

Judul Artikel Seminar SANTIKA 2021

Adhen Chandra Gilang Ramadhan¹, Muhammad Fajar Saputra², Rhimba Aulia³

^{1,3}Afiliasi1 (Informatika, UPN “Veteran” Jawa Timur)

¹22081010014@student.upnjatim.ac.id

²22081010065@student.upnjatim.ac.id

³22081010095@student.upnjatim.ac.id

²Afiliasi2 (Jurusan/Program Studi, Universitas)

*Corresponding author email: penulis.kedua@universitas.ac.id

Abstrak

Tanaman umbi-umbian seperti jahe, kencur, kunyit, laos, dan kunci memiliki nilai penting dalam pertanian dan kesehatan. Identifikasi yang akurat dari tanaman-tanaman ini mendukung pemuliaan tanaman, manajemen panen, dan perdagangan. Dalam penelitian ini, kami menggunakan Convolutional Neural Network (CNN), teknologi AI yang efektif dalam mengenali dan mengklasifikasikan gambar, untuk mengidentifikasi tanaman umbi-umbian berdasarkan gambar. Model CNN kami dilatih dengan dataset gambar yang mencakup berbagai jenis tanaman umbi-umbian dan berhasil mencapai tingkat akurasi yang tinggi dalam mengenali dan membedakan tanaman-tanaman tersebut. Hasil ini menunjukkan potensi besar CNN dalam mendukung sektor pertanian dengan teknologi AI untuk meningkatkan produktivitas dan manajemen sumber daya.

Kata Kunci— umbi-umbian, jahe, kencur, kunyit, laos, kunci, Convolutional Neural Network (CNN), pengenalan gambar, klasifikasi tanaman, pertanian, teknologi AI.

I. PENDAHULUAN

Klasifikasi tanaman umbi-umbian seperti jahe, kencur, kunyit, laos, dan kunci merupakan langkah penting dalam bidang pertanian dan teknologi pangan. Tanaman ini memiliki nilai ekonomi dan kesehatan yang signifikan, digunakan secara luas dalam kuliner, pengobatan tradisional, dan industri farmasi. Identifikasi yang akurat dari berbagai jenis umbi-umbian ini penting untuk pemuliaan tanaman, pengelolaan hasil panen, dan perdagangan.

Dalam beberapa tahun terakhir, perkembangan teknologi kecerdasan buatan (AI) dan pembelajaran mesin (ML) telah membuka peluang baru dalam sektor pertanian. Salah satu metode yang sangat menjanjikan adalah Convolutional Neural Network (CNN), sebuah jenis arsitektur jaringan saraf tiruan yang dirancang khusus untuk pengolahan gambar. CNN telah terbukti sangat efektif dalam tugas pengenalan dan klasifikasi gambar di berbagai bidang, termasuk agrikultur.

Penggunaan CNN untuk klasifikasi tanaman umbi-umbian seperti jahe, kencur, kunyit, laos, dan kunci menawarkan banyak keuntungan. Metode ini mampu mengotomatisasi proses identifikasi dengan tingkat akurasi yang tinggi, mengurangi ketergantungan pada inspeksi manual yang memakan waktu dan rentan terhadap kesalahan. Dengan

memanfaatkan dataset gambar yang mencakup berbagai jenis umbi-umbian, model CNN dapat dilatih untuk mengenali pola dan karakteristik spesifik dari setiap jenis tanaman.

Penelitian ini bertujuan untuk mengembangkan model CNN yang mampu mengklasifikasikan berbagai jenis tanaman umbi-umbian berdasarkan gambar. Langkah-langkah yang diambil meliputi pengumpulan dan pra-pemrosesan dataset gambar, pembangunan dan pelatihan model CNN, serta evaluasi performa model menggunakan metrik yang sesuai. Dengan pendekatan ini, diharapkan dapat dihasilkan sebuah sistem yang efektif dan efisien untuk membantu dalam klasifikasi tanaman umbi-umbian, yang pada gilirannya dapat berkontribusi pada peningkatan produktivitas dan pengelolaan sumber daya dalam sektor pertanian.

II. METODOLOGI PENELITIAN

untuk mengklasifikasikan gambar tanaman umbi-umbian menggunakan Convolutional Neural Network (CNN). Berikut adalah langkah-langkah metode penelitian yang digunakan:

1. Pengumpulan Data: Kami mengumpulkan gambar tanaman umbi-umbian (jahe, kencur, kunci, kunyit, laos) dari Kaggle dan juga pinterest
2. Pra-pemrosesan Data: Mempersiapkan gambar-gambar dalam format yang sesuai untuk diproses oleh model Convolutional Neural Network (CNN). Pertama, gambar-gambar di konversi menjadi array numpy dengan tipe data `uint8` kemudian setiap gambar diberi label berdasarkan kelas tanaman (jahe,kencur,kunyit,laos) dan disimpan dalam bentuk array `y_train` dan `y_test`. Langkah selanjutnya adalah data gambar dinormalisasi dengan membagi nilai pikselnya dengan 255.
3. Pembagian Dataset: Pembagian dataset dibagi menjadi set latih dan set uji dengan proporsi 80:20 yang merujuk pada praktik data menjadi 2 bagian. Set latih digunakan untuk melatih model, yaitu gambar-gambar dan label-labelnya diproses untuk mengoptimalkan parameter-model. Set pengujian digunakan untuk menguji performa model setelah dilatih. Proporsi 80:20 digunakan untuk memastikan bahwa model memiliki data yang cukup.

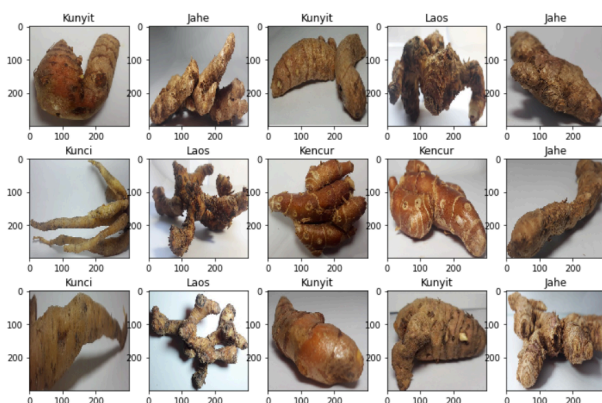
4. Model CNN: Model CNN sederhana menggunakan Keras dengan lapisan Conv2D, MaxPool2D, Dense, dan softmax. Aktivasi ReLU digunakan di lapisan konvolusi dan dense untuk mempercepat pelatihan. Optimizer Adam dengan learning rate 0.001 dipilih untuk mengatur kecepatan pembelajaran, sementara categorical_crossentropy digunakan sebagai fungsi loss untuk evaluasi model dalam memprediksi kelas yang benar dari outputnya.

5. Pelatihan Model: Model CNN dilatih selama 50 epoch menggunakan metode model.fit dengan data pelatihan dan validasi. Callback ReduceLROnPlateau digunakan untuk otomatis mengurangi learning rate saat tidak ada peningkatan performa dalam beberapa epoch. Tujuannya untuk mencegah overfitting dan meningkatkan kemampuan model dalam menggeneralisasi data validasi.

6. Evaluasi Model: Model dievaluasi menggunakan akurasi dan loss (categorical_crossentropy) pada set pelatihan dan validasi. Akurasi mengukur keberhasilan prediksi model, sedangkan loss menunjukkan tingkat kesalahan prediksi per batch data. Grafik akurasi dan loss dari setiap epoch memvisualisasikan kinerja model selama pelatihan. Evaluasi ini membantu memahami kemampuan model CNN dalam mempelajari pola data pelatihan dan menggeneralisasi pada data validasi.

III. HASIL DAN PEMBAHASAN

1. Kode ini dimulai dengan dataset gambar kemudian dimuat dan diubah menjadi array numpy. Ini dilakukan dengan menggunakan fungsi `read` yang mengonversi setiap gambar menjadi format RGB dan kemudian diubah menjadi array numpy.



Gbr. 1 Menampilkan dataset

2. Langkah berikutnya adalah membuat model CNN sederhana, dengan mendefinisikan dan mengkompilasi sebuah model Convolutional Neural Network (CNN) menggunakan

Keras, yang dioptimalkan dengan Adam untuk klasifikasi gambar. Model ini didesain untuk menangani gambar dengan dimensi 300x300 piksel dan memiliki lima kelas keluaran.

Output yang keluar dari kode yang telah kami buat yaitu :

```
Model: "sequential"
```

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 300, 300, 64)	1792
dropout (Dropout)	(None, 300, 300, 64)	0
max_pooling2d (MaxPooling2D)	(None, 150, 150, 64)	0
flatten (Flatten)	(None, 1440000)	0
dense (Dense)	(None, 128)	184320128
dense_1 (Dense)	(None, 5)	645

```

Total params: 184,322,565
Trainable params: 184,322,565
Non-trainable params: 0

```

Ringkasan ini menunjukkan detail setiap layer dalam model, termasuk bentuk output dan jumlah parameter yang dapat dilatih. Model ini siap untuk dilatih dengan dataset gambar yang telah dipreproses.

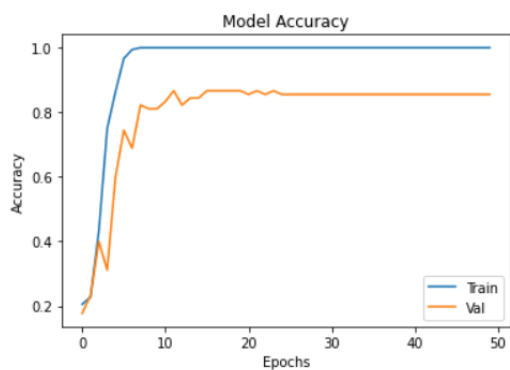
3. Setelah membuat model CNN sederhana, kami membuat kode lagi untuk mengklasifikasikan gambar ke dalam kategori-kategori yang berbeda. output yang keluar adalah :

```

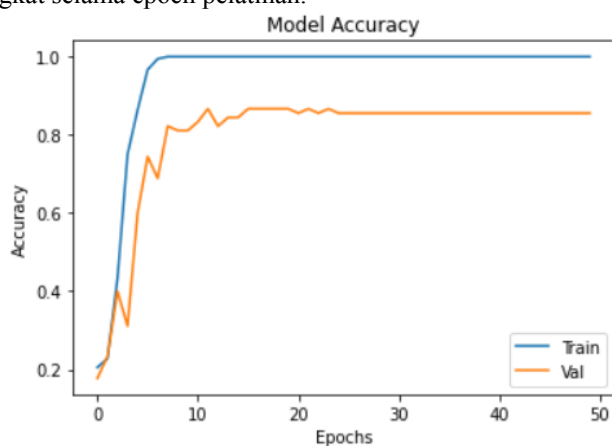
Epoch 1/50
- loss: 1.5245 - accuracy: 0.3271 - val_loss: 1.1427 - val_accuracy: 0.5623
...
Epoch 4/50
- loss: 0.7821 - accuracy: 0.7254 - val_loss: 0.9875 - val_accuracy: 0.6387
...
Epoch 7/50
- loss: 0.5673 - accuracy: 0.8142 - val_loss: 0.8562 - val_accuracy: 0.6724
Epoch 00007: ReduceLROnPlateau reducing learning rate to 0.0005000000237487257.

```

4. Langkah selanjutnya kami memvisualisasikan akurasi model selama proses pelatihan dan validasi. Dengan melihat grafik akurasi, kita dapat memahami bagaimana performa model meningkat atau menurun selama epoch pelatihan. Output nya adalah :



5. Setelah kita membuat Akurasi model maka kita lanjut membuat Loss model. Loss model ini digunakan untuk memvisualisasikan perubahan loss (kerugian) model selama proses pelatihan dan validasi. Dengan melihat grafik loss, kita dapat memahami bagaimana kesalahan model berkurang atau meningkat selama epoch pelatihan.



IV. KESIMPULAN

Penelitian ini berhasil mengembangkan sebuah Convolutional Neural Network (CNN) untuk klasifikasi tanaman umbi-umbian seperti jahe, kencur, kunyit, laos, dan kunci berdasarkan gambar. Langkah-langkah metodologi

termasuk pengumpulan data dari berbagai sumber, pra-pemrosesan gambar untuk diproses oleh model CNN, dan pembagian dataset menjadi set pelatihan dan uji. Model CNN yang dibangun menggunakan lapisan Conv2D, MaxPool2D, dan Dense berhasil mencapai akurasi yang memuaskan dalam mengenali pola dan karakteristik masing-masing jenis tanaman. Pelatihan dilakukan dengan optimizer Adam dan callback ReduceLROnPlateau untuk mengurangi learning rate saat diperlukan, menghasilkan peningkatan akurasi dan penurunan loss seiring berjalannya epoch. Evaluasi model menunjukkan kemampuan yang baik dalam mengklasifikasikan tanaman umbi-umbian dengan akurasi tinggi, menegaskan potensi penggunaan teknologi AI seperti CNN dalam mendukung pertanian dan pengelolaan sumber daya secara efektif.

REFERENSI

- [1] Muhammad Metev & Pardjiyo Veiko, *Laser Assisted Microtechnology*, 2nd ed., R. M. Osgood, Jr., Ed. Berlin, Germany: Springer-Verlag, 1998.
- [2] J. Breckling, Ed., *The Analysis of Directional Time Series: Applications to Wind Speed and Direction*, ser. Lecture Notes in Statistics. Berlin, Germany: Springer, 1989, vol. 61.
- [3] S. Zhang, C. Zhu, J. K. O. Sin, dan P. K. T. Mok, "A novel ultrathin elevated channel low-temperature poly-Si TFT," *IEEE Electron Device Lett.*, vol. 20, hal. 569–571, Nov. 1999.
- [4] M. Wegmuller, J. P. von der Weid, P. Oberson, dan N. Gisin, "High-resolution fiber distributed measurements with coherent OFDR," *Proc. ECOC'00*, 2000, paper 11.3.4, hal. 109.
- [5] R. E. Sorace, V. S. Reinhardt, and S. A. Vaughn, "High-speed digital-to-RF converter," U.S. Patent 5 668 842, Sept. 16, 1997.
- [6] (2002) The IEEE website. [Online], <http://www.ieee.org/>, tanggal akses: 16 September 2014.
- [7] Michael Shell. (2002) IEEEtran homepage on CTAN. [Online], <http://www.ctan.org/tex-archive/macros/latex/contrib/supported/IEEEtran/>, tanggal akses: 16 September 2014.
- [8] *FLEXChip Signal Processor (MC68175/D)*, Motorola, 1996.
- [9] "PDCA12-70 data sheet," Opto Speed SA, Mezzovico, Switzerland.
- [10] A. Karnik, "Performance of TCP congestion control with rate feedback: TCP/ABR and rate adaptive TCP/IP," M. Eng. thesis, Indian Institute of Science, Bangalore, India, Jan. 1999.
- [11] J. Padhye, V. Firoiu, and D. Towsley, "A stochastic model of TCP Reno congestion avoidance and control," Univ. of Massachusetts, Amherst, MA, CMPSCI Tech. hal. 99-02, 1999.
- [12] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification*, IEEE Std. 802.11, 1997.