

Model Identifikasi Penyakit Pada Tumbuhan Padi Berbasis DenseNet

Muhammad Pailus^{1*}, Dthomas Hatta Fudholi², Syarif Hidayat³

^{1,2,3}Program Studi Informatika, Program Magister, Universitas Islam Indonesia, Yogyakarta

*email: 18917123@students.uui.ac.id

Abstract

Errors in identifying diseases in rice plants can cause the potential for crop failure to increase by 18-80%, according to data from the Indonesian Ministry of Agriculture. This could be due to the lack of expertise in agriculture when compared to the amount of land in Indonesia. Recent research in the field of deep learning using neural networks has achieved remarkable improvements. Research on the identification of plant diseases in rice plants, using the MobileNet, NasNet and SqueezeNet architecture that supports mobile devices has been carried out. The experimental results show that the proposed architecture can achieve an accuracy of 93.3%. Motivated by previous research, this research will use DenseNet architecture (Dense Convolutional Network) to detect diseases in rice plants. The dataset used is relatively small, between 100-200 photos for each disease. To cover the lack of dataset augmentation is done to the dataset. The final results obtained are quite satisfactory with an accuracy of 96% with a Weighted Average of 97%.

Keywords: Rice Plant Disease, DenseNet, Deep Learning

Abstrak

Kesalahan dalam mengidentifikasi penyakit pada tumbuhan padi, bisa menyebabkan potensi kegagalan panen meningkat hingga 18%-80%, menurut data kementerian pertanian Indonesia. Hal ini bisa disebabkan karena masih kurangnya ekspert di bidang pertanian bila dibandingkan dengan jumlah lahan yang ada di Indonesia. Penelitian terbaru dalam bidang deep learning menggunakan neural network telah mencapai peningkatan yang luar biasa. Penelitian tentang identifikasi penyakit tanaman pada tumbuhan padi, dengan menggunakan arsitektur MobileNet, NasNet and SqueezeNet yang support untuk perangkat mobile pernah dilakukan. Hasil eksperimen menunjukkan bahwa arsitektur yang diusulkan dapat mencapai akurasi sebesar 93,3%. Termotivasi oleh penelitian sebelumnya pada penelitian ini akan menggunakan arsitektur DenseNet (Dense Convolutional Network) untuk mendeteksi penyakit pada tumbuhan padi. Dataset yang digunakan tergolong sedikit yaitu antara 100-200 foto untuk setiap penyakitnya. Untuk menutupi kurangnya dataset dilakukan augmentasi terhadap dataset. Hasil akhir yang didapatkan cukup memuaskan dengan akurasi mencapai 96% dengan Weighted Average 97%.

Kata kunci: Penyakit padi, DenseNet, Deep Learning

1. PENDAHULUAN

Mengetahui penyakit tanaman serta melakukan diagnosis dengan cepat dan akurat guna menentukan obat yang akan digunakan sangat penting untuk meningkatkan produksi pangan. Cara yang paling umum digunakan saat ini menggunakan tenaga ahli di bidang pertanian, yang cenderung lambat yang tidak bisa mengakomodir dalam jumlah yang banyak, dan beresiko menurunkan produksi pertanian. Mendiagnosis penyakit tanaman melalui pengamatan secara optical pada gejala yang muncul pada daun tanaman, mempunyai tingkat kesulitan yang cukup kompleks [1].

Penyakit Tanaman merupakan ancaman utama terhadap ketahanan pangan, tetapi untuk melakukan identifikasinya dengan cepat di beberapa bagian di dunia masih sulit dilakukan, karena keterbatasan infrastruktur [2]. oleh sebab itu peran teknologi yang bisa diakses dengan biaya rendah perlu disiapkan, sehingga adanya pemerataan dalam hal ketersediaan informasi penyakit tanaman. Dengan banyaknya ketersediaan kamera digital dan kemajuan di bidang computer vision, model otomasi dalam mendiagnosis dengan presisi tinggi di bidang agriculture sangat dibutuhkan, seperti fenotipe tanaman, rumah kaca pintar, dan sebagainya [3]. Terlebih lagi saat ini smartphone yang dimiliki oleh masing-masing individu tidak terkecuali petani, sudah memiliki spesifikasi yang lebih dari cukup untuk melakukan tugas ini, dengan bantuan deep learning sebagai backend untuk melakukan prosesnya.

Deep learning merupakan teknik modern terbaru untuk pemrosesan gambar dan analisis data, dengan menjanjikan hasil dan potensi yang besar. Karena deep learning telah berhasil diterapkan di berbagai domain, baru-baru ini memasuki juga domain pertanian[4]. Deep learning pada dasarnya merupakan sub bagian dari machine learning. Machine learning adalah sebuah program komputer yang mampu belajar dari pengalaman, sehingga semakin banyak pengalaman yang diberikan kepada machine learning akan meningkatkan performanya menjadi semakin baik. Namun dalam prakteknya, algoritma machine learning tradisional tidak akan membaik secara signifikan ketika sudah sampai titik tertentu, sebanyak apapun data yang diberikan. Untuk itulah deep learning dibuat untuk menyelesaikan masalah performa pada algoritma machine learning tradisional ketika pertumbuhan data semakin membesar [5].

Penelitian yang pernah dilakukan untuk mendeteksi penyakit tanaman mendapatkan hasil yang cukup beragam. Klasifikasi untuk mendeteksi penyakit tanaman dengan algoritma support vector machine. dengan akurasi antara 65% sampai 90% tergantung dari tipe dan tingkat dari sugar beet disease [6].Metode pembelajaran mesin tradisional terutama menggunakan fitur visual tingkat rendah yang dikombinasikan dengan berbagai algoritma klasifikasi. Meskipun metode ini telah mencapai klasifikasi yang baik, tetapi mengalami beberapa keterbatasan. Kebanyakan dari keterbatasannya karena mengandalkan fitur campur tangan manusia, yang tidak dapat memecahkan masalah semantic gap [7].

Algoritma yang cukup terkenal dalam pengolahan citra gambar salah satunya adalah CNN. Sebagai model dari deep learning CNN dapat menemukan secara otomatis fitur yang mendalam dari sebuah data [8] dan sudah sukses di banyak domain yang beragam . Metode ini telah mencapai efek klasifikasi yang baik dalam klasifikasi penyakit tanaman. Namun, CNN dalam skala besar ini akan dengan cepat menjadi over fitted ke data pelatihan ketika jumlah gambar penyakit tanaman terbatas [7].

Menggunakan dataset plant village secara eksklusif [9], akurasi tinggi hanya dapat diperoleh dengan model pelatihan menggunakan sampel kondisi

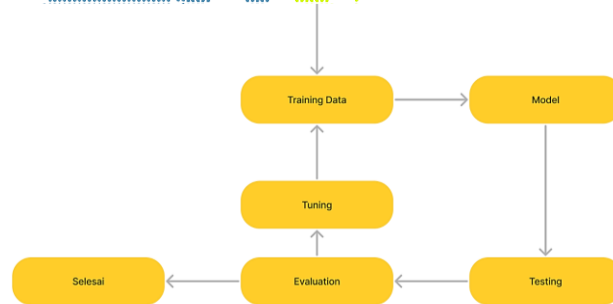
nyata. Deteksi infeksi tunggal atau multi infeksi pada daun tunggal maupun ganda merupakan tantangan tersendiri di bidang ini. Salah satu studi [10] mengusulkan dua model CNN, satu dilatih menggunakan gambar sampel daun penuh, dan yang kedua dilatih menggunakan sampel daun tersegmentasi yang mengandung gejala berbeda, dari dataset pelatihan yang sama dari model pertama. Temuan menunjukkan bahwa model kedua menunjukkan kinerja yang unggul dibandingkan model pertama berdasarkan pengukuran akurasi klasifikasi akhir dan Koefisien Dispersi Kuartil (QCoD) dari perbedaan kepercayaan antara model. Namun, itu tidak dapat mendeteksi beberapa infeksi untuk beberapa penyakit, dan perbaikan diperlukan pada tingkat segmentasi. Masalah lainnya termasuk deteksi objek kecil untuk kumpulan data gambar multi-label, deteksi hama, penentuan tingkat infeksi, identifikasi siklus hidup penyakit, dan gejala ringan [11]. Tantangan lain dalam domain ini adalah pengenalan buah untuk tujuan panen [12] mengusulkan arsitektur Yolo yang diubah, terdiri dari 11 lapisan, yang membagi gambar input menjadi dua kotak blok kecil untuk meningkatkan kinerja deteksi. Metode yang dikembangkan tidak dapat mendeteksi lebih dari satu objek dalam sel grid yang sama. Proses identifikasi dalam klasifikasi didasarkan pada bagaimana ROI di lokalisasi. Banyak teknik telah dibahas dalam kaitannya dengan masalah ini, termasuk segmentasi, deteksi objek dan metode hybrid yang memasok pengklasifikasi dengan informasi kontekstual yang terkait dengan ROI dan dengan demikian mempengaruhi kinerja mereka [7].

Sedangkan pada penelitian kali ini menggunakan DenseNet yang merupakan Arsitektur yang baru dari CNN [13], dimana fokusnya adalah untuk *Deep Learning Network* lebih mendalam akan tetapi pada saat bersamaan juga lebih efisien untuk di training, pada penelitian ini diharapkan dapat membuat models dengan akurasi yang baik diatas 90%

2. METODOLOGI PENELITIAN

Metodologi yang digunakan pada penelitian kali ini dimulai dari menggunakan dataset yang sudah tersedia [14] secara publik dan bebas digunakan. kemudian dilanjutkan dengan melakukan resize gambar, dan melakukan augmentasi pada dataset yang tersedia. Selanjutnya baru dilakukan proses training data. untuk lebih jelasnya pada gambar 1.





Gambar 1. Alur Metodologi Penelitian

2.1. Pengumpulan Data

Penelitian ini menggunakan dataset yang sudah tersedia pada tumbuhan padi dengan jumlah total data secara keseluruhan mencapai 1.444 data gambar dengan latar dan penyakit padi yang berbeda. Untuk menambah jumlah data training maka data di augmentasi, dengan terlebih dahulu melakukan resize pada data, sehingga mempunyai ukuran yang sama.

```

IMAGE_SIZE = 64

def read_image(filepath):
    return cv2.imread(os.path.join(data_dir, filepath)) # Loading a color image is the default flag
# Resize image to target size
def resize_image(image, image_size):
    return cv2.resize(image.copy(), image_size, interpolation=cv2.INTER_AREA)
    
```

Gambar 2. Proses melakukan resize image menjadi 64X64

Image yang sudah di resize kemudian ditampilkan untuk melihat apakah sudah berhasil sesuai atau belum.



Gambar 3. Dataset yang ukurannya sudah dilakukan resize

Setelah `resize` gambar berhasil maka gambar sudah siap untuk dilakukan augmentasi, untuk memperbanyak dataset untuk proses training.

2.2. Augmentasi Dataset

Untuk memperkaya dataset yang sudah ada, maka data dilakukan augmentasi, agar data yang didapatkan lebih banyak dan lebih beragam, sehingga nantinya diharapkan model lebih robust.

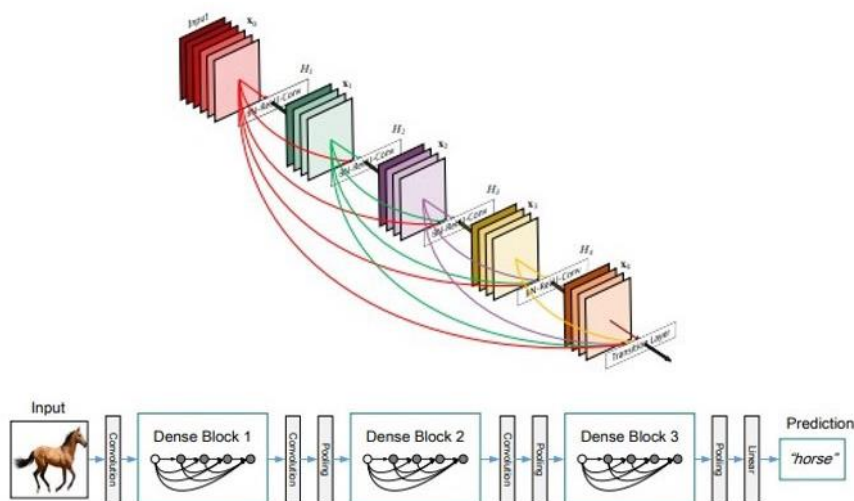
```
datagen = ImageDataGenerator(rotation_range=360, # Degree range for random rotations
                             width_shift_range=0.2, # Range for random horizontal shifts
                             height_shift_range=0.2, # Range for random vertical shifts
                             zoom_range=0.2, # Range for random zoom
                             horizontal_flip=True, # Randomly flip inputs horizontally
                             vertical_flip=True) # Randomly flip inputs vertically
```

Gambar 4. Augmentasi dataset

Setelah data dilakukan augmentasi maka dataset siap untuk di training dan masuk ke tahap selanjutnya.

2.3. Training Data

Untuk melakukan training data pada penelitian ini menggunakan DenseNet. DenseNet Sendiri dibuat untuk menyempurnakan CNN, dimana fokusnya adalah untuk *Deep Learning Network* lebih mendalam akan tetapi pada saat bersamaan juga lebih efisien untuk di training, dengan menggunakan koneksi yang pendek antara layernya. Berikut arsitektur dari DenseNet.



Gambar 5. Various blocks and layers in DenseNet (Source: Original DenseNet paper)

Penelitian ini menggunakan DenseNet 121 yang berjalan diatas platform keras, dengan konfigurasi Weight menggunakan **imagenet** dan *activation*-nya menggunakan **softmax**. dan optimizer dengan nilai **Adam(lr=0.002**,

beta_1=0.9, beta_2=0.999, epsilon=0.1, decay=0.0). perhatikan gambar 6 untuk melihat fungsi kode yang digunakan.

```
def build_densenet():
    densenet = DenseNet121(weights='imagenet', include_top=False)

    input = Input(shape=(SIZE, SIZE, N_ch))
    x = Conv2D(3, (3, 3), padding='same')(input)

    x = densenet(x)

    x = GlobalAveragePooling2D()(x)
    x = BatchNormalization()(x)
    x = Dropout(0.5)(x)
    x = Dense(256, activation='relu')(x)
    x = BatchNormalization()(x)
    x = Dropout(0.5)(x)

    # multi output
    output = Dense(3, activation='softmax', name='root')(x)

    # model
    model = Model(input, output)

    optimizer = Adam(lr=0.002, beta_1=0.9, beta_2=0.999, epsilon=0.1, decay=0.0)
    model.compile(loss='categorical_crossentropy', optimizer=optimizer, metrics=['accuracy'])
    model.summary()

    return model
```

Gambar 6. Fungsi untuk melakukan training data

2.4. Model

Model yang dihasilkan setelah proses training, disimpan dengan format **H5**. yang nanti jika diperlukan hanya perlu melakukan load model kemudian menjalankan fungsi predictor untuk memprediksi gambar dari tanaman padi untuk di identifikasi, jenis penyakitnya.

```
from tensorflow import keras
model.save('./models.h5')
```

Gambar 7. Menyimpan Model ke dalam Format **h5**

Hasil dari training didapat modal sebesar 128 MB, dalam format h5, semakin banyak dataset kemungkinan model yang dihasilkan juga akan semakin besar.

3. HASIL DAN PEMBAHASAN

Percobaan yang dilakukan pertama kali dengan menggunakan konfigurasi sebagai berikut, dengan pengujian menggunakan confusion matrix dan juga langsung testing terhadap gambar random yang ditemukan di internet, dengan tipe penyakit yang sama dengan yang ada di dataset, akan tetapi dengan tipe gambar yang jauh berbeda.

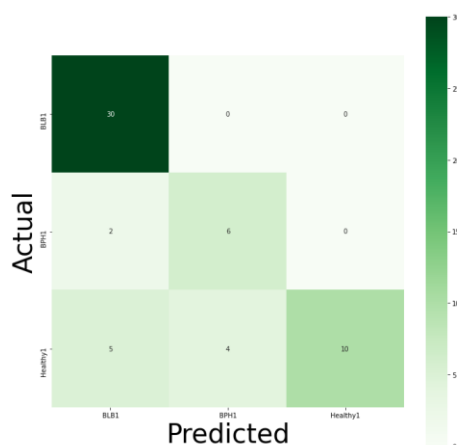
Tabel 1. Konfigurasi yang digunakan untuk training dataset.

EPOCHS	42
SIZE	64

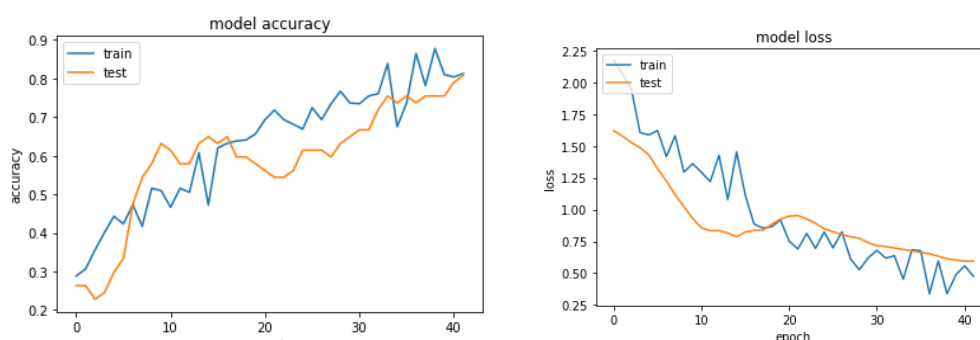
N_ch

3

Dengan hasil yang didapat kurang memuaskan dimana accuracy hanya **0.76** dengan rata-rata weighted average **00** dan hasil pengujian gambar seperti gambar 8.

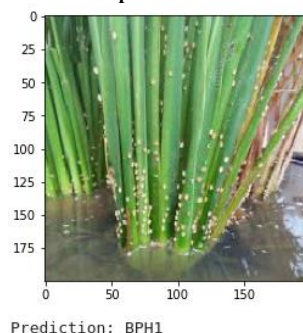


Gambar 8. Dengan Konfigurasi seperti tabel 1 hasil akurasi 0.80



Gambar 9. Grafik akurasi dan loss masih fluktuatif belum sesuai yang diharapkan.

Sedangkan pengujian pada gambar yang didapatkan secara random di internet dengan penyakit BPH didapatkan hasil berikut:



Gambar 10. Pengujian Pada gambar random di internet sudah sesuai

Hasil yang didapat dari pengujian masih dibawah 90% seperti klasifikasi laporan pada gambar 11.

Classification Report

	precision	recall	f1-score	support
BLB1	0.81	1.00	0.90	30
BPH1	0.60	0.75	0.67	8
Healthy1	1.00	0.53	0.69	19
accuracy			0.81	57
macro avg	0.80	0.76	0.75	57
weighted avg	0.84	0.81	0.79	57

Gambar 11. Hasil dari Klasifikasi report masih dibawah yang diharapkan.

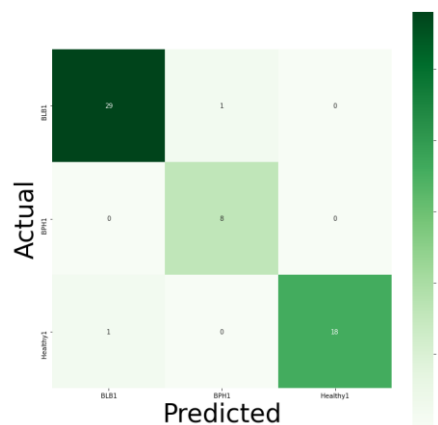
3.1. Tuning

Untuk meningkatkan hasil akurasi dari model, dengan cara mencoba mengubah satu parameter yang sebelumnya epoch **42** menjadi epoch **200** dan berikut hasil pengujian yang didapatkan cukup memuaskan mengacu pada gambar 12.

Tabel 2. Konfigurasi yang digunakan untuk training dataset dengan perubahan nilai epoch.

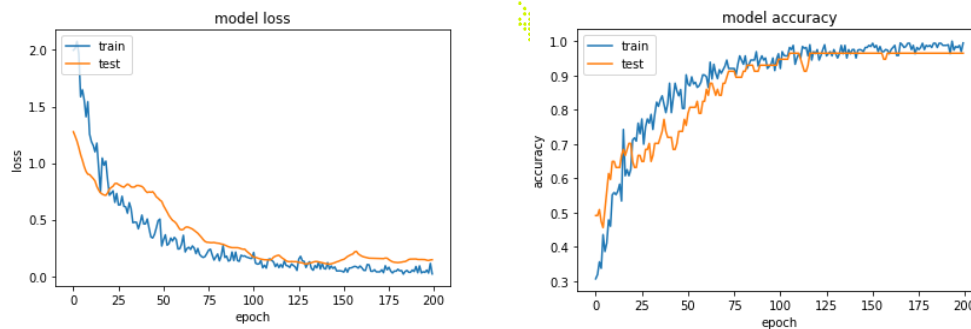
EPOCHS	200
SIZE	64
N_ch	3

Hasil pengujian pada data test didapatkan sebagai berikut pada gambar **b** heatmap



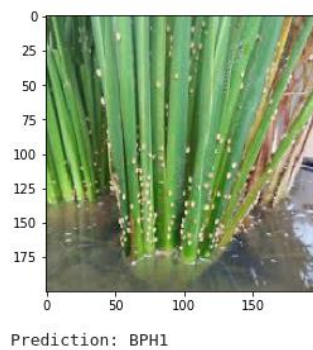
Gambar 12. Hasil pengujian dengan tuning

Sedangkan untuk hasil grafik akurasi yang didapat dengan perubahan epoch terlihat cukup memuaskan, akan tetapi hasil cenderung tetap ketika epoch mencapai angka **110**.



Gambar 13. Hasil merubah Epoch didapatkan hasil yang cukup memuaskan

Untuk pengujian dengan sampel gambar yang diambil dari internet didapatkan sebagai berikut.



Gambar 14. Dari penyakit BPH yang diambil dari internet secara random mendapatkan hasil sesuai dengan yang seharusnya.

Untuk report klasifikasi yang final didapatkan sebagai berikut.

Classification Report				
	precision	recall	f1-score	support
BLB1	0.97	0.97	0.97	30
BPH1	0.89	1.00	0.94	8
Healthy1	1.00	0.95	0.97	19
accuracy			0.96	57
macro avg	0.95	0.97	0.96	57
weighted avg	0.97	0.96	0.97	57

Gambar 15. Laporan Akhir Setelah Tuning

3.2. Diskusi

Proses identifikasi penyakit tanaman padi lebih awal sangat penting untuk dilakukan, karena dengan mengetahui jenis penyakit petani bisa memberikan pengendalian lebih awal yang bisa meningkatkan peluang untuk keberhasilan panen. Terbukti dengan data yang di *release* oleh kementerian pertanian jika bahwa penyakit tanaman bisa menyebabkan potensi gagal panen mencapai 18-80%.

Penelitian sebelumnya untuk mengidentifikasi dan mengenali penyakit pada tumbuhan padi pernah dilakukan dengan menggunakan arsitektur MobileNet, NasNet dan SqueezeNet, hasil yang didapatkan cukup bagus yaitu akurasi bisa mencapai 93%. Sedangkan pada penelitian ini akurasi dapat ditingkatkan kurang lebih 3% menjadi 96%, dengan cara melakukan tuning parameter. Selain itu proses augmentasi pada dataset juga mendukung akurasi pada model.

4. SIMPULAN

Dari eksperimen yang dilakukan pada penelitian ini dapat ditarik beberapa kesimpulan. mendeteksi penyakit tanaman dengan menggunakan DenseNet dengan ekstraksi fitur secara otomatis, didapatkan hasil yang cukup memuaskan dengan angka akurasi mencapai 96%. Dimana jumlah dataset sangat berpengaruh terhadap robust atau tidaknya model yang dihasilkan, disamping itu dengan mengubah tuning epoch ternyata didapatkan peningkatan akurasi yang sangat signifikan.

Penelitian ini hanya berfokus pada beberapa penyakit yang ada pada tumbuhan padi saja. nantinya diharapkan bisa dikembangkan pada domain tanaman yang berbeda. serta dapat meningkatkan akurasi dari model serta sensitifitasnya, sehingga akan didapatkan model untuk identifikasi penyakit tanaman yang lebih baik.

DAFTAR PUSTAKA

- [1] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Comput. Electron. Agric.*, vol. 145, pp. 311–318, 2018.
- [2] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Front. Plant Sci.*, vol. 7, p. 1419, 2016.
- [3] J. Wan *et al.*, "Deep learning for content-based image retrieval: A comprehensive study," in *Proceedings of the 22nd ACM international conference on Multimedia*, 2014, pp. 157–166.
- [4] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," *Comput. Electron. Agric.*, vol. 147, pp. 70–90, 2018.
- [5] J. Brownlee, *Deep learning for computer vision: image classification, object detection, and face recognition in python*. Machine Learning Mastery, 2019.
- [6] T. Rumpf, A.-K. Mahlein, U. Steiner, E.-C. Oerke, H.-W. Dehne, and L. Plümer, "Early detection and classification of plant diseases with support vector machines based on hyperspectral reflectance," *Comput. Electron. Agric.*, vol. 74, no. 1, Art. no. 1, 2010.
- [7] J. Wang, L. Chen, J. Zhang, Y. Yuan, M. Li, and W. Zeng, "CNN transfer learning for automatic image-based classification of crop disease," in *Chinese Conference on Image and Graphics Technologies*, 2018, pp. 319–329.
- [8] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no.

- 7553, Art. no. 7553, 2015.
- [9] S. S. Chouhan, U. P. Singh, A. Kaul, and S. Jain, "A data repository of leaf images: Practice towards plant conservation with plant pathology," in *2019 4th International Conference on Information Systems and Computer Networks (ISCON)*, 2019, pp. 700–707.
 - [10] P. Sharma, Y. P. S. Berwal, and W. Ghai, "Performance analysis of deep learning CNN models for disease detection in plants using image segmentation," *Inf. Process. Agric.*, vol. 7, no. 4, pp. 566–574, 2020.
 - [11] E. Fujita, H. Uga, S. Kagiwada, and H. Iyatomi, "A practical plant diagnosis system for field leaf images and feature visualization," *Int. J. Eng. Technol.*, vol. 7, no. 4.11, pp. 49–54, 2018.
 - [12] K. Bresilla, G. D. Perulli, A. Boini, B. Morandi, L. Corelli Grappadelli, and L. Manfrini, "Single-shot convolution neural networks for real-time fruit detection within the tree," *Front. Plant Sci.*, vol. 10, p. 611, 2019.
 - [13] G. Huang, S. Liu, L. Van der Maaten, and K. Q. Weinberger, "Condensenet: An efficient densenet using learned group convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2752–2761.
 - [14] C. Rafeed Rahman *et al.*, "Identification and Recognition of Rice Diseases and Pests Using Convolutional Neural Networks," *ArXiv E-Prints*, p. arXiv-1812, 2018.