

6. Implementasi Deep Learning Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan Convolutional Neural Network

by Agung Nilogiri

Submission date: 05-Feb-2022 12:30PM (UTC+0800)

Submission ID: 1755329047

File name: asarkan_Citra_Daun_Menggunakan_Convolutional_Neural_Network.docx (191.01K)

Word count: 2627

Character count: 15930

Implementasi Deep Learning Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan Convolutional Neural Network

Sarirotul Ilahiyah¹⁾, Agung Nilogiri²⁾

^{1,2)}Jurusan Teknik Informatika, Fakultas Teknik, Universitas Muhammadiyah Jember
Email : ¹⁾ ila.ilahiyah@gmail.com, ²⁾ agungnilogiri@unmuhjember.ac.id

ABSTRAK

The Convolutional Neural Network (CNN) is a Deep Learning technique that is based on the Multilayer Peceptron (MLP), which is meant to analyze two-dimensional data like images or sound. CNN is built on the notion of translation invariance, which allows it to recognize objects in an image in a wide range of positions. Alexnet has classified a total of 2000 leaf photos. Alexnet is a CNN architecture created by Krizhevsky that contains eight feature extraction layers. There are five convolution layers and three pooling layers in this layer. Alexnet features two Fully Connected layers in its classification layer, each with 4096 neurons. Pada akhir layer terdapat pengklasifikasian kedalam 20 kategori menggunakan aktifasi *softmax*. Rata-rata akurasi dari hasil klasifikasi mencapai 85%. Sedangkan akurasi dari identifikasi berhasil mencapai 90% yang didapatkan dari pengujian 40 citra.

Kata Kunci : *Deep Learning, Convolutional Neural Network, Alexnet.*

1. PENDAHULUAN

Data besar, juga dikenal sebagai Big Data, memiliki kualitas yang sangat besar, bervariasi, dan berkembang dengan cepat. Itu juga bisa tidak terstruktur, sehingga sulit untuk dianalisis menggunakan metode tradisional. Kelemahan lain dalam teknik tradisional adalah hanya dapat memproses sejumlah kelas target yang terbatas dan hanya dapat mengenali item di tengah gambar atau tidak dapat mengenali hal-hal di berbagai kemungkinan tempat dalam gambar (translation invariance). Keterbatasan lain, seperti rotasi objek dan perbedaan ukuran, ada di samping varians dalam penempatan objek (penskalaan). Pembelajaran mesin juga dapat mempelajari prinsip invarian terjemahan, meskipun memerlukan parameter yang jauh lebih banyak daripada pembelajaran mendalam, yang didasarkan pada prinsip ini. (built-in) (Putra, 2018).

Deep Learning adalah disiplin pembelajaran mesin yang didasarkan pada Jaringan Saraf Tiruan (JST) atau dapat dianggap sebagai perkembangan dari JST. Komputer belajar mengklasifikasikan gambar atau suara secara langsung menggunakan *deep learning*. *Convolutional Neural Network (CNN/ConvNet)* adalah teknik pembelajaran mendalam yang didasarkan pada *Multilayer Perceptron (MPL)*, yang dimaksudkan untuk menafsirkan data dua dimensi seperti gambar atau suara. CNN dapat belajar langsung dari gambar, mengurangi jumlah pemrograman yang diperlukan.

Selanjutnya, karena algoritma pembelajaran mesin tradisional hanya mengandalkan CPU dan RAM untuk komputasi, kinerja komputasi ditentukan oleh karakteristik ini. Sementara pendekatan pembelajaran mendalam memanfaatkan kemampuan GPU selain CPU dan RAM dalam proses komputasi, prosedur komputasi data besar dapat

diselesaikan lebih cepat.

Dengan menggunakan CNN, dataset citra daun dari Neeraj Kumar yang mempunyai 20 jenis genus tumbuhan dan masing-masing genus memiliki seratus citra daun beresolusi tinggi dapat diidentifikasi lebih mudah. Kemudahan ini karena CNN dapat mengenali objek dalam citra pada berbagai macam posisi yang mungkin (*translation invariance*).

2. TINJAUAN PUSTAKA

2.1 Convolutional Neural Network

Convolutional Neural Networks (CNN / ConvNet) adalah jenis teknik pembelajaran mendalam yang berkembang dari *Multilayer Perceptrons (MLPs)*. MLP digunakan untuk menangani data dalam dua dimensi, seperti gambar atau suara. Dengan menggunakan metode *supervised learning*, CNN digunakan untuk mengklasifikasikan data berlabel. Cara kerja *supervised learning* adalah adanya data latih dan variabel sasaran, dengan tujuan mengelompokkan data ke dalam data yang sudah ada.

Lapisan CNN termasuk neuron yang tersusun dalam tiga dimensi (lebar, tinggi, kedalaman). Parameter lebar dan tinggi menunjukkan ukuran lapisan, sedangkan parameter kedalaman menunjukkan jumlah lapisan. CNN mungkin berisi puluhan hingga ratusan lapisan, yang masing-masing belajar mengenali gambar yang berbeda. Setiap gambar pelatihan dikenakan pemrosesan gambar pada resolusi yang berbeda, dan output dari setiap gambar diproses dan digunakan sebagai input untuk lapisan berikutnya. Pemrosesan gambar adalah fitur yang sangat berguna yang dapat dimulai.

semudah kecerahan dan batas, atau serumit karakteristik yang mencirikan objek secara unik berdasarkan ketebalan lapisan (Mathworks, 2017).

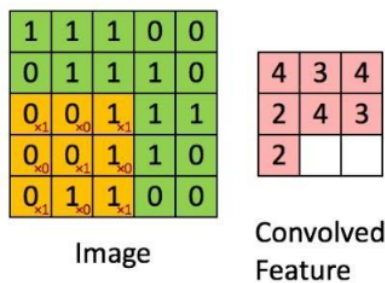
Secara umum, ada dua jenis lapisan di CNN. Lapisan pertama adalah lapisan ekstraksi fitur, yang diposisikan di bagian atas arsitektur. Arsitektur dibangun dari beberapa level, yang masing-masing terdiri dari neuron yang terhubung ke wilayah lokal lapisan sebelumnya. Jenis layer pertama adalah *convolutional*, sedangkan tipe kedua adalah *pooling*. Setiap lapisan diberi fungsi aktivasi, berputar antara jenis pertama dan

kedua. Lapisan ini menerima input gambar secara langsung dan memprosesnya untuk memberikan output vektor untuk lapisan berikutnya. Lapisan kedua adalah lapisan kategorisasi, yang terdiri dari banyak lapisan, yang masing-masing terdiri dari neuron yang sepenuhnya digabungkan ke neuron di tingkat sebelumnya. Layer ini mendapatkan input berupa vektor dari output layer ekstraksi fitur gambar dan kemudian mentransformasikannya menjadi Multi Neural Network dengan menambahkan banyak hidden layer. Hasil keluaran berupa akurasi klasifikasi suatu kelas.

2.2 Operasi Konvolusi

Konvolusi didefinisikan sebagai proses untuk memperoleh suatu piksel didasarkan pada nilai piksel itu sendiri dan tetangganya dengan melibatkan suatu matriks yang disebut kernel yang merepresentasikan

pembobotan (Kusumanto et al. 2011). Operasi ini menerapkan fungsi output sebagai Feature Map dari masukan citra. Masukan dan keluaran ini dapat dilihat sebagai dua argumen bernilai riil (Goodfellow et al 2016). Operasi konvolusi ini ditunjukkan dalam Gambar 1.

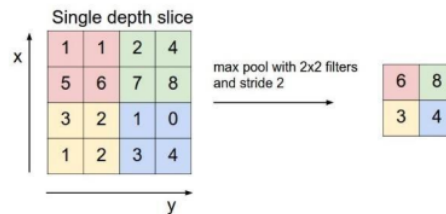


Gambar 1. Operasi Konvolusi

2.3 Pooling Layer

2.4 Pooling Layer mengambil fungsi input dan Feature Map dan melakukan berbagai operasi statistik berdasarkan nilai piksel terdekat. Lapisan Pooling biasanya diperkenalkan setelah beberapa lapisan konvolusi dalam model CNN. Pooling layer yang

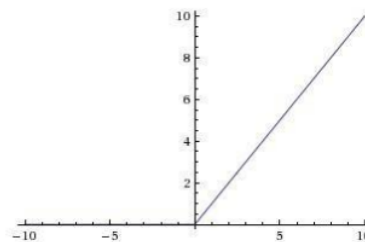
diimplementasikan antara convolution layer berturut-turut dalam arsitektur model CNN dapat secara bertahap mengurangi volume output pada Feature Map, mengurangi jumlah parameter dan komputasi dalam jaringan dan memungkinkan kontrol overfitting. Lapisan Pooling mengoptimalkan ukuran setiap tumpukan Peta Fitur. Metode penyatuan lapisan yang paling sering digunakan adalah menerapkan filter 2x2 dalam langkah dua dan kemudian beroperasi pada setiap irisan input. Bentuk seperti itu akan mengecilkan Feature Map hingga 75% dari ukuran aslinya. (2018) (Priyono). Gambar 2 mengilustrasikan proses Max Pooling.



Gambar 2. Contoh Operasi Max Pooling

2.5 Aktivasi ReLu

Lapisan aktivasi ReLU (Rectified Linear Unit) pada model CNN menggunakan fungsi $f(x)=\max(0,x)$, yang menunjukkan bahwa fungsi ini menerapkan threshold bernilai nol pada nilai piksel pada citra input. Aktivasi ini menetapkan semua nilai piksel dalam gambar yang kurang dari nol hingga 0..



Gambar 3. Aktivasi ReLu

2.6 Fully-Connected Layer

Lapisan ini biasanya digunakan dalam aplikasi MLP dan bertanggung jawab untuk mengubah dimensi data sehingga dapat

dikategorikan secara linier. Sebelum setiap neuron di lapisan konvolusi dapat dimasukkan ke dalam lapisan yang sepenuhnya terhubung, itu harus diubah menjadi data satu dimensi. Karena fakta

bahwa ini mengakibatkan hilangnya informasi spasial dan tidak dapat diubah, lapisan yang terhubung penuh hanya dapat diimplementasikan di ujung jaringan.

2.7 Aktivasi Softmax

Aktivasi Softmax atau Softmax Classifier adalah varian lain dari pendekatan Logistic Regression yang mampu mengklasifikasikan lebih dari dua kelas. Kriteria klasifikasi yang sering digunakan oleh algoritma Regresi Logistik adalah tugas untuk kategorisasi kelas biner. Persamaan berikut muncul di Softmax.

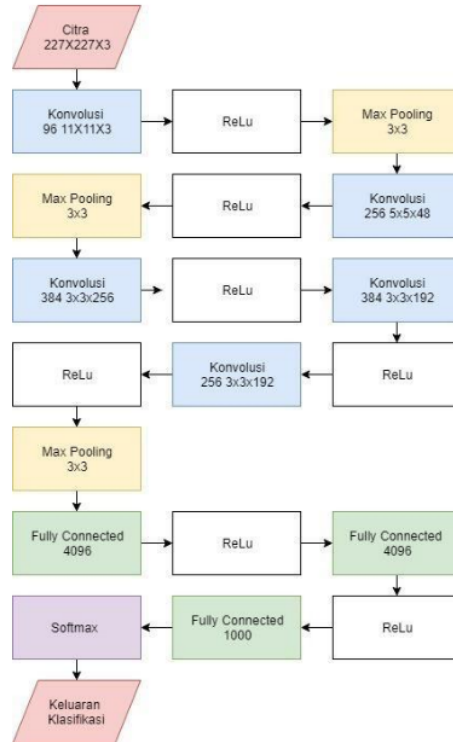
$$f_j(Z) = \frac{e^{z_j}}{\sum_k e^{z_k}} \quad (1)$$

Notasi f_j menunjukkan bahwa fungsi mengembalikan nilai untuk elemen ke- j dalam vektor keluaran kelas. Parameter z menentukan hipotesis yang disediakan model pelatihan agar fungsi Softmax mengklasifikasikannya. Selain itu, Softmax menghasilkan temuan yang lebih intuitif dan memiliki interpretasi yang lebih probabilistik dibandingkan metode klasifikasi lainnya. Softmax memungkinkan perhitungan probabilitas untuk label apa pun. Sebuah vektor nilai nyata akan diekstraksi dari label yang ada dan diubah menjadi vektor nilai antara nol dan satu, yang jika disatukan akan sama dengan satu.

3. METODE PENELITIAN

Pada penelitian implementasi *deep learning* pada identifikasi jenis tumbuhan berdasarkan citra daun menggunakan *Convolutional Neural Network* (CNN) ini digunakan arsitektur CNN dari Krizhevsky et al.(2012) yang disebut dengan AlexNet.

Secara garis besar, cara kerja sistem arsitektur Alexnet dibagi menjadi dua kelompok *layer*. Pertama adalah *layer* ekstraksi fitur yang tersusun dari *layer* konvolusi dan *pooling layer*, dan kedua adalah *layer* klasifikasi.



KEMER... SEBANYAK 30...
dengan *stride* 4 dan *padding* 0. Lapisan konvolusi yang berukuran 11x11x3 sebanyak 96 ini pada awal jaringan menangkap fitur citra dasar, seperti tepian dan gumpalan. Keluaran

dari layer ini berukuran 55x55x96 yang dapat dihitung menggunakan rumus Dimensi Keluaran seperti berikut :

$$\begin{aligned} \text{[? ? ?]} &= \left[\frac{\text{[? ? ?]} + 2 \text{[?]} - \text{[?]}}{227 + 2 \cdot 0 - 11} \right] + 1 \\ \text{[? ? ?]} &= \left[\frac{4}{55} \right] + 1 = \end{aligned}$$

Untuk menghitung parameter dari layer ini dapat dihitung dengan cara :

$$\begin{aligned} 55 \times 55 \times 96 &= 290400 \text{ neuron} \\ \text{Masing-masing memiliki } 11 \times 11 \times 3 &= 363 \text{ bobot} + 1 \text{ bias} \\ \text{Total parameter layer ini} &= 290400 \\ &\times 364 = 105,705,600 \text{ parameter} \end{aligned}$$

3) ReLU

ReLU merupakan lapisan aktivasi pada model CNN yang mengaplikasikan fungsi $f(x) = \max(0, x)$ yang berarti fungsi ini melakukan *thresholding* dengan nilai nol terhadap nilai piksel pada input citra.

4) Max pooling : 3x3 stride [2 2] dan padding [0 0]

Pooling layer digunakan untuk mengurangi ukuran dan mempercepat perhitungan, serta membuat beberapa fitur yang diperkirakan sedikit lebih akurat (Priyono 2018). Dimensi keluaran dari layer ini berukuran 27x27x96 yang didapatkan dari :

$$\text{[? ? ?]} = \left[\frac{55 + 2 \cdot 0 - 3}{2} \right] + 1 = 27$$

5) Convolution : 256 5x5x48 dengan stride [1 1] dan padding [2 2]

Proses konvolusi dengan ukuran kernel 5x5x48 sebanyak 256, dengan *stride* 1 dan *padding* 2. Layer konvolusi kedua yang berukuran 5x5x48 sebanyak 256 *mask* merespon sudut dan konjungsi tepi warna Keluaran dari layer ini berukuran 27x27x256 yang dapat dihitung menggunakan rumus Dimensi Keluaran seperti berikut :

$$\text{[? ? ?]} = \left[\frac{27 + 2 \cdot 2 - 5}{1} \right] + 1 = 27$$

6) ReLU

7) Max Pooling : 3x3 dengan stride [2 2] dan padding [0 0]

Dimensi keluaran dari layer ini berukuran 13x13x256 yang didapatkan dari :

$$\text{[? ? ?]} = \left[\frac{27 + 2 \cdot 0 - 3}{2} \right] + 1 = 13$$

8) Convolution : 384 3x3x256 dengan stride [1 1] dan padding [1 1]

Proses konvolusi dengan ukuran kernel 3x3x256 sebanyak 384, dengan *stride* 1 dan *padding* 1. Layer konvolusi ketiga yang memiliki ukuran kernel 3x3x256 menangkap tekstur. Keluaran dari layer ini berukuran 13x13x384 yang dapat dihitung menggunakan rumus Dimensi Keluaran seperti berikut :

$$\text{[? ? ?]} = \left[\frac{13 + 2 \cdot 1 - 3}{1} \right] + 1 = 13$$

9) ReLU

10) Convolution : 384 3x3x192 dengan stride [1 1] dan padding [1 1]

Proses konvolusi dengan ukuran kernel 3x3x192 sebanyak 384, dengan *stride* 1 dan *padding* 1. Layer konvolusi keempat dengan ukuran kernel 3x3x192 menunjukkan menunjukkan bagian-bagian objek Keluaran dari layer ini berukuran 13x13x384 yang dapat dihitung menggunakan rumus Dimensi Keluaran seperti berikut :

$$\text{[? ? ?]} = \left[\frac{13 + 2 \cdot 1 - 3}{13 \ 1} \right] + 1 =$$

11) ReLU

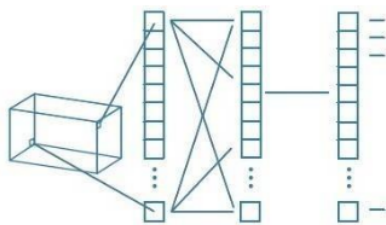
- 12) *Convolution* : 256 3x3x192 dengan *stride* [1 1] dan *padding* [1 1]
 Proses konvolusi dengan ukuran kernel 3x3x192 sebanyak 256, dengan *stride* 1 dan *padding* 1. Layer konvolusi kelima dengan ukuran kernel 3x3x192 menunjukkan seluruh objek. Keluaran dari layer ini berukuran 13x13x384 yang dapat dihitung menggunakan rumus Dimensi Keluaran seperti berikut :

$$[\begin{matrix} \square & \square & \square \\ \square & \square & \square \\ \square & \square & \square \end{matrix}] = [\frac{13 + 2 \cdot 1 - 3}{1}] + 1 = 13$$

- 13) ReLU
 14) *Max Pooling* : 3x3 dengan *stride* [2 2] dan *padding* [0 0]
 Dimensi keluaran dari layer ini berukuran 6x6x256 yang didapatkan dari :

$$[\begin{matrix} \square & \square & \square \\ \square & \square & \square \\ \square & \square & \square \end{matrix}] = [\frac{13 + 2 \cdot 0 - 3}{2}] + 1 = 6$$

- 15) *Fully Connected* : 4096 layer
 Untuk dapat diproses dalam *Fully Connected Layer*, masukan yang berukuran 6x6x256 harus di *reshape* menjadi ukuran yang kedalamannya hanya sebanyak satu layer. Maka, ukuran yang akan diproses pada layer ini adalah 9216x1 yang didapatkan dari perkalian 6x6x256.



Gambar 5. Fully Connected Layer

- 16) ReLU

- 17) *Fully Connected* : 4096 layer
 18) ReLU
 19) *Fully Connected* : 1000 layer
 20) *Softmax*

Aktifasi *Softmax* atau *Softmax Classifier* merupakan aktifasi yang sering dipakai untuk menangani kasus *multiclass classification*, karena *output layer* biasanya memiliki lebih dari satu *neuron*.

- 21) Keluaran klasifikasi

4. HASIL DAN PEMBAHASAN

Validasi silang digunakan untuk memvalidasi algoritma pada langkah ini. Validasi silang adalah teknik statistik untuk menganalisis dan membandingkan algoritma pembelajaran yang membagi data menjadi dua segmen: satu untuk pembelajaran atau pelatihan dan satu lagi untuk memvalidasi model. Validasi silang memerlukan persilangan berturut-turut dari set pelatihan dan validasi untuk memastikan bahwa setiap bagian data memiliki peluang untuk divalidasi.

Tes ini mempartisi dataset menjadi 10 folder, yang masing-masing berisi 2000 foto yang dibagi secara acak menjadi gambar pengujian dan pelatihan. Setiap folder berisi 1800 foto latihan dan 200 gambar uji. Setiap folder dicoba tiga kali untuk menentukan akurasi. Tabel 1 memuat hasil pengujian yang diperoleh dengan K-Fold Cross Validation.

Lipat 10 mencapai rata-rata terbesar di tiga uji coba, dengan peringkat akurasi 88,6 persen. Sedangkan Fold 4 memiliki rata-rata nilai akurasi terendah yaitu 83 persen. Presisi maksimum dicapai dengan Fold 7, yang mencapai tingkat akurasi 90 persen pada percobaan pertama dari tiga percobaan. Akurasi sistem seperti yang ditentukan oleh metode K-Fold

Cross Validation sebesar 85 persen yang didapat dari hasil rata-rata seluruh akurasi *Fold* pada setiap percobaan. *Fold* 10, sebagai *fold* yang mempunyai nilai akurasi rata-rata tertinggi digunakan sebagai data latih pada percobaan sekenario berikutnya.

Tabel 1. Akurasi Hasil Percobaan dengan Metode *K-Fold Cross Validation*

Fold	Percobaa n-1 (%)	Percobaa n-2 (%)	Percobaa n-3 (%)	Rata-rata (%)
1	84,5	84,5	83,5	84,1
2	85,5	84	86	85,1
3	86	85,5	84,5	85,3
4	84	83	82	83
5	83	86	84,5	84,5
6	85,5	86,5	89	87
7	90	82,5	85	85,8
8	84,5	86,5	82	84,3
9	83	84,5	85	84,1
10	89,5	89	87,5	88,6
Rata-rata	85,55	85,2	84,9	85,2

Pada percobaan kedua, citra uji menggunakan citra baru yang tidak digunakan pada sistem. Jumlah citra yang digunakan pada percobaan kedua ini sebanyak 40 citra yang mewakili 20 kategori genus. Jadi, tiap-tiap genus mempunyai dua citra yang akan diuji. Hasil percobaan kedua ini dapat dilihat pada Tabel 2.

Tabel 2. Hasil Percobaan dengan Citra Uji Baru

Percobaan	Prediksi Benar	Prediksi Salah	Akurasi
1	36	4	90 %
2	38	2	95 %
3	35	5	87.5%
Rerata			90,8 %

Hasil pengujian 40 citra baru yang dilakukan sebanyak tiga kali percobaan

mendapatkan nilai akurasi yang cukup tinggi. Percobaan pertama mendapatkan prediksi benar sebanyak 36 citra dari 40 citra uji sehingga mendapatkan nilai akurasi sebesar 90 persen. Pada percobaan kedua hanya didapati dua prediksi salah dari 40 citra uji dan menghasilkan nilai akurasi 95 persen. Sedangkan pada percobaan ketiga menghasilkan nilai akurasi sebesar 87 persen. Dari ketiga percobaan tersebut, didapatkan nilai akurasi sistem sebesar 90,8 persen.

5. KESIMPULAN DAN SARAN

Dari hasil pengamatan selama perancangan, implementasi, dan proses uji coba sistem, dapat diambil kesimpulan yaitu : (1) Percobaan menggunakan *fold cross validation* dengan nilai $k=10$ didapatkan presisi tertinggi pada percobaan pertama dari *fold* ke 7 yakni dengan nilai akurasi sebesar 90%, (2) Tingkat rata-rata akurasi klasifikasi sistem yang didapatkan dari percobaan *fold cross validation* dengan nilai $k=10$ yakni sebesar 85,21%, (3) Sistem yang telah dibuat dapat mengidentifikasi jenis genus tumbuhan dengan nilai akurasi sistem sebesar 90,8%.

DAFTAR PUSTAKA

- Ghifary, Muhammad. 2015. Deep Convolutional Neural Network. Diakses Maret 2, 2018. <https://ghifar.wordpress.com/2015/07/21/deep-convolutional-neural-networks-part-1/>.
- Gonzalez, Rafael C, and Richard E Woods. 2008. Digital Image Processing Third Edition. 3rd ed. New Jersey: Pearson Prentice Hall.
- Goodfellow et al. 2016. Deep Learning. Diakses Maret 1, 2018. <http://www.deeplearningbook.org/>.

- Kadir, Abdul, and Adhi Susanto. 2013. *Teori Dan Aplikasi Pengolahan Citra Digital*. Yogyakarta: Andi.
- Krizhevsky, Alex, Ilya Sutskever, and Hinton Geoffrey E. 2012. "ImageNet Classification with Deep Convolutional Neural Networks." *Advances in Neural Information Processing Systems* 25 (NIPS2012), 1–9. <https://doi.org/10.1109/5.726791>.
- 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.
- Kusumanto, R D, Alan Novi Tomponu, Setyo Pambudi, Jurusan Teknik Komputer, and Politeknik Negeri Sriwijaya. 2011. "Klasifikasi Warna Menggunakan Pengolahan Model Warna HSV" 2 (2): 83–87.
- Lecun, Yann, Yoshua Bengio, and Geoffrey Hinton. 2015. "Deep Learning" 521. <https://doi.org/10.1038/nature14539>.
- Prasetyo, Eko. 2011. *Pengolahan Citra Digital dan Aplikasinya Menggunakan Matlab*. Yogyakarta: Andi.
- Prijono, Benny. 2018. *Convolutional Neural Networks (CNN) Introduction*. 7 Maret. Diakses Maret 29, 2018. <https://indoml.com/2018/03/07/student-notes-convolutional-neural-networks-cnn-introduction/>.
- Putra, Jan Wira Gotama. 2018. *Pengenalan Konsep Pembelajaran Mesin Dan Deep Learning*. 1.0. Tokyo: Tokyo Institute of Technology.
- Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting." *Journal of Machine Learning Research* 15: 1929–58. <https://doi.org/10.1214/12-AOS1000>.
- Wei, Donglai, Bolei Zhou, Antonio Torralba, and William Freeman. n.d. "Understanding Intra-Class Knowledge Inside CNN" 6 (2): 6–12. http://vision03.csail.mit.edu/cnn_art/.
- Zeiler, Matthew D, and Rob Fergus. 2014. "Visualizing and Understanding Convolutional Networks" 1: 818–33. http://vision03.csail.mit.edu/cnn_art/.

6. Implementasi Deep Learning Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan Convolutional Neural Network

ORIGINALITY REPORT

20%

SIMILARITY INDEX

17%

INTERNET SOURCES

8%

PUBLICATIONS

11%

STUDENT PAPERS

MATCH ALL SOURCES (ONLY SELECTED SOURCE PRINTED)

5%

★ Submitted to Culver-Stockton College

Student Paper

Exclude quotes On

Exclude matches Off

Exclude bibliography On