulting accuracy reached 92.6%. While the accuracy of identification is able to reach 92% obtained from testing 50 images. The conclusion of this study is that the convolutional neural network (CNN) algorithm built relatively able to identify plant types based on leaf imagery and the level of effectiveness of the model identifies plant types based on unfavorable leaf image.

Keywords : *Convolutional Neural Network*, VGG-16, Identification

1. PENDAHULUAN

Salah satu makhluk hidup yang dibutuhkan oleh manusia adalah tumbuhan. Selain penyedia oksigen bagi manusia, tumbuhan juga dapat berguna sebagai obat – obatan, bahan makanan, kosmetik, dan sumber pangan dan sandang [16]. Banyaknya jenis tumbuhan yang ada, bahkan manusia masih sulit untuk membedakan jenis dari tumbuhan itu sendiri. Pada umumnya, setiap tumbuhan memiliki ciri tersendiri. Langkah terbaik sebelum memanfaatkan tumbuhan adalah mengidentifikasinya. Hal ini dilakukan karena banyak kemiripan antara satu tumbuhan dengan tumbuhan lainnya, seperti kemiripan daun [17].

Perkembangan teknologi yang sangat pesat tidak dapat dipungkiri di era digital ini. Selain komputer yang dituntut untuk dapat membantu menyelesaikan pekerjaan manusia dengan lebih cepat, kini banyak juga berkembangnya *software* yang mampu meniru kecerdasan manusia (kecerdasan buatan) [9]. Kecerdasan buatan (*artificial intelligence*) merupakan pemrograman komputer yang dapat menjalankan intruksi atau aplikasi yang dapat melakukan suatu hal dimana dalam pandangan manusia disebut cerdas [13]. Seiring berkembangnya kecerdasan buatan, diharapkan komputer dapat meniru pola fungsi otak manusia, seperti berfikir dan belajar. Nantinya, dapat diharapkan menjadi suatu pilihan dalam mengidentifikasi jenis tumbuhan.

Dalam pengolahan citra terdapat banyak metode untuk dapat mengolahnya, salah satu metode yang dapat digunakan untuk pengolahan citra adalah metode *Convolutional Neural Network* (CNN). CNN (*Convolutional Neural Network*) adalah salah satu algoritma dari *deep learning*. *Deep Learning* merupakan bagian dari *machine learning* yang mempelajari representasi data dengan memungkinkan model komputasi yang terdiri dari beberapa lapisan pemrosesan [10]. Algoritma CNN dapat mengelola data dalam bentuk dua dimensi, seperti gambar dan suara. Tetapi, seperti halnya metode *deep learning* lainnya, proses pelatihan model yang lama menjadi kelemahan yang dimiliki oleh metode CNN [14]. Namun, seiring

perkembangan teknologi optimasi GPU (*Graphics Processing Unit*) kelemahan tersebut bisa teratasi sehingga proses pelatihan dapat berlangsung dengan cepat.

Pada Penelitian ini berkaitan dengan pengimplementasian CNN (*Convolutional Neural Network*) untuk mengidentifikasi jenis tumbuhan berdasarkan citra daun. Berdasarkan penelitian sebelumnya, metode yang digunakan untuk identifikasi jenis tumbuhan berdasarkan citra daun adalah dengan menggunakan metode arsitektur Alexnet. Akurasi yang diperoleh dari hasil klasifikasi mencapai akurasi 85%, dan pada akurasi identifikasi mendapatkan 90% dari 40 citra yang diuji [7].

Dataset yang digunakan pada penelitian ini diambil dari penelitian sebelumnya yaitu milik Neeraj Kumar [8] yang dipublikasikan dalam *Proceedings of the 12th European Conference on Computer Vision (ECCV)* pada bulan Oktober 2012. Data yang digunakan sebanyak 2300 citra dengan 23 *genus*. Arsitektur CNN yang digunakan adalah VGG16.

Pada penelitian ini model CNN yang digunakan yaitu VGG-16 dengan memakai 224×224 untuk ukuran *input* citra. Diharapkan arsitektur VGG16 dapat melakukan identifikasi jenis tumbuhan dengan mendapatkan akurasi yang tinggi, serta dengan proses pelatihan model yang cepat dibanding peneliti sebelumnya.

2. TINJAUAN PUSTAKA

A. Convolutional Neural Network

Convolutional Neural Network (CNN) merupakan algoritma pengembangan dari *Multilayer Perceptron* (MLP) dan termasuk dari salah satu algoritma *deep learning* yang dapat melatih *dataset* besar dengan jutaan parameter pada data dua dimensi, contohnya gambar dan suara [4]. LeCun adalah orang yang pertama kali memperkenalkan CNN, terinspirasi dari model *neurocognition* yang sudah dibentuk sebelumnya. LeNet merupakan bentuk CNN yang pertama kali diterapkan oleh LeCun dalam masalah pengenalan angka tulis tangan [1].

Convolutional Neural Network (CNN) memiliki suatu *layer* dengan susunan neuron

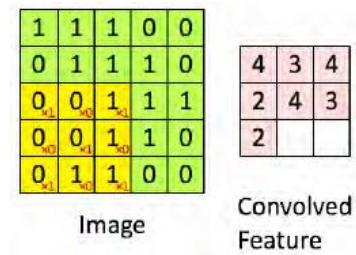
tiga dimensi (lebar, tinggi, kedalaman). Lebar dan tinggi adalah ukuran berdasarkan lapisan tersebut, sedangkan kedalaman sendiri ditetapkan pada jumlah lapisan. Lapisan tersebut kemudian secara berurutan melewati serangkaian pemrosesan. Pemrosesan dapat berupa *convolution layer*, *pooling layer*, *normalization layer*, *fully connected layer*, *loss layer*, dan lain sebagainya.

Secara umum lapisan pada CNN dibagi menjadi dua tipe, yaitu lapisan ekstraksi fitur (*feature extraction layer*) dan lapisan klasifikasi (*classification layer*). Pada lapisan pertama (lapisan ekstraksi fitur), terletak pada awal arsitektur atas beberapa lapisan dan di setiap lapisan tersusun atas neuron yang terhubung ke daerah local (*local region*) dari lapisan sebelumnya. *Convolutional layer* merupakan lapisan jenis pertama dan *pooling layer* pada lapisan kedua. Fungsi aktivasi diberlakukan di setiap lapisan. Pada lapisan ekstraksi fitur ini yang menerima masukan berupa gambar secara langsung dan memprosesnya sehingga keluaran berupa vektor untuk diolah pada lapisan selanjutnya.

Lapisan yang kedua yaitu *classification layer* (lapisan klasifikasi), lapisan ini terdiri dari beberapa lapisan dan masing-masing lapisan terdiri dari neuron yang terhubung secara penuh (*fully connected*) dengan lapisan lainnya. Lapisan ini menerima masukan dari hasil keluaran lapisan ekstraksi fitur yang berupa gambar vektor kemudian ditransformasikan seperti *multi neural network* dengan ditambahkan beberapa *hidden layer*. *Output* nantinya pada lapisan ini berupa skoring kelas untuk klasifikasi.

B. Convolution Layer

Convolution layer merupakan lapisan yang pertama kali menerima masukan citra secara langsung ke dalam arsitektur. Sama seperti operasi konvolusi, operasi pada lapisan ini adalah untuk melakukan operasi kombinasi linier *filter* terhadap daerah lokal [21].

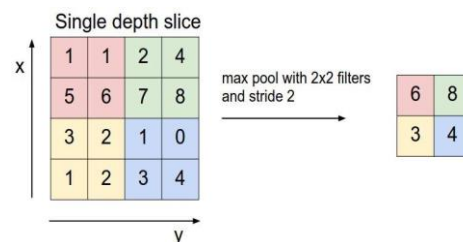


Gambar 1 Operasi Konvolusi
 Sumber: (Ilahiyah, 2018)

C. Pooling Layer

Pooling Layer biasanya digunakan setelah *convolutional layer*. *Pooling layer* adalah lapisan yang memanfaatkan fungsi dari *feature map* sebagai masukan dan memprosesnya melalui berbagai operasi statistik berdasarkan nilai piksel yang terdekat. Dalam model CNN, *Pooling layer* biasanya disisipkan setelah beberapa lapisan konvolusi secara teratur. *Pooling layer* yang disisipkan diantara lapisan konvolusi terdiri mempunyai *filter* dengan ukuran tertentu dan *stride* yang bergerak keseluruhan area *feature map* [12]. *Pooling layer* mengurangi ukuran spasial dan jumlah parameter pada jaringan untuk mempercepat komputasi dan mengendalikan terjadinya *overfitting*.

Terdapat beberapa jenis *pooling layer*, antara lain yaitu *average pooling*, *max pooling*, *Lp pooling*. Dalam sebagian besar CNN, metode *pooling layer* yang sering digunakan adalah *max pooling*. *max pooling* mengambil nilai yang paling maksimum pada masing-masing grid kecil yang telah dibagi untuk menata matriks citra yang telah direduksi, grid kecil tersebut merupakan hasil dari *output* operasi konvolusi seperti yang ditunjukkan pada Gambar 1.



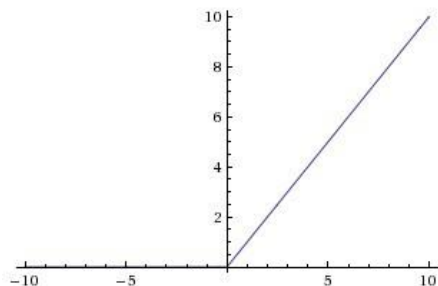
Gambar 2 Contoh Operasi Max Pooling
 Sumber: (Ilahiyah, 2018)

Gambar 2, menampilkan operasi menggunakan stride 2 dan ukuran filter 2x2.

Dimulai dengan ukuran input 4×4 , dari 4 angka masing-masing yang telah diambil pada masukan tersebut dipilih nilai yang paling maksimum dan membuat keluaran dengan ukuran baru menjadi 2×2

D. ReLU

Aktivasi ReLU (*Rectified Linear Unit*) merupakan lapisan aktivasi yang banyak digunakan pada model CNN. Kesederhanaan dan keandalan aktivasi ReLU karena peningkatan kinerjanya dibanding aktivasi lain yang tidak konsisten terhadap model dan *dataset* membuat banyak praktisi menyukai aktivasi ini [15]. Aktivasi fungsi ReLU melakukan *Thresholding* terhadap nilai nol pada piksel *input* citra. Aktivasi ReLU membuat nilai piksel yang mempunyai nilai kurang dari nol maka akan dijadikan menjadi 0 pada suatu citra tersebut. Dapat didefinisikan secara matematis yaitu :



Gambar 3 Aktivasi ReLU
 Sumber: (Anugerah, 2018)

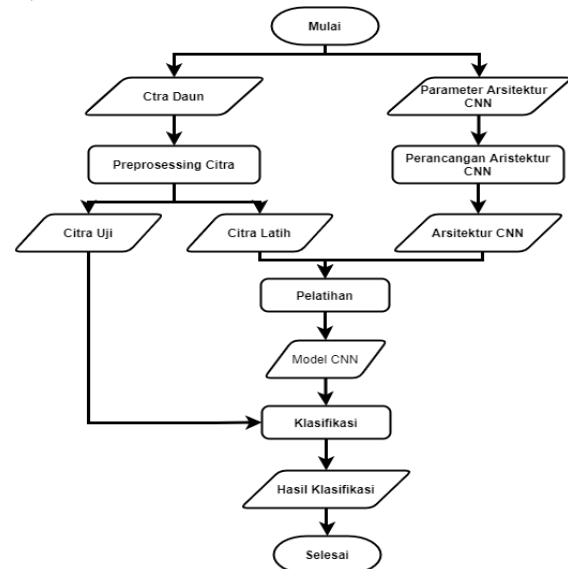
E. Softmax

Aktivasi *softmax* atau *softmax Classifier* adalah jenis fungsi aktivasi lain yang digunakan dalam CNN. Aktivasi *softmax* merupakan gabungan dari aktivasi *sigmoid*. Namun, aktivasi *softmax* tidak seperti aktivasi *sigmoid* yang dapat digunakan untuk klasifikasi biner. Aktivasi *softmax* biasanya menghitung probabilitas pada setiap kategori / kelas target dari semua kelas target yang mungkin. Saat digunakan dalam model multi-klasifikasi, aktivasi *softmax* akan mengembalikan setiap kelas dan kelas target akan mempunyai probabilitas tertinggi [11].

$$f(x) = \frac{e^{x_i}}{\sum_{j=1}^k e^{x_j}} \quad (1)$$

3. METODE PENELITIAN

Pada tugas akhir ini akan dirancang beberapa tahapan seperti pada diagram alir desain sistem yang ditunjukkan dengan Gambar 4.



Gambar 4 Diagram Alir Desain Sistem
 Sumber: Hasil Pengamatan

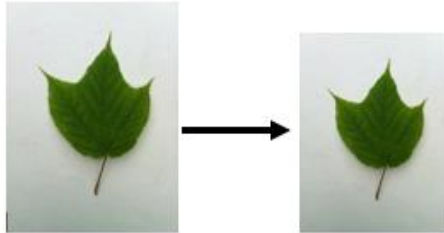
A. PraProses Data

Pada tahapan ini merupakan suatu proses untuk mempersiapkan dataset *Leafsnap* sebelum dilakukannya proses lain. Data citra dari dataset *Leafsnap* dilakukan pengolahan hingga siap digunakan oleh arsitektur CNN yang dibangun. Praproses data menerima masukan dari data citra, kemudian data itu dibagi menjadi dua bagian yaitu data pengujian dan data pelatihan. Selanjutnya, citra ditransformasi dengan *me-resize* piksel citra.

a. Resize Citra

Citra yang ada pada dataset *Leafsnap* memiliki ukuran resolusi yang berbeda-beda. Arsitektur CNN yang dipakai pada penelitian tugas akhir ini hanya menerima *input* dengan ukuran resolusi yang sama ketika proses pelatihan. Proses *resize* bertujuan untuk

mengubah ukuran asli dari citra menjadi ukuran yang akan ditetapkan. Pada penelitian ini ukuran resolusi yang ditetapkan yaitu (224×224).



Gambar 5 Ilustrasi Resize Citra
 Sumber: Hasil Pengamatan

B. Pembangunan Model CNN

Pada pembangunan model CNN, Arsitektur yang digunakan terdiri dari dua bagian utama, yaitu lapisan ekstraksi dan lapisan klasifikasi. Pada tugas akhir ini, peneliti menggunakan arsitektur VGG-16 sebagai model CNN yang digunakan.

a. Perancangan Arsitektur VGG-16

Jika disusun, arsitektur VGG-16 terdiri dari beberapa proses yang rinciannya adalah:

1. Masukkan Citra

Masukan citra dari arsitektur ini berukuran 224×224×3. Maka dari itu, citra yang mempunyai ukuran berbeda akan di *resize* terlebih dahulu.

2. Convolution layer 1 dan 2

Pada proses konvolusi yang pertama dan kedua yaitu dengan ukuran 224×224×3 sebanyak 64 kali, dengan menggunakan kernel 3×3, *stride* 1, dan *zero padding*. Keluaran hasil dari konvolusi ini berukuran 224×224×64 yang dapat dihitung menggunakan rumus:

$$N_{out} = \left\lfloor \frac{N_{in} + 2p - k}{s} \right\rfloor + 1$$

$$N_{out} = \left\lfloor \frac{224 + 2 - 3}{1} \right\rfloor + 1 = 224$$

3. ReLU

ReLU merupakan fungsi aktivasi yang digunakan pada arsitektur VGG-16. Fungsi aktivasi ReLU menerapkan fungsi $f(x)=\max(0,x)$ yang berarti membuat nilai

piksel yang mempunyai nilai kurang dari nol maka akan dijadikan menjadi 0.

4. Max pooling

Pooling layer yang digunakan pada arsitektur VGG-16 memiliki ukuran yang sama semua dengan ukuran kernel 2×2 dan *stride* 2. *Pooling layer* ini untuk memperkecil ukuran dan dapat mempercepat penghitungan. Ukuran keluaran dari lapisan ini adalah 112 × 112 × 64, yang dapat dihitung menggunakan rumus berikut:

$$N_{out} = \left\lfloor \frac{224 + 0 - 2}{2} \right\rfloor + 1 = 112$$

5. Convolution layer 3 dan 4.

Proses konvolusi pada lapisan yang ketiga dan keempat dengan ukuran inputan 112×112×3 dengan 128 filter, menggunakan kernel 3×3, *stride* 1, dan *zero padding*. Keluaran hasil dari konvolusi ini berukuran 112×112×128 yang dapat dihitung menggunakan rumus:

$$N_{out} = \left\lfloor \frac{112 + 2 - 3}{1} \right\rfloor + 1 = 112$$

6. ReLU

7. Max pooling

Dimensi yang dihasilkan pada layer ini berukuran 56×56×128.

$$N_{out} = \left\lfloor \frac{112 + 0 - 2}{2} \right\rfloor + 1 = 56$$

8. Convolution layer 5, 6, dan 7.

Proses konvolusi pada lapisan yang ketiga dan keempat dengan ukuran inputan 56×56×3 dengan 256 filter, menggunakan kernel 3×3, *stride* 1, dan *zero padding*. Keluaran hasil dari konvolusi ini berukuran 56×56×256 yang dapat dihitung menggunakan rumus:

$$N_{out} = \left\lfloor \frac{56 + 2 - 3}{1} \right\rfloor + 1 = 56$$

9. ReLU

10. Max pooling

Dimensi yang dihasilkan pada layer ini berukuran 28×28×256.

$$N_{out} = \left\lfloor \frac{56 + 0 - 2}{2} \right\rfloor + 1 = 28$$

11. Convolution layer 8, 9, dan 10.

Proses konvolusi pada lapisan yang ketiga dan keempat dengan ukuran inputan $28 \times 28 \times 3$ dengan 512 filter, menggunakan kernel 3×3 , *stride* 1, dan *zero padding*. Keluaran hasil dari konvolusi ini berukuran $28 \times 28 \times 512$ yang dapat dihitung menggunakan rumus:

$$N_{out} = \left\lfloor \frac{28 + 2 - 3}{1} \right\rfloor + 1 = 28$$

12. ReLU

13. *Max pooling*

Dimensi yang dihasilkan pada layer ini berukuran $14 \times 14 \times 512$.

$$N_{out} = \left\lfloor \frac{28 + 0 - 2}{2} \right\rfloor + 1 = 14$$

14. *Convolution layer* 11, 12, dan 13.

Proses konvolusi pada lapisan yang ketiga dan keempat dengan ukuran inputan $14 \times 14 \times 3$ dengan 512 filter, menggunakan kernel 3×3 , *stride* 1, dan *zero padding*. Keluaran hasil dari konvolusi ini berukuran $14 \times 14 \times 512$ yang dapat dihitung menggunakan rumus:

$$N_{out} = \left\lfloor \frac{14 + 2 - 3}{1} \right\rfloor + 1 = 14$$

15. ReLU

16. *Max pooling*

Dimensi yang dihasilkan pada layer ini berukuran $7 \times 7 \times 512$.

$$N_{out} = \left\lfloor \frac{14 + 0 - 2}{2} \right\rfloor + 1 = 7$$

17. *Fully Connected Layer*

Agar dapat dilakukan proses *Fully Connected layer*, keluaran dari lapisan ekstraksi fitur, harus dilakukan "flatten" atau biasa disebut *reshape feature map* agar vektor sebelum dapat digunakan sebagai masukan dari *fully-connected layer*.

Arsitektur VGG-16 mempunyai ketentuan didalamnya yang mana pada *fully-connected layer* yang pertama dan kedua hanya memiliki 4096 neuron dan *fully-connected layer* yang ketiga hanya memiliki 1000 neuron. Setiap *fully-connected layer* pada arsitektur VGG-16 diikuti dengan fungsi aktivasi ReLU. Ukuran yang diolah pada lapisan ini adalah

25088×1 yang diperoleh dengan mengalikan $7 \times 7 \times 512$. Maka, ukuran inputan harus dilakukan regulasi *dropout* agar menjadi 4096 inputan.

18. *Softmax*

Aktifasi *Softmax* atau *Softmax Classifier* adalah aktifasi biasanya digunakan untuk model multi-klasifikasi, karena lapisan keluaran biasanya mempunyai lebih dari satu neuron.

C. Pelatihan Model

Dalam melakukan klasifikasi, sangat diperlukan untuk menguji kinerja model yang dipakai. Pada *Convolution Neural Network*, pengujian dilakukan untuk mengetahui performa dari arsitektur yang dipakai. Tugas akhir ini menggunakan metode *K-fold Cross Validation* sebagai metode pengujianya. *K-fold Cross Validation* adalah salah satu metode pengujian yang paling banyak digunakan untuk memperkirakan kesalahan prediksi model yang sebenarnya dan untuk menyesuaikan model [3].

Sebelum pada tahapan ini, dari 2300 data citra yang digunakan akan dibagi menjadi 90%:10%, dimana untuk citra latih terdapat 2070 citra dan untuk citra uji mempunyai 230 citra. Agar pada setiap bloknya memiliki jumlah data yang sama, peneliti menentukan nilai $K = 10$. Sehingga jumlah total citra latih sebanyak 2070 citra, akan dibagi di setiap bloknya menjadi 207 citra.

4. HASIL DAN PEMBAHASAN

A. Hasil Uji Coba *K-Fold Cross Validation*

Tahapan pengujian algoritma yang dilakukan yaitu dengan menggunakan *cross validation*. Pengujian pada penelitian ini membagi data *train* menjadi 10 folder yang dimana terdapat 2070 data citra yang telah dibagi menjadi data *train* dan data *validation* secara acak. Secara keseluruhan data *train* pada setiap folder memiliki 1863 citra, sedangkan untuk data *validation* berjumlah 207 citra. Masing-masing folder dilakukan percobaan sebanyak tiga kali dan dihitung akurasi. Tabel dari hasil pengujian menggunakan K-

Fold Cross Validation dapat dilihat pada Tabel 1.

Tabel 1 Hasil Pengujian Menggunakan K-
Fold Cross Vaidation

Fold	Percobaan 1 (%)	Percobaan 2 (%)	Percobaan 3 (%)	Rata-rata (%)
1	88,9	87,4	87,9	88,1
2	91,3	93,2	93,2	92,6
3	90,3	92,2	91,3	91,3
4	90,8	89,9	87,4	89,4
5	88,9	87,4	88,9	88,4
6	91,8	89,9	88,9	90,2
7	90,8	90,3	91,3	90,8
8	91,8	91,3	90,8	91,3
9	92,3	90,3	90,8	91,1
10	87,9	89,5	86,5	88

Sumber: Hasil Penelitian

Pada percobaan tiga kali diantaranya, *fold 2* mendapatkan nilai akurasi rata-rata tertinggi dengan 92,6%. Sedangkan *Fold 10* memperoleh nilai akurasi rata-rata terkecil dengan 88%. *Fold 2* pada percobaan kedua dan ketiga mendapatkan akurasi tertinggi dari seluruh percobaan yang ada dengan mendapatkan nilai 93,2%.

B. Hasil Uji Coba Data *Testing*

Skenario berikutnya *fold 2* digunakan sebagai data latih, karena pada *fold* ini mempunyai nilai akurasi rata-rata tertinggi. Selanjutnya pada percobaan terakhir, yaitu menggunakan data *testing* yang dimana data ini tidak pernah sama sekali digunakan pada sistem. Pada uji coba ini, jumlah citra yang diambil dari data *testing* sebanyak 50 citra yang mewakili seluruh *genus*. Hasil percobaan dapat dilihat pada Tabel 2.

Tabel 1 Hasil Uji Coba Menggunakan Data
Testing

	Prediksi Benar	Prediksi Salah	Akurasi (%)
Uji Coba	46	4	92

Sumber: Hasil Penelitian

Pengujian yang dilakukan menggunakan 50 citra baru mendapatkan nilai akurasi yang cukup tinggi. Pada uji cobanya mendapatkan prediksi benar sebanyak 46 citra dan prediksi salah sebanyak 4 citra sehingga mendapatkan nilai akurasi sebesar 92%.

C. Hasil Uji Coba Efektivitas Model

Skenario pada uji coba kali ini yaitu untuk menguji apakah model mengalami *overfitting* atau *underfitting*. *Fold 2* merupakan model yang digunakan pada pengujian ini karena memiliki akurasi rata-rata tertinggi. Selanjutnya data yang digunakan untuk uji coba tersebut menggunakan data yang belum dipakai sama sekali selama pelatihan maupun uji coba data *testing*. Pengujian yang dilakukan menggunakan 100 citra dengan yang mewakili seluruh *genus*.

a. *Fold 2* Percobaan 1

Performa dari model *fold 2* percobaan pertama dapat dilihat pada Tabel 1 dimana dapat dikatakan cukup baik menurut referensi dikarenakan mampu mendapatkan akurasi 91,3%. Jika dilihat dari banyaknya nilai sensitivitas lebih tinggi dibanding dengan nilai spesifisitas pada setiap *genus*, maka dapat dikatakan model *fold 2* mampu memprediksi *genus* tersebut dengan baik.

Pada uji coba ini, model *fold 2* percobaan pertama diuji kembali dengan menggunakan data baru yang belum terpakai selama pelatihan maupun uji coba data *testing*. Tabel dari hasil pengujian – dapat dilihat pada Tabel 3 dan Tabel 4.

Tabel 2 Confusion Matrix Percobaan 1

	PREDIKSI																							
ACTUAL	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	
0	3																							
1		2									1		1											
2			5																					
3				2	1													1						
4					5																			
5			1		4	4																		
6						4	4																	
7							4	4								1								
8									4	4							1	1						
9										3			1											
10								1			3												1	
11												4												
12									1				4											
13														2					2					
14	1														3									
15															1	4								
16																4	4							
17						1											3							
18																		4	4					
19																			4	4				
20																				4	4			
21								1													3			
22																						2		5

Sumber: Hasil Penelitian

Tabel 4 Akurasi, Sensitivitas, dan Spesifisitas Percobaan 1

Genus	Akurasi	Sensitivitas	Spesifisitas
Abies	0,9800	0,7500	0,9896
Acer	0,9800	0,5000	1,0000
Aesculus	0,9900	1,0000	0,9895
Amelanchier	0,9800	0,5000	1,0000
Betula	0,9900	1,0000	0,9895
Carya	0,9900	0,8000	1,0000
Catalpa	0,9900	1,0000	0,9896
Cedrus	0,9900	0,8000	1,0000
Camaecyparis	0,9800	0,8000	0,9895
Cornus	0,9700	0,7500	0,9792
Crataegus	0,9700	0,6000	0,9895
Cryptomeria	1,0000	1,0000	1,0000
Magnolia	0,9700	0,8000	0,9790
Malus	0,9800	0,5000	1,0000
Picea	0,9500	0,7500	0,9583
Pinus	0,9800	0,8000	0,9895
Populus	0,9600	0,8000	0,9684
Prunus	0,9900	0,7500	1,0000
Quercus	1,0000	1,0000	1,0000
Salix	1,0000	1,0000	1,0000
Sytrax	0,9900	0,7500	1,0000
Tilia	0,9900	1,0000	0,9898
Ulmus	1,0000	1,0000	1,0000

Sumber: Hasil Penelitian

Hasil dari uji coba ini, model *fold 2* percobaan pertama mendapatkan nilai akurasi sebesar 81%, sensitivitas 81%, dan spesifisitas 99% . Jika dilihat dari performa setiap *genus*, terdapat *genus* Aesculus, Betula, Catalpa, dan Tilia yang mendapati nilai sensitivitas lebih tinggi dibandingkan dengan nilai spesifisitas. Sedangkan untuk *genus* yang lain, memiliki nilai spesifisitas lebih tinggi maupun sama dibanding nilai sensitivitasnya.

Jika dilihat dari performanya, hasil pengujian *K-Fold Cross Validation* lebih baik dibanding dengan uji coba ini. Sehingga dapat diberi kesimpulan bahwa model *fold 2* percobaan pertama kurang baik dalam keefektivitasan untuk mendeteksi *genus* pada tumbuhan. Dapat dikatakan bahwa model *fold 2* percobaan pertama mengalami *overfitting* karena pada saat pelatihan memiliki performa yang baik, tetapi memiliki kinerja yang kurang baik ketika diuji dengan data baru yang belum pernah terlihat (*unseen data*).

b. Fold 2 Percobaan 2

Uji coba selanjutnya, yaitu menggunakan model *fold 2* percobaan kedua. Dimana model ini merupakan model dengan nilai akurasi tertinggi pada semua percobaan. Performa model *fold 2* percobaan kedua pada saat pelatihan dapat dikatakan baik, dengan mendapatkan nilai akurasi 93,2%.

Pada uji coba ini, skenario yang dilakukan sama seperti uji coba sebelumnya, dimana menggunakan 100 data citra baru (*unseen data*). Tabel hasil dari uji coba *fold 2* percobaan kedua dapat dilihat pada Tabel 5 dan Tabel 6.

Tabel 5 Confusion Matrix Percobaan 2

		PREDIKSI																						
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
ACTUAL	0	4																						
	1		1	1				1													1			
	2			3							1	1												
	3				3										1									
	4					5																		
	5			1		3	3									1								
	6						4																	
	7							5																
	8								5															
	9									3						1								
	10				1						3									1				
	11											4												
	12												4										1	
	13							1						3										
	14														4									
	15															1	4							
	16																4							1
	17														1			3						
	18																		4					
	19																			4				
	20																						3	
	21																							2
22																							5	

Sumber: Hasil Penelitian

Tabel 6 Akurasi, Sensitivitas, dan Spesifisitas Percobaan 2

Genus	Akurasi	Sensitivitas	Spesifisitas
Abies	1,0000	1,0000	1,0000
Acer	0,9700	0,2500	1,0000
Aesculus	0,9600	0,6000	0,9790
Amelanchier	0,9900	0,7500	1,0000
Betula	0,9900	1,0000	0,9895
Carya	0,9800	0,6000	1,0000
Catalpa	0,9800	1,0000	0,9792
Cedrus	1,0000	1,0000	1,0000
Camaecyparis	1,0000	1,0000	1,0000
Cornus	0,9700	0,7500	0,9792
Crataegus	0,9700	0,6000	0,9895
Cryptomeria	1,0000	1,0000	1,0000
Magnolia	0,9900	0,8000	1,0000
Malus	0,9700	0,7500	0,9792
Picea	0,9800	1,0000	0,9792
Pinus	0,9800	0,8000	0,9895
Populus	0,9900	0,8000	1,0000
Prunus	0,9800	0,7500	0,9896
Quercus	0,9900	1,0000	0,9896
Salix	1,0000	1,0000	1,0000
Sytrax	0,9800	0,7500	0,9896
Tilia	0,9900	1,0000	0,9898
Ulmus	1,0000	1,0000	1,0000

Sumber: Hasil Penelitian

Model *fold* 2 percobaan kedua pada uji coba ini mampu mendapatkan nilai akurasi sebesar 83%, sensitivitas 83%, dan spesifisitas 99%. Dilihat dari performa setiap *genus*, terdapat lima *genus* yang memiliki nilai sensitivitas lebih tinggi dibanding dengan spesifisitasnya. *Genus* tersebut yaitu Betula, Catalpa, Picea, Quercus, dan Tilia. Sedangkan untuk *genus* yang lain memiliki nilai yang sama ataupun lebih rendah nilai sensitivitasnya dibanding dengan spesifisitasnya.

Jika dibandingkan dengan hasil pengujian *K-Fold Cross Validation*, hasil uji coba ini tidak lebih baik. Dapat diambil kesimpulan bahwa model *fold* 2 percobaan kedua dalam keefektivitasan untuk mendeteksi *genus* pada tumbuhan kurang baik. Dilihat dari performa akurasi, model *fold* 2 percobaan kedua mengalami *overfitting* karena kinerja pada saat pelatihan yang baik, namun kinerja saat uji coba dengan menggunakan data baru (*unseen data*) kurang baik.

c. *Fold* 2 Percobaan 3

Model *fold* 2 percobaan ketiga sama seperti model *fold* 2 percobaan kedua, yaitu mendapatkan nilai akurasi tertinggi sebesar 93,2%. Performa *fold* 2 percobaan ketiga dapat dikatakan cukup baik pada saat pelatihan jika dilihat dari pelatihan dan performa setiap *genus*.

Uji coba terakhir ini, model yang digunakan yaitu *fold* 2 percobaan ketiga. Skenario uji coba ini menggunakan 100 data citra baru (*unseen data*). Hasil dari uji coba ini dapat dilihat pada Tabel 7 dan Tabel 8.

Tabel 7 Confusion Matrix Percobaan 3

		PREDICT																				
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
ACTUAL	0	4																				
	1		3								1											
	2			5																		
	3				1											3						
	4					5																
	5						4															
	6							4														
	7								4													
	8									5												
	9										3						1					
	10											1										
	11												2									
	12													4								
	13														4							
	14															3						
	15																4					
	16																	1	4			
	17																		5			
	18																			3		
	19																				3	
	20																					4
	21																					
22																						

Sumber: Hasil Penelitian

Tabel 8 Akurasi, Sensitivitas, dan Spesifisitas Percobaan 3

Genus	Akurasi	Sensitivitas	Spesifisitas
Abies	1,0000	1,0000	1,0000
Acer	0,9900	0,7500	1,0000
Aesculus	0,9900	1,0000	0,9895
Amelanchier	0,9400	0,2500	0,9688
Betula	1,0000	1,0000	1,0000
Carya	0,9900	0,8000	1,0000
Catalpa	0,9900	1,0000	0,9896
Cedrus	0,9900	0,8000	1,0000
Camaecyparis	0,9900	1,0000	0,9895
Cornus	0,9800	0,7500	0,9896
Crataegus	0,9600	0,4000	0,9895
Cryptomeria	1,0000	1,0000	1,0000
Magnolia	0,9900	0,8000	1,0000
Malus	0,9500	0,7500	0,9583
Picea	0,9800	1,0000	0,9792
Pinus	0,9900	0,8000	1,0000
Populus	1,0000	1,0000	1,0000
Prunus	0,9800	0,7500	0,9896
Quercus	0,9900	0,7500	1,0000
Salix	1,0000	1,0000	1,0000
Sytrax	0,9900	0,7500	1,0000
Tilia	1,0000	1,0000	1,0000
Ulmus	0,9900	1,0000	0,9895

Sumber: Hasil Penelitian

Pada uji coba terakhir ini, model *fold* 2 percobaan ketiga mendapatkan nilai akurasi sebesar 84%, sensitivitas 84%, dan spesifisitas 99%. Performa model terhadap uji coba ini, hanya terdapat lima genus yang memiliki sensitivitas lebih tinggi dibanding dengan spesifisitasnya seperti Aesculus, Catalpa, Camaecyparis, Picea, dan Ulmus. Untuk genus yang lain nilai sensitivitasnya lebih rendah dibanding dengan spesifisitasnya ataupun memiliki nilai yang sama.

Sehingga dari pernyataan di atas, dapat diambil kesimpulan bahwa model *fold* 2 percobaan ketiga kurang baik keefektifan dalam mendeksi *genus* pada tumbuhan. Selanjutnya, untuk kinerja model *fold* 2 percobaan ketiga dapat dikatakan *overfitting*, karena kinerja saat uji coba dengan data baru (*unseen* data) kurang baik dibanding dengan kinerja pada saat pelatihan.

5. KESIMPULAN

Berdasarkan hasil analisis selama perancangan, implementasi, dan uji coba sistem, dapat diambil kesimpulan sebagai berikut:

1. Tingkat rata-rata akurasi yang diperoleh dari pengujian menggunakan *fold cross validation* dengan nilai $K = 10$ mendapatkan nilai akurasi 92,6%. Sehingga dapat diberi kesimpulan bahwa metode *Convolutional Neural Network* (CNN) mampu dalam mengklasifikasi *genus* tumbuhan dengan baik. Hasil klasifikasi menggunakan data citra baru sebanyak 50 citra untuk menguji model menunjukkan terdapat 46 citra yang berhasil diklasifikasi dengan benar dan dapat disimpulkan bahwa nilai akurasi sistem sebesar 92%.
2. Berdasarkan uji coba efektivitas model, model *fold* 2 percobaan pertama mendapatkan nilai akurasi sebesar 81%, sensitivitas 81%, dan spesifisitas 99%. Pada model *fold* 2 percobaan kedua mendapatkan nilai akurasi sebesar 83%, sensitivitas 83%, dan spesifisitas 99%. Terakhir, model *fold* 2 percobaan ketiga mendapatkan nilai akurasi sebesar 84%,

sensitivitas 84%, dan spesifisitas 99%. Sensitivitas lebih kecil dibanding dengan spesifisitas artinya tingkat efektivitas model dalam mengidentifikasi jenis tumbuhan berdasarkan citra daun kurang baik.

6. SARAN

Saran yang diberikan untuk penelitian ini adalah sebagai berikut :

1. Menerapkan modifikasi seperti mengubah nilai filter, kombinasi nilai kernel, *learning rate*, dan algoritma optimasinya yang mungkin dapat menjadi penelitian lebih lanjut bagaimana pengaruh terhadap akurasi yang diperoleh.
2. Menambahkan lebih banyak data pada setiap kelasnya sehingga lebih banyak fitur yang didapatkan karena mungkin dapat meningkatkan akurasi dalam mengklasifikasi gambar.
3. Menerapkan data augmentasi secara *offline* ataupun *online* untuk menambah keberagaman citra data.

DAFTAR PUSTAKA

- [1] Anugerah, A. G. 2018. Klasifikasi Tingkat Keganasan Kanker Paru-paru pada Citra Computed Tomography (CT) Scan Menggunakan Metode Convolutional Neural Network= Classification Level Of Cancer Rate Rations In Computed Tomography (Ct) Scan Using Convolutional Neural Network Method (Doctoral dissertation, Institut Teknologi Sepuluh Nopember).
- [2] Baldi, P., & Sadowski, P. J. 2013. "Understanding dropout" . *Advances in neural information processing systems*, 26, 2814-2822.
- [3] Berrar, D. 2019. Cross-validation. *Encyclopedia of bioinformatics and computational biology*, 1, 542-545.
- [4] Chauhan, R., Ghanshala, K. K., & Joshi, R. C. 2018. Convolutional neural network (CNN) for image detection and recognition. In *2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC)* (pp. 278-282). IEEE.
- [5] Giana, S. S. (2014). *ANALISA HASIL IMAGE SHARPENING DENGAN MENGGUNAKAN METODE HIGH PASS FILTER* (Doctoral dissertation, Universitas Islam Negeri Sultan Syarif Kasim Riau).
- [6] Setiawan, H. H. 2018. "Klasifikasi Jenis Buah Pisang Dengan Image Processing Menggunakan Metode Backpropagation". Skripsi S1, Universitas Sanata Dharma.
- [7] Ilahiyah, S., & Nilogiri, A. 2018. "Implementasi Deep Learning Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan Convolutional Neural Network" . *JUSTINDO (Jurnal Sistem Dan Teknologi Informasi Indonesia)*, 3(2), 49-56.
- [8] Kumar, N., Belhumeur, P. N., Biswas, A., Jacobs, D. W., Kress, W. J., Lopez, I., & Soares, V. B. 2012. Leafsnap : A Computer Vision System for Automatic Plant Species Identification, 1–14.
- [9] Kusumaningrum, T. F. 2018. Implementasi Convolution Neural Network (CNN) Untuk Klasifikasi Jamur Konsumsi di Indonesia Menggunakan Keras.
- [10] Lecun, Yann, Yoshua Bengio, and Geoffrey Hinton. 2015. "Deep Learning" 521. <https://doi.org/10.1038/nature14539>.
- [11] Nwankpa, C., Ijomah, W., Gachagan, A., & Marshall, S. 2018. Activation functions: Comparison of trends in practice and research for deep learning. *arXiv preprint arXiv:1811.03378*.
- [12] Nurfitra, R. D., & Ariyanto, G. 2018. "Implementasi Deep Learning Berbasis Tensorflow Untuk Pengenalan Sidik Jari" . *Emitor: Jurnal Teknik Elektro*, 18(1), 22-27
- [13] Pujoseno, J. (2018). Implementasi

- Deep Learning Menggunakan Convolutional Neural Network Untuk Klasifikasi Alat Tulis.
- [14] Putra, I. W. S. E. 2016. *Klasifikasi citra menggunakan convolutional neural network (CNN) pada caltech 101* (Doctoral dissertation, Institut Teknologi Sepuluh Nopember).
- [15] Ramachandran, P., Zoph, B., & Le, Q. V. 2017. Searching for activation functions. *arXiv preprint arXiv:1710.05941*.
- [16] Setiaji, A., Hidayatno, A., & Christyono, Y. 2016. "Identifikasi Jenis Tumbuhan Berdasarkan Tulang Daun Menggunakan Alihragam Wavelet" . *Transient: Jurnal Ilmiah Teknik Elektro*, 5(1), 23-28.
- [17] Sibero, A. F., & Saleh, A. 2020. "Identifikasi Tanaman Herbal Berdasarkan Citra Daun Menggunakan Cosine Similarity dan Features Extraction" . *JURNAL MAHAJANA INFORMASI*, 5(1), 94-104.
- [18] Simonyan, K., & Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- [19] Vishwajith, S. 2017. Deep-Leafsnap. Diakses Februari 26, 2021. <https://github.com/sujithv28/Deep-Leafsnap>.
- [20] Wu, J. 2017. Introduction to convolutional neural networks. *National Key Lab for Novel Software Technology. Nanjing University. China*, 5, 23.
- [21] Zufar, M., & Setiyono, B. 2016. "Convolutional Neural Networks Untuk Pengenalan Wajah Secara Real-time" . *Jurnal Sains dan Seni ITS*, 5(2), 128862