

Klasifikasi Penyakit Pada Tanaman Singkong Menggunakan Arsitektur VGGNET Berbasis *Deep Learning*

Classification Of Diseases In Cassava Plant Using Deep Learning Based VGGNET Architecture

1st Ardian Eka Nugraha
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
ardianekan@student.telkomuniver
sity.ac.id

2nd Syamsul Rizal
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
syamsul@telkomuniversity.ac.id

3rd Nor Kumalasari Caesar Pratiwi
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
caecarnkcp@telkomuniversity.ac.i
d

Abstrak—Singkong adalah salah satu jenis tanaman ubi kayu yang mudah ditemui di Indonesia dan menjadi salah satu makanan pokok masyarakat Indonesia. Namun produksi singkong terhitung menurun dari tahun 2014 – 2018. Penurunan ini dapat disebabkan oleh berbagai hal, salah satunya penyakit singkong. Gejala penyakit singkong dapat diperiksa melalui kondisi daun tanaman singkong. Meskipun begitu, diperlukan pengetahuan lebih untuk dapat membedakan gejala satu penyakit dengan yang lainnya. Salah satu solusi yang dapat digunakan adalah metode *deep learning* yaitu *convolutional neural network* (CNN) untuk mengklasifikasikan penyakit singkong. Penulis menggunakan salah satu model CNN dari arsitektur VGGNet yaitu VGG16. Parameter uji untuk performa model CNN adalah akurasi, presisi, *recall*, dan *F1-score*. Model dilatih dengan *dataset* berupa citra RGB dengan ukuran 224×224 *pixel* yang dibagi menjadi 5 kelas (CMD, CGM, CBB, CBSD, dan daun sehat) dengan jumlah sebanyak 9430 citra. Proses pelatihan melibatkan *hyperparameter batch size*, *optimizer*, dan *learning rate* yang divariasikan. Hasil pengujian terbaik adalah model dengan *batch size* 32, *optimizer* SGD, dan *learning rate* 0,001 dengan akurasi *training* sebesar 82,53% dan akurasi *validation* sebesar 75%.

Kata kunci — penyakit singkong, *convolutional neural network*, VGGNet, *image classification*

Abstract—*Cassava is one type of cassava plant that is easily found in Indonesia and is one of the*

staple foods of the Indonesian people. However, cassava production has decreased from 2014 – 2018. This decline can be caused by various things, one of which is cassava disease. Symptoms of cassava disease can be checked through the condition of the leaves of the cassava plant. Even so, more knowledge is needed to be able to distinguish the symptoms of one disease from another. One solution that can be used is a deep learning method, namely convolutional neural network (CNN) to classify cassava diseases. The author uses one of the CNN models from the VGGNet architecture, namely VGG16. The test parameters for the CNN model performance are accuracy, precision, recall, and F1-score. The model uses a dataset in the form of RGB images with a size of 224×224 pixels which are divided into 5 classes (CMD, CGM, CBB, CBSD, and healthy leaves) with a total of 9430 images. The training process involves a variable batch size, optimizer, and learning rate hyperparameter. The best test results are models with batch size 32, optimizer SGD, and learning rate of 0.001 with training accuracy of 82.53% and validation accuracy of 75%.

Keywords — *cassava diseases, convolutional neural network, VGGNet, image classification*

I. PENDAHULUAN

Singkong merupakan jenis tanaman ketela pohon atau ubi kayu yang mudah ditemui karena tersebar luas di Indonesia dan menjadi salah satu makanan pokok di Indonesia. Namun, produksi singkong justru menurun dari tahun 2014 – 2018 [1]. Penurunan produksi ini dapat disebabkan oleh berbagai hal, salah

satunya adalah penyakit yang menyerang tanaman singkong. Untuk itu, kualitas singkong harus diperhatikan salah satunya dengan cara melakukan pengidentifikasian penyakit pada singkong secara cepat. Salah satu cara yang dapat digunakan adalah memanfaatkan teknologi *artificial intelligence* (AI) yang sedang berkembang untuk dapat mengklasifikasikan penyakit pada singkong dengan mudah.

Teknologi AI adalah salah satu bidang teknologi yang sedang berkembang saat ini. AI atau kecerdasan buatan adalah suatu bidang teknologi yang mempelajari tentang analisis dari suatu sistem yang bertindak secara cerdas [2]. Salah satu bidang dari AI yang sedang banyak digunakan adalah teknik pembelajaran mesin atau *machine learning* (ML). Definisi dari ML yaitu sebuah teknik yang berkaitan dengan membangun algoritma berdasarkan beberapa contoh fenomena [3]. Dalam kasus klasifikasi penyakit singkong, ML digunakan untuk mengklasifikasikan gejala penyakit singkong yang terlihat pada daun. Penggunaan salah satu cabang ML yaitu *deep learning* dengan metode *convolutional neural network* (CNN). Klasifikasi menggunakan metode ini sudah sering dilakukan dan mendapatkan hasil yang cukup baik. Penelitian tentang klasifikasi singkong [4] menggunakan arsitektur CNN *MobileNetV2* yang dilatih dengan *dataset* berjumlah 9430 mendapatkan hasil dimana model CNN dapat mengklasifikasikan 8 dari 10 data uji dengan tepat. Penelitian [5] membandingkan 4 arsitektur berbeda yakni *ResNet50*, *VGG16*, *VGG19*, dan *AlexNet* dalam mengklasifikasikan berbagai penyakit tanaman dengan *dataset* berupa gambar yang berjumlah 87867 gambar menghasilkan akurasi tertinggi sebesar 98,42% yang diraih oleh arsitektur *ResNet50*. Penelitian [6] menggunakan arsitektur *VGG16* dalam mengklasifikasikan hama daun tembakau menghasilkan akurasi diatas 90% dari setiap percobaan yang berarti bahwa arsitektur *VGG16* memiliki performa yang bagus dalam klasifikasi gambar. Dari

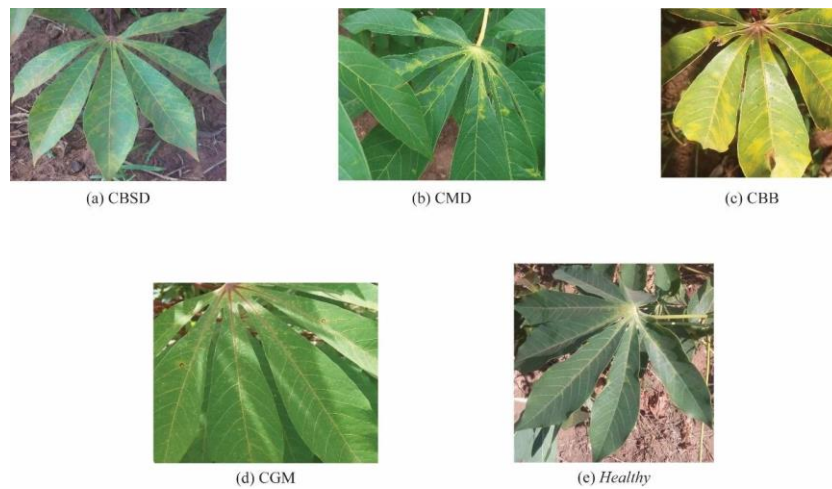
beberapa penelitian, dapat disimpulkan bahwa metode CNN dapat melakukan klasifikasi pada gambar dengan hasil cukup memuaskan.

Pada penelitian ini penulis menggunakan model CNN dengan arsitektur *VGGNet* untuk melakukan klasifikasi pada penyakit singkong. Model CNN tersebut dilatih dengan *dataset* berupa gambar berwarna yang berjumlah 9430 gambar yang terbagi ke dalam 5 kelas, *Cassava Mosaic Disease*, (CMD), *Cassava Green Mite* (CGM), *Cassava Bacterial Blight* (CBB), *Cassava Brown Streak Disease* (CBSD), dan daun sehat. Model dilatih dengan variasi *hyperparameter* berupa *batch size*, *optimizer*, dan *learning rate* yang berbeda. Model dengan konfigurasi *hyperparameter* terbaik lalu dianalisis performanya menggunakan *confusion matrix* untuk mendapatkan nilai presisi, *recall*, dan *F1-score*.

II. KAJIAN TEORI

A. Penyakit Tanaman Singkong

Tanaman singkong yang terjangkit penyakit dapat menurunkan kualitas singkong itu sendiri. Pendeteksian penyakit pada singkong biasanya dapat terlihat dari perubahan pada bagian tanaman misalnya pada bagian daun dan batang. Daun tanaman singkong yang terjangkit penyakit biasanya menunjukkan tanda – tanda seperti berubahnya warna dan bentuk dari daun yang dapat dilihat secara kasat mata. Diagnosis penyakit pada tanaman singkong yang cepat dan tepat dapat mempermudah proses pencegahan pada penyakit agar tidak merusak tanaman singkong. Beberapa contoh penyakit pada tanaman singkong yang gejalanya masih dapat dilihat secara kasat mata antara lain *Cassava Mosaic Disease* (CMD), *Cassava Green Mite* (CGM), *Cassava Bacterial Blight* (CBB), dan *Cassava Brown Streak Disease* (CBSD). Fokus dari penelitian ini adalah klasifikasi keempat penyakit tersebut berdasarkan gejala yang tampak pada daun singkong.



GAMBAR 1
DAUN SINGKONG YANG TERJANGKIT PENYAKIT DAN SEHAT

B. Pengolahan Citra Digital

Digital image processing bisa disebut sebagai proses pengolahan citra digital dengan menggunakan perangkat komputasi digital. Citra digital disini merujuk pada Citra RGB adalah citra (gambar) yang memiliki nilai – nilai tetap untuk mengindikasikan intensitas warna gambar dari masing – masing *pixel* [7]. *Pixel* sendiri adalah elemen pada citra digital yang direpresentasikan dengan angka numerik. Pada citra *grayscale*, intensitas *pixel* hanya direpresentasikan dengan satu nilai dan berkisar dari rentang nilai 0 – 255. Sedangkan pada citra berwarna, intensitas *pixel* direpresentasikan menjadi tiga nilai yang masing – masing mencerminkan nilai *Red*, *Green*, dan *Blue* (RGB). *Digital image processing* melibatkan sejumlah teknik dan algoritma yang dapat dilakukan.

C. Convolutional Neural Network (CNN)

CNN merupakan jaringan syaraf tiruan (*artificial neural network*) yang biasa digunakan untuk menganalisis gambar dalam bidang *machine learning*. CNN sendiri adalah salah satu metode *deep learning* yang berfokus pada pelatihan *neural network* dengan proses konvolusi didalamnya. CNN terdiri dari beberapa *layers* diantaranya *input layer*, *output layer*, serta beberapa *hidden layer*. Dalam *hidden layer* pada CNN terbagi lagi ke dalam beberapa *layer* yaitu *convolutional layer*, *pooling layer*, *fully connected layer*, dan *normalization layer* [8].

Proses pada CNN terbagi menjadi 2 yaitu proses *feature extraction* dan *classification* [9]. Pada proses *feature extraction* terjadi proses konvolusi pada *convolutional layer* yang akan memetakan fitur pada citra masukan dengan

melakukan operasi perkalian *dot* antara fitur citra dengan *filter*. Keluaran dari *convolutional layer* adalah fitur dari citra yang disebut *feature map* [9]. Dalam *convolutional layer* terdapat fungsi aktivasi *Rectified Linear Unit* (ReLU) yang membuat *feature map* akan selalu bernilai nol atau lebih. Selanjutnya *feature map* tersebut akan masuk ke dalam *pooling layer* yang berguna untuk mengurangi ukuran dari data citra masukan [9]. Setelah melalui *pooling layer*, akan dilanjutkan pada proses *classification* yang terjadi pada *fully connected layer* dimana proses *flatten* dilakukan untuk mengubah dimensi *feature map* yang berupa *array* multidimensi menjadi *array* satu dimensi [9].

D. Arsitektur VGGNet

VGGNet adalah arsitektur CNN rancangan Karen Simonyan dan Andrew Zisserman [10] yang mendapat peringkat pertama untuk *localization* dan peringkat kedua untuk *classification* pada *ImageNet Large Scale Visual Recognition Challenge* (ILSVRC) 2014. Arsitektur VGGNet ini menggunakan *convolutional filter* yang tergolong kecil yakni 3×3 dengan menambahkan kedalaman *network* hingga 16 – 19 *layers* [10]. Penggunaan kedalaman *network* dan *filter* seperti ini diklaim dapat menghasilkan akurasi yang jauh lebih tinggi.

TABEL 1
ARSITEKTUR VGGNET

VGG16	VGG19
16 weight layers	19 weight layers
Input (224×224 RGB image)	
Conv (3×3) – 64	Conv (3×3) – 64
Conv (3×3) – 64	Conv (3×3) – 64
Maxpool	
Conv (3×3) – 128	Conv (3×3) – 128
Conv (3×3) – 128	Conv (3×3) – 128
Maxpool	

Conv (3×3) – 256	Conv (3×3) – 256
Conv (3×3) – 256	Conv (3×3) – 256
Conv (3×3) – 256	Conv (3×3) – 256
<i>Maxpool</i>	
Conv (3×3) – 512	Conv (3×3) – 512
Conv (3×3) – 512	Conv (3×3) – 512
Conv (3×3) – 512	Conv (3×3) – 512
<i>Maxpool</i>	
<i>Fully connected – 4096</i>	
<i>Fully connected – 4096</i>	
<i>Fully connected – 1000</i>	
<i>Softmax</i>	

E. Transfer Learning

Metode *transfer learning* ini menggunakan model yang telah dilatih sebelumnya atau disebut *pre-trained model* yang lalu akan dilatih kembali menggunakan *dataset* yang berbeda. Dasar dari metode *transfer learning* ini yaitu *domain adaptation*. *Domain adaptation* adalah proses mengadaptasi satu atau lebih *domain* sumber untuk mengambil informasi yang bertujuan untuk meningkatkan performa dari target yang dilatih [11]. *Transfer learning* umumnya digunakan saat adanya keterbatasan data yang digunakan. Hal ini bisa terjadi karena berbagai kemungkinan seperti data yang diambil tidak mencukupi, ataupun data yang ingin digunakan tidak bisa diakses.

III. METODE

A. Desain Sistem

Pada penelitian ini, model *VGGNet* menjadi arsitektur yang akan diuji performanya dalam mengklasifikasikan penyakit singkong. Model yang digunakan adalah *VGG16* yang memiliki 16 *layers*. Model ini akan diuji dengan berbagai macam kombinasi *hyperparameter* yaitu *batch size*, *optimizer*, dan *learning rate*. Model dilatih menggunakan *dataset* berupa gambar penyakit singkong yang sudah melalui tahap *preprocessing*. Model lalu akan diuji melalui hasil akurasi dan *loss* yang didapatkan serta dari parameter presisi, *recall*, dan *F1-score*.

B. Dataset

Dataset yang digunakan berasal dari TensorFlow Datasets dengan judul *Cassava Disease Classification* yang merupakan bagian dari *Fine-Grained Visual Categorization workshop* dalam konferensi CVPR 2019 [12]. *Dataset* berupa gambar berwarna dengan format JPG yang berjumlah sebanyak 9430 dengan label masing – masing kelasnya. Dari total data dibagi menjadi 80% data latih (7545 gambar) dan 20% data validasi dan uji (1885 gambar). *Dataset* ini terbagi ke dalam 5 kelas yaitu CMD dengan 4431 gambar, CBSD dengan 2406 gambar, CGM dengan 1289 gambar, CBB dengan 777 gambar,

dan daun sehat dengan 527 gambar.

Sebelum digunakan sebagai data dalam tahap pelatihan, *dataset* ini akan melalui proses *preprocessing* terlebih dahulu. Tahapan *preprocessing* yang digunakan adalah normalisasi dan *resize*. Tahap normalisasi digunakan untuk mengubah skala intensitas warna pada gambar yang semula 0 – 255 menjadi hanya 0 – 1. *Resize* digunakan untuk mengubah ukuran gambar menjadi 224×224 *pixels*.

C. Parameter Pengujian Sistem

Beberapa parameter yang menjadi tolak ukur dari performa model CNN adalah akurasi, presisi, *recall*, dan *F1-score*.

1. Akurasi

Akurasi dapat dikatakan sebagai perbandingan antara data yang diprediksi secara benar dengan semua prediksi yang dilakukan [13].

$$\text{Akurasi} = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

2. Presisi

Presisi menunjukkan rasio seberapa banyak nilai yang benar – benar positif dari semua nilai yang diprediksi positif [13].

$$\text{Presisi} = \frac{TP}{TP+FP} \quad (2)$$

3. Recall

Recall menunjukkan jumlah nilai yang benar – benar positif dari semua nilai yang sebenarnya positif [13].

$$\text{Recall} = \frac{TP}{TP+FN} \quad (3)$$

4. F1-score

F1-score atau biasa disebut *f-measure* adalah *harmonic mean* dari presisi dan *recall* [13].

$$F1 - \text{Score} = 2 \times \frac{\text{Presisi} \times \text{Recall}}{\text{Presisi} + \text{Recall}} \quad (4)$$

IV. HASIL DAN PEMBAHASAN

A. Skenario Pengujian

Model *VGGNet* dilatih dengan berbagai kombinasi *hyperparameter* seperti *batch size*, *optimizer*, dan *learning rate* untuk mendapatkan hasil yang berbeda. Terdapat 3 *batch size* yang digunakan yaitu 16, 32, dan 64. Tiga jenis *optimizer* juga digunakan dalam proses pelatihan

diantaranya Adam, RMSprop, dan SGD. *Learning rate* yang digunakan divariasikan dari 0,01; 0,001; dan 0,0001. Pada setiap pelatihan model digunakan iterasi *epoch* sebanyak 25. Setelah mendapatkan konfigurasi *hyperparameter* terbaik yang didapat melalui nilai akurasi dan *loss*, model tersebut akan diuji kinerjanya melalui parameter presisi, *recall*, dan *F1-score*.

B. Hasil dan Analisis Pengujian

Berikut hasil dari pengujian untuk masing – masing scenario yaitu *batch size*, *optimizer*, dan *learning rate* secara berurutan.

1. Pengujian Skenario *Batch Size*

Pada pengujian ini *hyperparameter* selain *batch size* disetarakan yaitu *optimizer* RMSprop dan *learning rate* 0,001 serta menggunakan 25 *epochs*.

TABEL 2
HASIL PENGUJIAN *BATCH SIZE*

<i>Batch Size</i>	<i>Accuracy (%)</i>	<i>Validation Accuracy (%)</i>	<i>Loss</i>	<i>Validation Loss</i>
16	94,21	68,75	0,2141	5,9996
32	96,10	68,75	0,1310	3,0340
64	96,82	64,06	0,1104	4,0744

Nilai *batch size* terbaik yang diambil adalah 32 karena memiliki *validation loss* terkecil namun tidak berbeda jauh dari segi akurasi dan *loss* dengan *batch size* 64.

2. Pengujian Skenario *Optimizer*

Pengujian *optimizer* dilakukan dengan menyetarakan *hyperparameter* lain yaitu menggunakan *batch size* 32 (sesuai scenario *batch size*) dan *learning rate* 0,001 dengan iterasi *epoch* sebanyak 25.

TABEL 3
HASIL PENGUJIAN *OPTIMIZER*

<i>Optimizer</i>	<i>Accuracy (%)</i>	<i>Validation Accuracy (%)</i>	<i>Loss</i>	<i>Validation Loss</i>
RMSprop	95,88	71,88	0,1375	3,3398
Adam	94,78	75,00	0,1315	1,5506
SGD	81,79	78,12	0,5251	0,7817

Optimizer terbaik yang diambil adalah SGD karena tidak terlalu menunjukkan gejala *overfitting* serta SGD lebih seimbang dalam hasil *loss*.

Setelah mendapatkan *batch size* dan *optimizer*, kedua *hyperparameter* tersebut digunakan pada skenario *learning rate* ini dengan menggunakan iterasi *epoch* yang tetap sama yaitu 25.

3. Pengujian Skenario *Learning Rate*

TABEL 4
HASIL PENGUJIAN *LEARNING RATE*

<i>Learning Rate</i>	<i>Accuracy (%)</i>	<i>Validation Accuracy (%)</i>	<i>Loss</i>	<i>Validation Loss</i>
0,01	81,48	75,00	0,4870	0,8664
0,001	82,53	75,00	0,5137	0,8096
0,0001	68,81	71,88	0,8451	0,8744

Learning rate 0,001 dipilih menjadi *learning rate* terbaik dalam skenario ini karena memiliki akurasi terbaik serta memiliki nilai *loss*

pada validasi yang terkecil.

4. Pengujian Parameter Menggunakan Confusion Matrix

Dari ketiga skenario awal didapatkan konfigurasi *hyperparameter* untuk model terbaik yaitu *batch size* 32, *optimizer* SGD, dan *learning rate* 0,001. Model ini selanjutnya diuji dari *confusion matrix* yang dapat digunakan untuk mendapatkan nilai presisi, *recall*, dan *F1-score* dari model.



GAMBAR 2
CONFUSION MATRIX MODEL TERBAIK

Dari gambar tersebut, dapat ditemukan nilai *True Positive* (kondisi ketika nilai positif diprediksi positif) dari masing – masing kelas. Nilai tersebut bisa dilihat dari diagonal pojok kiri atas menuju pojok kanan bawah. Berikut tabel untuk menjabarkan nilai *true positive* masing – masing kelas.

TABEL 5
NILAI TRUE POSITIVE

Kelas	Jumlah data latih	<i>True Positive</i>
CMD	3545	90%
CBSD	1925	83%
CGM	1031	44%
CBB	622	22%
Healthy	422	41%

Dari data berupa *confusion matrix* yang memiliki nilai TP, nilai parameter lain seperti presisi, *recall*, dan *F1-score* dapat dihitung menggunakan persamaan yang terdapat pada subbab 3.3. Nilai presisi, *recall*, dan *F1-score* yang dihasilkan tiap kelas berbanding lurus dengan nilai TP masing – masing kelas. Meskipun memang ada beberapa nilai yang tidak demikian, tetapi secara garis besar pola tersebut terbukti dimana semakin besar nilai TP, maka semakin besar pula nilai ketiga parameter lainnya.

TABEL 6
HASIL PRESISI, RECALL, DAN F1-SCORE

Kelas	Jumlah data uji	Presisi	<i>Recall</i>	<i>F1-score</i>
CMD	886	0,82	0,90	0,86
CBSD	481	0,66	0,83	0,73
CGM	258	0,59	0,44	0,50
CBB	155	0,59	0,22	0,32
Healthy	105	0,67	0,41	0,51

C. Perbandingan Dengan Penelitian Sebelumnya

Pada bagian “Pendahuluan” disebutkan beberapa penelitian yang menggunakan CNN dalam mengklasifikasikan gambar. Disebutkan pula penelitian [4] yang melakukan klasifikasi terhadap penyakit singkong. Penelitian tersebut menjadi perbandingan utama karena banyaknya aspek yang hampir sama antara penelitian tersebut dengan penelitian ini. Dari segi hasil yang didapatkan juga terlihat tidak jauh berbeda sehingga dapat disimpulkan bahwa dengan data yang sama maka hasil tidak akan berbeda jauh.

TABEL 7
PERBANDINGAN DENGAN PENELITIAN SEBELUMNYA

Arsitektur	VGG16	MobileNetV2 (Penelitian sebelumnya)
Jumlah dataset	9430	9430
Jumlah kelas	5	5
Epochs	25	50
Training set	80%	70%
Validation set	20%	30%
Training accuracy	82,53%	79,68%
Training loss	0,5137	0,6166
Validation accuracy	75%	75,48%
Validation loss	0,8096	0,7021

V. KESIMPULAN

Klasifikasi penyakit singkong dengan menggunakan metode yang diusulkan yaitu *convolutional neural network* (CNN) dengan arsitektur VGG16 berhasil dilakukan dengan hasil yang cukup baik. Setelah melalui beberapa skenario pengujian, konfigurasi model terbaik didapatkan yaitu *batch size* 32, *optimizer* SGD, dan *learning rate* 0,001. Model tersebut memiliki performa yang lebih baik dibanding dengan konfigurasi lainnya. Model tersebut memperoleh akurasi *training* sebesar 82,53% dengan akurasi *validation* sebesar 75%.

Meskipun nilai akurasi tidak terlalu besar, namun performa model dari segi pelatihan dan validasi tidak jauh berbeda sehingga model tidak terlalu mengalami *overfitting*. Dari segi presisi, *recall*, dan *F1-score*, model mendapatkan performa yang berbeda tiap kelas. Masalah utama yang dapat menyebabkan hal tersebut adalah tidak seimbang jumlah data untuk masing – masing kelas.

REFERENSI

- [1] “Data Lima Tahun Terakhir,” *Kementerian Pertanian Republik Indonesia*, 2018. <https://www.pertanian.go.id/home/?show=page&act=view&id=61> (diakses Nov 06, 2021).
- [2] D. L. Poole dan A. K. Mackworth, *Artificial Intelligence Foundation of Computational Agents*, 1st Ed. Cambridge University Press, 2017.
- [3] A. Burkov, *The Hundred-Page Machine Learning Book*. Andriy Burkov, 2019.
- [4] R. Surya dan E. Gautama, “Cassava Leaf Disease Detection Using Convolutional Neural Networks,” *2020 6th Int. Conf. Sci. Inf. Technol. Embrac. Ind. 4.0 Towar. Innov. Disaster Manag. ICSITech 2020*, no. Cmd, hal. 97–102, 2020.
- [5] I. Z. Mukti dan D. Biswas, “Transfer Learning Based Plant Diseases Detection Using ResNet50,” *2019 4th Int. Conf. Electr. Inf. Commun. Technol. EICT 2019*, no. December, hal. 1–6, 2019.
- [6] D. I. Swasono, H. Tjandrasa, dan C. Fathicah, “Classification of tobacco leaf pests using VGG16 transfer learning,” *Proc. 2019 Int. Conf. Inf. Commun. Technol. Syst. ICTS 2019*, hal. 176–181, 2019.
- [7] A. Prabhakar, Neeti, dan R. Devi, “Different Color Detection in an RGB Image,” *Int. J. Dev. Res.*, vol. 7, no. 8, hal. 14503–14506, 2017.
- [8] M. Hussain, J. J. Bird, dan D. R. Faria, “A study on CNN transfer learning for image classification,” *Adv. Intell. Syst. Comput.*, vol. 840, hal. 191–202, 2019.
- [9] N. K. C. Pratiwi, Y. N. Fu’adah, dan E. Edwar, “Early Detection of Deforestation through Satellite Land Geospatial Images based on CNN Architecture,” *J. Infotel*, vol. 13, no. 2, hal. 54–62, 2021.
- [10] K. Simonyan dan A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, hal. 1–14, 2015.
- [11] K. Weiss, T. M. Khoshgoftaar, dan D. D. Wang, *A survey of transfer learning*, vol. 3, no. 1. Springer International Publishing, 2016.
- [12] E. Mwebaze, T. Gebru, A. Frome, S. Nsumba, dan J. Tusubira, “iCassava 2019 Fine-Grained Visual Categorization Challenge,” 2019.
- [13] M. Vakili, M. Ghamsari, dan M. Rezaei, “Performance Analysis and Comparison of Machine and Deep Learning Algorithms for IoT Data Classification,” 2020.