

Implementation of Deep Learning for Classification Type of Orange Using The Method Convolutional Neural Network

Implementasi *Deep Learning* untuk Klasifikasi Jenis Jeruk dengan Menggunakan Metode *Convolutional Neural Network*

Irvan Denata¹, Tedy Rismawan², Ikhwan Ruslianto³

^{1,2,3}Jurusan Rekayasa Sistem Komputer, Fakultas MIPA Universitas Tanjungpura

¹irvandenata@student.untan.ac.id, ^{2*}tedyrismawan@siskom.untan.ac.id,

^{3*}ikhwanruslianto@siskom.untan.ac.id

Informasi Artikel

Received: 10 July 2021

Revised: 15 August 2021

Accepted: 30 August 2021

Published: 31 October 2021

Abstract

Orange is a type of fruit that is easily found in Sambas Regency. The types that are widely sold are Siam oranges, madu susu and susu. Each type of orange has a different quality and a different price. The price difference often results in fraud committed by traders against buyers to the detriment of the buyer. This is because differentiating types of oranges based on the appearance of the fruit does not have a standard. Therefore, in this study, a citrus fruit classification system was created based on images by implementing deep learning. The method of deep learning used in this research is Convolutional Neural Network (CNN) with AlexNet architecture. The types of oranges that will be observed are madu oranges, madu susu, and siam. The data used are 2250 images of oranges with each class totaling 750 images with a size of 227x227 pixels. The training data is 1575 images and the test data is 675 images. The training is carried out with a total of 10 epochs and each epoch will produce a model. System testing is carried out based on the model generated in the training process. Each model will be observed results in the form of accuracy which is calculated using a confusion matrix. The most optimal model was generated from training in epoch the 9th which resulted in an accuracy of 94.81%.

Abstrak

Jeruk merupakan jenis buah yang mudah ditemukan di Kabupaten Sambas. Jenis-jenis jeruk yang banyak dibudidayakan adalah siam, madu susu, dan madu. Setiap jenis jeruk memiliki kualitas yang berbeda dimana harga dapat ditentukan dari jenis tersebut. Perbedaan harga tersebut tidak jarang mengakibatkan terjadinya penipuan

Keywords: convolutional neural network, orange, deep learning, alexnet
Kata kunci: convolutional neural network, jeruk, deep learning, alexnet

yang dilakukan oleh pedagang. Penipuan tersebut dapat terjadi karena tidak ada standar yang baku dalam membedakan jenis jeruk berdasarkan rupa. Oleh karena itu, pada penelitian ini di buat sistem klasifikasi jenis jeruk berdasarkan gambar buah dengan mengimplementasikan *deep learning*. Metode *deep learning* yang digunakan pada penelitian ini adalah *Convolutional Neural Network* (CNN) dengan arsitektur AlexNet. Jenis jeruk yang akan diamati ialah madu, madu susu, dan siam. Data yang digunakan berjumlah 2250 gambar jeruk dimana masing-masing kelas berjumlah 750 gambar dengan ukuran 227x227 piksel. Data latih berjumlah 1575 gambar dan data uji berjumlah 675 gambar. Pelatihan dilakukan dengan menggunakan 10 epoch dan setiap epoch akan menghasilkan model. Pengujian sistem dilakukan berdasarkan model yang dihasilkan pada proses pelatihan. Setiap model akan diamati hasil berupa akurasi yang dihitung dengan menggunakan *confusion matrix*. Model paling optimal dihasilkan dari pelatihan pada *epoch* ke-9 dimana menghasilkan akurasi pengujian sebesar 94,81%.

1. Pendahuluan

Sambas merupakan salah satu kabupaten dengan produksi jeruk terbanyak di Kalimantan Barat. Berdasarkan data dari *website* Badan Pusat Statistik Kabupaten Sambas, produksi buah jeruk pada tahun 2018 mencapai 1.190.770 kuintal[1]. Jenis jeruk yang banyak dijual ialah siam, madu susu, dan madu. Setiap jenis jeruk memiliki perbedaan kualitas terutama dari segi rasa yang dimiliki. Jeruk madu memiliki rasa yang lebih manis dan lebih segar dibandingkan dengan jeruk siam dan jeruk madu susu memiliki rasa yang lebih segar dibandingkan dengan jeruk siam.

Perbedaan kualitas tersebut tidak jarang dimanfaatkan oleh oknum tidak bertanggung jawab untuk melakukan penipuan. Penipuan dapat terjadi dikarenakan warna dan bentuk buah jeruk pada umumnya memiliki kemiripan yang sangat tinggi. Selain itu, dalam membedakan buah jenis buah tidak memiliki standar yang baku. Proses membedakan jenis jeruk pada umumnya masih dilakukan secara konvensional berdasarkan pengalaman yang didapat sebelumnya. Kondisi ini menyebabkan perlunya suatu alat yang dapat membantu proses klasifikasi varietas jeruk dengan lebih akurat.

Klasifikasi jenis jeruk dapat dilakukan dengan berbagai cara, salah satunya yaitu dengan bantuan *image processing* atau pengolahan gambar. Teknologi yang terkenal dengan akurasinya dalam melakukan klasifikasi melalui pengolahan gambar ialah *deep learning*. Teknologi ini merupakan evolusi dari teknologi *Machine Learning* yang berkerja dengan menirukan cara kerja penalaran manusia berdasarkan proses pelatihan, dan proses pembelajaran[2]. Implementasi dari *deep learning* memiliki tantangan tersendiri dimana terdapat banyak parameter yang mempengaruhi kinerjanya. Jumlah *epoch* menjadi salah satu parameter yang penting dimana mempengaruhi lamanya proses pelatihan dan kualitas dari model yang dihasilkan. Salah satu algoritma *Deep Learning* yang terkenal dengan akurasi yang tinggi adalah *Convolutional Neural*

Network (CNN) [3]. Algoritma CNN bekerja dengan melakukan ekstraksi fitur dan klasifikasi. CNN pernah diterapkan oleh Felix dkk. untuk melakukan identifikasi jenis tanaman melalui citra daun. Pada penelitian tersebut pelatihan dan pengujian fitur menggunakan algoritma *feedforward propagation* dan *backpropagation*. Berdasarkan penelitian tersebut didapatkan hasil akurasi yang tergolong rendah yaitu sebesar 76% [4].

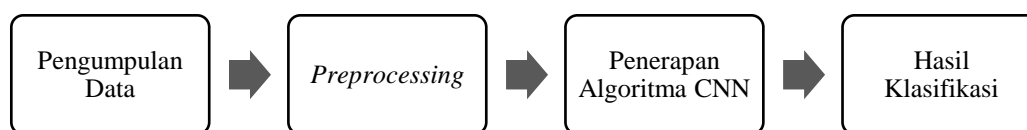
Metode CNN juga pernah diterapkan oleh Ilahiyah dan Nilogiri dengan judul Implementasi *Deep Learning* Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan *Convolutional Neural Network*. Penelitian tersebut menerapkan metode CNN dengan arsitektur Alexnet. Berdasarkan pengujian sistem yang menggunakan metode *Cross Validation* didapatkan nilai akurasi sistem sebesar 90,8% [5].

Penelitian terkait klasifikasi buah jeruk juga pernah dilakukan oleh Agustian dkk. dengan judul Klasifikasi Buah Jeruk Menggunakan Metode *Naive Bayes* Berdasarkan Analisis Tekstur dan Normalisasi Warna. Penelitian tersebut menggunakan *Gray Level Co-occurrence Matrix* (GLCM) untuk ekstraksi warna dan *Naive Bayes* untuk proses klasifikasi. Pada penelitian ini membedakan 4 jenis jeruk yaitu, jeruk pontianak, jeruk beby, jeruk mandarin, dan jeruk *orange*. Proses klasifikasi dilakukan berdasarkan analisis tekstur dan normalisasi warna. Hasil akurasi yang didapatkan dari pengujian sebesar 91,6%. Tingkat akurasi yang tinggi disebabkan karena warna dari setiap kelas memiliki perbedaan yang signifikan[6].

Berdasarkan uraian yang telah di paparkan pada paragraf sebelumnya maka untuk mengenali jenis buah jeruk dengan lebih akurat perlu dilakukan penelitian implementasi *Deep Learning* untuk klasifikasi jenis jeruk dengan menggunakan metode *Convolutional Neural Network*.

2. Metode

Penelitian ini dilakukan ke dalam beberapa tahap. Tahapan-tahapan tersebut dapat dilihat pada Gambar 1.



Gambar 1. Tahapan Penelitian

Pada tahap pertama dilakukan pengumpulan data gambar jeruk yang mana data tersebut akan dibagi menjadi data uji dan data latih. Setelah data gambar terkumpul dilakukan *preprocessing* seperti pemotongan dan penerapan *filter* untuk meningkatkan kualitas dan menghapuskan *noise* pada gambar. Tahap selanjutnya ialah penerapan algoritma CNN yang mana terdapat proses pelatihan dan proses pengujian. Hasil yang didapatkan akan berupa *labeling* dari gambar yang dimasukkan.

2.1. Pengumpulan Data

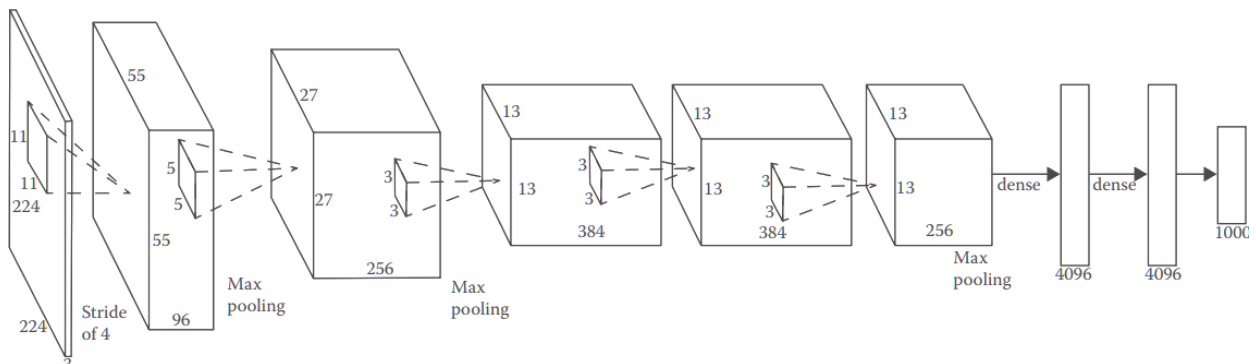
Pengumpulan data pada penelitian ini dilakukan dengan observasi. Data yang diambil berupa gambar buah jeruk dari kebun jeruk yang berada di Kabupaten Sambas. Buah jeruk yang telah dikumpulkan kemudian di foto untuk digunakan sebagai data pada penelitian ini. Foto dari buah jeruk diambil dengan ukuran serupa sehingga kualitas data dalam keadaan jelas. Kualitas data akan mempengaruhi model yang akan dihasilkan sistem. Gambar jeruk yang diambil berjumlah 2250 gambar dari 45 buah jeruk. Setiap kelas berjumlah 15 buah jeruk yang mana akan dilakukan augmentasi data pada gambar dari sudut yang berbeda secara acak sebanyak 50 gambar untuk setiap buah jeruk, sehingga dihasilkan 750 gambar jeruk untuk setiap kelasnya. Fungsi dari augmentasi data ialah merupakan sebuah cara untuk mengumpulkan data dengan teknik seperti *cropping*, *padding*, dan *flipping horizontal* sehingga meningkatkan jumlah data secara signifikan tanpa benar-benar mengumpulkan data baru[7].

2.2. Preprocessing

Data jeruk yang terkumpul berupa gambar. Gambar-gambar tersebut tidak semua memiliki kualitas yang jernih dan tidak terdapat *noise*. Berdasarkan hal tersebut dilakukan *preprocessing* untuk meningkatkan kualitas dari gambar dan mengurangi *noise* dari gambar yang akan digunakan. *Preprocessing* dilakukan dengan proses *cropping* gambar dan menerapkan *filter gaussian* sehingga mengurangi *noise* yang terdapat pada gambar[8].

2.3. Convolutional Neural Network

Convolutional Neural Network (CNN) ialah salah satu algoritma paling populer yang digunakan untuk *deep learning* dimana model pembelajarannya dikhususkan untuk melakukan klasifikasi langsung pada media 2 dimensi seperti gambar, video, teks atau suara.



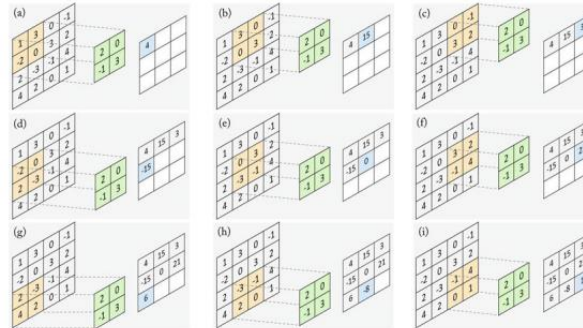
Gambar 2. CNN Arsitektur Alexnet[9]

Gambar 2. merupakan arsitektur AlexNet. Arsitektur tersebut digunakan oleh Alex dkk. dalam kompetisi *ImageNet Large Scale Visual Recognition Challenge* 2012. Pada kompetisi tersebut metode CNN yang di usulkan oleh Alex dkk. berhasil mengungguli metode *deep learning* lainnya seperti *Support Vector Machine* (SVM) pada kasus klasifikasi citra[10].

Secara umum arsitektur dari CNN dibagi menjadi 2 tahap, yaitu *feature learning* dan *classification*. *Feature learning* terdiri dari 3 lapisan proses. Proses-proses ini sering disebut dengan *convolution*, *activation*, dan *pooling*. Pada tahap *classification* terdiri dari *fully connected layer* yang mana lapisan tersebut tersusun dari lapisan-lapisan neuron dan diakhir terdapat fungsi *softmax* untuk menentukan kelas dari citra yang di masukkan.

2.3.1. Convolution Layer

Convolution layer merupakan komponen terpenting dari metode CNN. Pada layer ini terdiri dari satu set filter yang akan digunakan pada proses konvolusi terhadap masukan yang diberikan. Proses tersebut akan menghasilkan feature map. Setiap filter pada convolution layer terdiri dari sebuah grid yang berisikan nilai-nilai berbeda. Setiap nilai yang terdapat pada filter berperan seperti bobot yang akan di perbaharui setelah proses konvolusi. Proses konvolusi merupakan perkalian dari filter terhadap convolution layer dan hasil dari perkalian tersebut akan dijumlahkan[11].



Gambar 3. Proses Kovolusi[11]

Gambar 3. menunjukkan proses konvolusi. Grid dengan warna jingga menunjukkan convolution layer, grid dengan warna hijau menunjukkan filter dan grid dengan warna biru menunjukan hasil dari proses konvolusi. Proses pergeseran filter bersifat konstan sesuai dengan besarnya stride yang didefinisikan. Stride merupakan besar pergeseran filter. Secara matematis proses konvolusi ditunjukkan pada Persamaan 1[12].

$$x^l_{i,j} = \sum_m \sum_n w^l_{m,n} \cdot o^{l-1}_{i+m,j+n} + b^l \quad (1)$$

Variabel $x^l_{i,j}$ merupakan keluaran dari hasil proses konvolusi. Variabel $w^l_{m,n}$ merupakan kernel, $o_{i,j}$ merupakan input, dan b merupakan bias.

Perbaikan bobot dari kernel pada CNN menerapkan backpropagation yang mana mencari nilai gradient dari kernel dan gradient loss dari layer sebelumnya. Gradient kernel didapatkan dengan menggunakan persamaan 2[12].

$$\frac{\partial E}{\partial w^l_{m,n}} = rot_{180^\circ} \{ \delta^l_{i,j} \} * o^{l-1}_{m,n} \quad (2)$$

$\delta^l_{i,j}$ merupakan faktor error layer l . Sedangkan untuk menentukan nilai gradient input didapatkan dengan menggunakan persamaan 3[12].

$$\delta^l_{i,j} = \sum_{m=0}^{k^l-1} \sum_{n=0}^{k^l-1} \delta^{l+1}_{i'-m,j'-n} \cdot \frac{\partial x^{l+1}_{i'-m,j'-n}}{\partial x^l_{i,j}} \quad (3)$$

Pada penelitian ini dalam melakukan perhitungan pada proses konvolusi digunakan algoritma im2col yang mana berkerja dengan mengubah gambar yang semula memiliki lebih dari 1 channel menjadi 1 dimensi dan menjadi kolom. Proses tersebut akan meningkatkan waktu proses komputasi sehingga lebih cepat dari proses konvolusi native[12].

2.3.2. Fungsi Aktivasi

Fungsi aktivasi merupakan operasi matematik yang ditargetkan pada sebuah keluaran. Fungsi ini memiliki peran menentukan apakah suatu neuron aktif atau tidak berdasarkan *wighter sum* dari suatu *input*. Pada penelitian ini menggunakan 2 jenis fungsi aktivasi. Fungsi yang digunakan ialah *Reactifier Linear Unit* (ReLU) dan *Softmax*. Fungsi dari ReLU ialah untuk membatasi nilai negatif pada masukan dengan mengubahnya menjadi nol. Fungsi dari *Softmax* ialah untuk menentukan kelas pada tahap akhir metode CNN. Penjelasan lebih lanjut dari kedua fungsi aktivasi tersebut sebagai berikut.

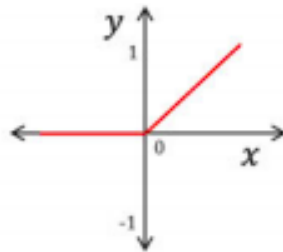
- 1) *Reactifier Linear Unit* (ReLU), fungsi ini merupakan fungsi sederhana yang memiliki kegunaan untuk kepentingan praktis khusus karena perhitungannya yang cepat. Fungsi ini mengubah masukan menjadi nilai 0 jika negatif dan mempertahankan nilainya jika positif. Secara matematis fungsi ReLU ditunjukkan seperti pada Persamaan 4[13].

$$f_{ReLU} = \max(0, x) \quad (4)$$

persamaan 2.6. membuat semua nilai negatif menjadi nol. Nilai gradient dari fungsi aktivasi ini dapat dihitung dengan Persamaan 5[14].

$$df_{ReLU} = (F > 0). 1 \quad (5)$$

dimana F merupakan nilai *gradient* dari *input* pada layer sebelumnya.



Gambar 4. Fungsi Aktivasi ReLu[11]

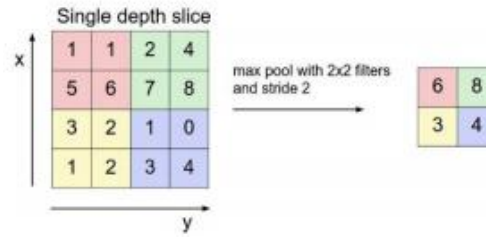
- 2) *Softmax*, fungsi aktivasi ini bekerja untuk melakukan perhitungan probabilitas pada setiap kelas target yang memungkinkan dan akan membantu menentukan kelas target pada masukan yang diberikan. Nilai *softmax* berada pada rentang probabilitas keluaran dari 0 hingga 1 dan semua probabilitas sama dengan 1. Secara matematis fungsi *softmax* ditunjukkan pada Persamaan 6[5].

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad (6)$$

Dengan e = eksponen, $i = 1, \dots, K$ dan $z = (z_1, \dots, z_K) \in R^K$

2.3.3. Pooling Layer

Pooling Layer merupakan sebuah lapisan yang terletak diantara *convolution layer*. Lapisan ini memiliki fungsi untuk mengurangi *volume output* pada *feature map* sehingga komputasi yang nantinya akan dilakukan akan berlangsung lebih cepat pada lapisan selanjutnya.



Gambar 5. Proses *Pooling*[5]

Gambar 5. menunjukkan proses dari *max-pooling*. Keluaran dari proses *pooling* ini ialah sebuah matrik dengan dimensi yang lebih kecil dari citra awal.

2.3.4. Fully Connected Layer

Fully connected layer merupakan sebuah lapisan *neural network* yang mana semua neuron aktivasi dari lapisan sebelumnya terhubung dengan neuron lapisan selanjutnya. Perbedaan yang dimiliki *fully connected layer* dengan *convolution layer* biasa adalah neuron yang dimiliki pada *convolution layer* terhubung hanya ke daerah tertentu, sedangkan pada *fully connected layer* memiliki neuron yang semuanya terhubung.

2.3.5. Cross Entropy Loss

Cross entropy loss merupakan *loss function* yang digunakan untuk masalah klasifikasi banyak kelas. *Cross entropy* akan menghitung skor yang merangkum perbedaan rata-rata antara distribusi probabilitas aktual dan prediksi untuk semua kelas dalam masalah yang diamati. Skor diminimalkan dan nilai *cross entropy* yang sempurna adalah 0. Pada metode CNN digunakan fungsi *cross entropy* seperti ditunjukkan pada Persamaan [15].

$$cost = - \sum_j y_j \log(z_j) \quad (7)$$

Keterangan:

$cost$ = skor ke i dari perbedaan nilai prediksi dengan nilai aktual

z_j = nilai masukan ke- j

y_j = Target dalam bentuk matrik 0 atau 1.

Pada proses pelatihan CNN diperlukan *loss function* yang bertujuan untuk menghitung *cost* dari hasil klasifikasi model. Nilai tersebut akan digunakan pada proses perbaikan model dengan menggunakan metode *backpropagation*.

2.4. Pelatihan dan Pengujian CNN

Pelatihan dilakukan untuk menghasilkan model yang berisikan nilai bobot dan bias yang berguna sebagai parameter dalam melakukan klasifikasi. Pelatihan dilakukan sebanyak 10 *epoch*. *Epoch* adalah parameter yang menjadi kontrol banyaknya perulangan pelatihan setelah seluruh data selesai dilatih. Pada tahap pelatihan setiap *epoch* akan menghasilkan model yang mana akan di uji pada tahap pengujian. Parameter yang diberikan pada pelatihan yaitu *learning rate* 0,0001 dan *epsilon* 0,00000001.

Tahap pengujian berguna untuk dilihat pengaruh dari jumlah *epoch* pada proses pelatihan CNN dan menentukan model paling optimal dari pelatihan. Pengujian dilakukan menggunakan metode *confusion matrix* dengan menghitung akurasi, presisi, *recall*, dan *F-measure*[16].

Perhitungan dilakukan untuk setiap model yang dihasilkan pada proses pelatihan sehingga di dapatkan model yang paling optimal.

3. Hasil dan Pembahasan

3.1. Hasil Pelatihan

Proses pelatihan dilakukan sebanyak 10 *epoch*. Setiap *epoch* akan menghasilkan model yang berisikan nilai parameter kernel, bobot, dan bias. Model yang dihasilkan untuk setiap *epoch* memiliki nilai *loss* dan akurasi. Nilai-nilai tersebut ditunjukkan pada Tabel 1.

Tabel 1. Hasil Pelatihan CNN

<i>Epoch</i>	Akurasi pelatihan (%)	<i>Loss</i>
1	94,03	0,1839616
2	97,02	0,1010554
3	97,52	0,0842002
4	98,6	0,0408925
5	98,73	0,0382395
6	99,11	0,026562
7	99,87	0,0113577
8	99,94	0,0101629
9	99,68	0,0090549
10	99,55	0,0121571

Pada Tabel 1. dapat dilihat akurasi pelatihan tertinggi didapat pada *epoch* ke-8 dengan akurasi 99.94% dan nilai *loss* terkecil didapat pada *epoch* ke-9 dengan nilai 0.0090549. Hasil yang didapat dari proses pelatihan menunjukkan akurasi model dari pelatihan cenderung meningkat untuk setiap *epoch*, namun ketika mencapai *epoch* ke-9 akurasi pelatihan menurun dimana pada *epoch* ke-8 akurasi pelatihan mencapai 99.94% kemudian ketika berada pada *epoch* ke-9 akurasi pelatihan menjadi 99.68%. Akurasi pada *epoch* ke-9 lebih kecil dari akurasi pada *epoch* ke-8 namun nilai *loss* dari model *epoch* ke-9 paling rendah dari seluruh model yang dihasilkan.

3.2. Hasil Pengujian

Model yang dihasilkan dari proses pelatihan akan dilakukan pengujian dengan data uji yang tidak terdapat pada data latih. Data uji berjumlah 675 data dengan masing-masing kelas terdapat 225 data. Akurasi, presisi, *recall*, dan *F1-measure* dari setiap model yang telah dihasilkan akan dihitung dengan menggunakan metode *confusion matrix*.

Tabel 2. Hasil Pengujian CNN

<i>Epoch</i>	Presisi (%)	<i>Recall</i> (%)	<i>F1-Measure</i> (%)	Akurasi (%)
1	88,59	88,3	88,44	88,3
2	87,64	86,96	87,29	86,96
3	86,60	88,15	87,36	88,15
4	94,30	87,56	90,80	87,56

Tabel 2. Lanjutan

5	92,06	92,0	92,02	92,0
6	93,74	93,63	93,68	93,63
7	92,43	92,44	92,43	92,44
8	92,01	92,0	92,00	92,0
9	94,90	94,81	94,85	94,81
10	92,89	93,19	93,03	93,19

Tabel 2. merupakan hasil dari proses pengujian. Seluruh model yang dihasilkan dari proses pelatihan di uji dengan menggunakan 675 data yang tidak digunakan pada proses pelatihan. Hasil yang didapatkan untuk akurasi tertinggi dimiliki oleh model yang dihasilkan pada *epoch* ke-9 dengan nilai akurasi sebesar 94,81%.

Model tersebut menghasilkan nilai presisi, *recall*, dan *F1-measure* tertinggi sebesar 94.90%, 94.81%, dan 94.85%. Pada proses pelatihan model dari *epoch* ke-9 menunjukkan akurasi latih lebih kecil dari model pada *epoch* ke-8 namun memiliki nilai *loss* lebih rendah dan memiliki akurasi paling tinggi pada pengujian terhadap data uji. Berdasarkan hasil tersebut, pada proses pelatihan semakin tinggi *epoch* yang digunakan tidak selalu menghasilkan model yang optimal dimana model pada *epoch* ke-10 yang merupakan model dengan jumlah *epoch* tertinggi memiliki akurasi uji lebih rendah dari model yang dihasilkan pada *epoch* ke-9. Jumlah *epoch* yang tinggi akan menyebabkan semakin lama proses pelatihan dilakukan. Selain itu, dalam menentukan batas akhir pelatihan tidak dapat hanya dengan menggunakan batas jumlah *epoch* karena berdasarkan hasil yang didapat jumlah *epoch* ke-10 tidak menghasilkan model yang paling optimal.

4. Kesimpulan

Kesimpulan yang dapat diambil dari hasil penelitian yang telah dilakukan pada klasifikasi jenis jeruk dengan menggunakan metode *convolutional neural network* ini yaitu pelatihan yang dilakukan dengan jumlah 10 *epoch* menunjukkan hasil dimana akurasi latih mengalami peningkatan dari *epoch* ke-1 hingga *epoch* ke-8, kemudian akurasi latih menurun pada *epoch* ke-9. Akurasi latih tertinggi yang dimiliki oleh model yang dihasilkan pada *epoch* ke-8, namun, ketika di lakukan pengujian pada model tersebut tidak menghasilkan akurasi uji tertinggi. Selain itu, model yang dihasilkan pada *epoch* ke-10 yang merupakan model dengan jumlah *epoch* tertinggi tidak menghasilkan akurasi uji tertinggi. Akurasi uji tertinggi dimiliki oleh model yang dihasilkan pada *epoch* ke-9. Berdasarkan hasil tersebut, semakin tinggi jumlah *epoch* tidak menjamin akan menghasilkan model yang optimal. Model yang paling optimal yang dihasilkan pada penelitian ini memiliki nilai *loss* paling rendah sebesar 0,0090549 yang mana merupakan model yang dihasilkan pada *epoch* ke-9. Berdasarkan hasil tersebut, dalam menentukan model paling optimal, parameter nilai *loss* terkecil menunjukkan hasil lebih baik dibandingkan dengan parameter akurasi pelatihan dan jumlah *epoch* tertinggi. Model yang dihasilkan pada *epoch* ke-9 yang merupakan model paling optimal memiliki nilai presisi, *recall*, dan *F1-measure* tertinggi dari model yang lain sebesar 94.90%, 94.81%, dan 94.85%. Selain itu model yang dihasilkan pada *epoch* ke-9 tersebut menghasilkan akurasi pengujian paling tinggi sebesar 94,81%.

Daftar Pustaka

- [1] “BPS Kabupaten Sambas.”
<https://sambaskab.bps.go.id/dynamictable/2019/10/18/67/jumlah-tanaman-menghasilkan-dan-produksi-tanaman-buah-buahan-dan-sayuran-tahunan-di-kabupaten-sambas-tahun-2016-2018.html> (accessed Sep. 01, 2021).
- [2] A. Santoso and G. Ariyanto, “IMPLEMENTASI DEEP LEARNING BERBASIS KERAS UNTUK PENGENALAN WAJAH,” *Jurnal Teknik Elektro*, vol. 18, no. 01, [Online]. Available: <https://www.mathworks.com/discovery/convol>
- [3] A. Kurniadi and M. Fal Sadikin, “Implementasi Convolutional Neural Network Untuk Klasifikasi Varietas Pada Citra Daun Sawi Menggunakan Keras Implementation of Neural Network Convolutions For Classification of Variety on Image of Collards Meat Leaves Using The Keras,” vol. 4, no. 1, pp. 25–33, 2020, [Online]. Available: <http://e-journal.unipma.ac.id/index.php/doubleclick>
- [4] F. Felix, J. Wijaya, S. P. Sutra, P. W. Kosasih, and P. Sirait, “Implementasi Convolutional Neural Network Untuk Identifikasi Jenis Tanaman Melalui Daun,” *Jurnal SIFO Mikroskil*, vol. 21, no. 1, pp. 1–10, 2020.
- [5] S. Ilahiyah and A. Nilogiri, “Implementasi Deep Learning Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan Convolutional Neural Network,” *JUSTINDO (Jurnal Sistem dan Teknologi Informasi Indonesia)*, vol. 3, no. 2, pp. 49–56, 2018.
- [6] W. Agustian, S. Setyaningsih, and Q. Arie, “Klasifikasi Buah Jeruk Menggunakan Metode Naive Bayes Berdasarkan Analisis Tekstur dan Normalisasi Warna,” *Jurnal Online Mahasiswa (JOM) Bidang Ilmu Komputer/Informatika*, vol. 2, no. 2, 2015.
- [7] J. Sanjaya and M. Ayub, “Augmentasi Data Pengenalan Citra Mobil Menggunakan Pendekatan Random Crop, Rotate, dan Mixup,” *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 6, no. 2, Aug. 2020, doi: 10.28932/jutisi.v6i2.2688.
- [8] A. Wedianto, H. Latipa Sari, and Y. Suzantri H, “ANALISA PERBANDINGAN METODE FILTER GAUSSIAN, MEAN DAN MEDIAN TERHADAP REDUKSI NOISE,” *Jurnal Media Infotama*, vol. 12, Feb. 2016.
- [9] R. Venkatesan and B. Li, *Convolution Neural Networks in Visual Computing A Concise Guide*. Florida: CRC Press, 2017.
- [10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks”, [Online]. Available: <http://code.google.com/p/cuda-convnet/>
- [11] S. Khan, H. Rahmani, S. A. A. Shah, and M. Bennamoun, *A Guide to Convolutional Neural Networks for Computer Vision*, vol. 8, no. 1. Morgan publishers & cLaypool, 2018. doi: 10.2200/s00822ed1v01y201712cov015.
- [12] Y. Achmad, R. C. Wihandika, and C. Dewi, “Klasifikasi emosi berdasarkan ciri wajah wenggunakan convolutional neural network,” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 3, no. 11, pp. 10595–10604, 2019.

- [13] M. Satria Wibawa, “Pengaruh Fungsi Aktivasi, Optimisasi dan Jumlah Epoch Terhadap Performa Jaringan Saraf Tiruan,” *JURNAL SISTEM DAN INFORMATIKA*, vol. 11, Nov. 2016.
- [14] I. W. S. E. Putra, A. Y. Wijaya, and R. Soelaiman, “Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101,” *Jurnal Teknik ITS*, vol. 5, no. 1, 2016, doi: 10.12962/j23373539.v5i1.15696.
- [15] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge: MIT Press, 2016.
- [16] I. M. Erwin, E. Prakasa, and B. Sugiarto, “Kayu7Net: Identifikasi Dan Evaluasi F-Measure Citra Kayu Berbasis Deep Convolution Neural Network (Dcnn) Kayu7Net : Identification and F-Measure Evaluation Wood Image Based on Deep Convolution Neural Networks (Dcnn),” *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*, vol. x, no. 30, pp. 1–10, 2019.