

Klasifikasi Tekstur Kematangan Buah Jeruk Manis Berdasarkan Tingkat Kecerahan Warna dengan Metode *Deep Learning Convolutional Neural Network*

Budi Yanto¹, Luth Fimawahib², Asep Supriyanto³, B. Herawan Hayadi⁴, Rinanda Rizki Pratama⁵
^{1,2,3,5}Universitas Pasir Pengaraian, Jl. Tuanku Tambusai, Rambah Hilir, Rokan Hulu
⁴Universitas Prof. Dr. Hazarin, SH, Jl. Ahmad Yani, No. 1 Bengkulu
Email: budiyantost@gmail.com¹, luthfimawahib@gmail.com², asep.tif@gmail.com³,
b.herawan.hayadi@gmail.com⁴, rinandarizkipratama@gmail.com⁵

Abstract - Sweet orange is very much consumed by humans because oranges are rich in vitamin C, sweet oranges can be consumed directly to drink. The classification carried out to determine proper (good) and unfit (rotten) oranges still uses manual methods, The classification has several weaknesses, namely the existence of human visual limitations, is influenced by the psychological condition of the observations and takes a long time. One of the classification methods for sweet orange fruit with a computerized system the Convolutional Neural Network (CNN) is algorithm deep learning to the development of the Multilayer Perceptron (MLP) with 100 datasets of sweet orange images, the classification accuracy rate was 97.5184%. the classification was carried out, the result was 67.8221%. Testing of 10 citrus fruit images divided into 5 good citrus images and 5 rotten citrus images at 96% for training 92% for testing which were considered to have been able to classify the appropriateness of sweet orange fruit very well. The graph of the results of the accuracy testing is 0.92 or 92%. This result is quite good, for the RGB histogram display the orange image is good

Keyword - Convolutional Neural Network (CNN), Deep Learning, Sweet orange

Intisari - Jeruk manis sangat banyak dikonsumsi oleh manusia, karena jeruk kaya akan vitamin C. Pada umumnya proses klasifikasi untuk menentukan jeruk yang layak (baik) dan tidak berkualitas (busuk) masih menggunakan cara manual yaitu pengamatan langsung secara visual terhadap buah. Ada beberapa kelemahan pengklasifikasian dengan cara ini, yaitu adanya keterbatasan penglihatan manusia, dipengaruhi oleh kondisi psikologis pengamatan dan membutuhkan waktu yang lama terutama untuk perkebunan besar. Salah satu cara untuk mengklasifikasikan jeruk manis melalui sistem komputer memakai algoritma *deep learning convolutional neural network (CNN)*. Dengan dikembangkannya pada lapisan *multilayer perceptron (MLP)*, algoritma tersebut dapat diolah dua dimensi data, terhadap gambar serta mampu melakukan klasifikasi pada citra dengan kelas-kelas yang lebih banyak atau besar. Pada penelitian ini dengan 100 *dataset* gambar jeruk manis menunjukkan tingkat akurasi klasifikasi maka hasilnya sebesar 97.5184 %. dilakukan klasifikasi maka hasilnya sebesar 67.8221 %. pengujian sebanyak 10 citra buah jeruk yang terbagi menjadi 5 citra jeruk bagus dan 5 citra jeruk busuk sebesar 96% untuk *training* 92% untuk *testing* yang dinilai telah mampu melakukan klasifikasi kelayakan buah jeruk manis dengan sangat baik. Grafik hasil akurasi *testing* nya adalah 0.92 atau 92%. Hasil ini cukup bagus, terhadap Tampilan histogram RGB citra jeruk bagus

Kata Kunci - Convolutional Neural Network (CNN), Deep Learning, Jeruk Manis

I. PENDAHULUAN

Tanaman jeruk manis (*Citrus Sinensis*) adalah tanaman buah tahunan yang berasal dari Asia.[1] Diyakini bahwa Cina adalah tempat pertama bagi jeruk yang tumbuh. Selama ratusan tahun, jeruk telah tumbuh di Indonesia, baik secara alami maupun dibudidayakan. Tanaman *orange* di Indonesia adalah peninggalan Belanda yang membawa jeruk manis dan keprok dari Amerika dan Italia. [2]. Tanaman Jeruk manis paling cocok ditanam di daerah subtropis yang memiliki suhu rata-rata 20 – 25°C.[3] Jeruk manis mempunyai kedudukan paling istimewa diantara jenis jeruk lain dan merupakan kunci bagi industri jeruk diseluruh dunia. Sebab jeruk jenis ini merupakan bahan pembuat minuman yang sangat baik dan buahnya dapat dibuat (selai)[4]. Selain itu, bunga, biji dan Kulitnya dapat diambil minyaknya[5]. Buah non-klimaerik jeruk ini ketika matangnya buah, hanya dapat diperoleh dipetik langsung dipohon atau tidak dapat diperam setelah proses panen[6]. Sehingga penting untuk memetik ataupun memanen buah jeruk saat matangnya[7]. Memanen buah dengan rasa asam tidak akan berubah jadi manis karena tidak ada proses pematangan saat setelah panen [8]. Proses klarifikasi yang dilakukan untuk menentukan jeruk layak (bagus) dan tidak layak (busuk) bahkan begitu banyak inputan menggunakan cara manual [9] [10]. Manual berarti dibuat sesuai dengan pengamatan visual langsung pada buah yang akan diklasifikasikan [11]. Kelemahan dari klasifikasi manual sangat dipengaruhi oleh subjektivitas operator penyortiran sehingga, dalam kondisi tertentu, itu tidak konsisten dengan proses klasifikasi [12].

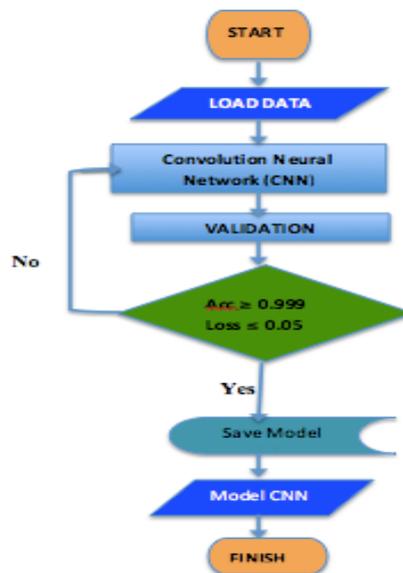
Convolutional Neural Network (CNN) bahagian dari algoritma *deep learning* dikembangkan dengan *Multilayer Perceptron* (MLP) dirancang sebagai mengolah data dalam bentuk dua dimensi, misalnya suara dan gambar [13]. CNN digunakan mengklasifikasi data yang terlabel dengan menfungsikan metode *supervised learning*[14], [15]. Bagian daripada jenis *neural network* biasanya digunakan didata image adalah CNN. Pada tingkat kedalaman jaringan CNN termasuk jenis *Deep Neural Network* dan sering digunakan dalam data citra image [16]. Tahap pembelajaran menggunakan *backpropagation*, klasifikasi menggunakan *feedforward* dua bahagian tersebut termasuk kedalam CNN. Keuntungan dari CNN dibandingkan dengan metode klasifikasi lain semakin terlihat ketika digunakan untuk mengklasifikasikan pada citra gambar dengan lebih banyak atau kelas yang besar [17].

Deep Learning merupakan bagian *Machine Learning* dengan basis Jaringan Syaraf Tiruan (JST)[18] atau pembaharuan JST dalam mendedikasikan komputer untuk dapat berpikir melakukan layaknya seperti manusia[19]. *Deep Learning*, pada komputer dikategorikan kedalam mengklasifikasi pengambilan dari gambar, suara, teks, atau video [20]. Seperti pada komputer pengujian dan dilatih memakai data set yang memiliki label dan dalam jumlah besar selanjutnya diubah kedalam nilai piksel pada gambar untuk dijadikan representasi internal atau *feature vector*[21] kemudian selanjutnya pengklasifikasian didapatkan dan digunakan pada deteksi untuk mengklasifikasikan pola pada masukan *input* [22]. *Deep learning* merupakan pembelajaran representasi untuk membentuk arsitektur jaringan syaraf tiruan dengan multi lapisan layer[23]. *Input layer*, *hidden layer*, dan *output layer* dalam lapisan *deep learning*, [24]. Pada *hidden layer* dengan lapisan yang banyak untuk dapat membuat komposisi algoritma dalam hal meminimalisir nilai *error* pada *output* [25]

Atas dasar penelitian terkait yang menggunakan metode *Neuron Network* yang konklusif adalah penerapan jaringan *neuron convolutional network* untuk klasifikasi gambar tomat menggunakan Keras (Taiara Shafira, 2018), yang merupakan subjek penelitian. Pada pencarian kali ini adalah tomat, menggunakan R studio dan Keras, [26]. Dengan hasil penelitian Tingkat keakuratan data uji yang diperoleh dari analisa pemodelan yang terbentuk adalah 90% klasifikasi gambar tomat. Data baru klasifikasi adalah 10 gambar tomat dalam pengujian model tomat yang terbentuk, yaitu setiap gambar berhasil diklasifikasikan dengan benar. Untuk itu disini diterapkannlah bagaimana klasifikasi terhadap Buah Jeruk Manis dengan menggunakan metode tersebut.

II. SIGNIFIKANSI STUDI

Penelitian ini dilakukan dengan melaksanakan beberapa fase tahapan yang terkait. Tahapan dijelaskan dalam metode penelitian. Metode penelitian dijelaskan menjadi bentuk skema yang jelas, terorganisir, dan sistematis.



Gambar 1 Kerangka Metodologi

Tahapan pertama adalah menginputkan citra 64x64x3, tinggi dan lebar citra gambar 64 dan citra pada gambar yaitu 3 *channel red, green, blue (RGB)* dengan matriks berbeda-beda.[27] Uji data latih 60 kali. Selanjutnya input gambar kepada tahap *convolution* untuk konvolusi ini memiliki terhadap filter yang sudah ditetapkan. Filter digunakan untuk menentukan pola deteksi tepi dilanjutkan proses perkalian matriks dengan nilai yang sudah di inputkan. Hasil konvolusi 64 piksel ukuran kernel 3x3. hasil 64 *feature map*. Selanjutnya dilakukan proses penghilangan nilai negatif dengan fungsi fungsi aktivasi *ReLU (Rectified Linear Unit)*. setelah proses konvolusi[28]. Tahapan berikutnya yaitu Proses *Pooling* ini menggunakan ukuran 2x2 dengan *stride* 1 dimana nilai setiap pegeseran setiap kernel matriks berjumlah satu [29].

$$\frac{409}{1} = 4.09\% \quad \text{overall accuracy} = \frac{TTP \text{ all}}{\text{Total Number of Testing Entries}} \times 100\% \quad (1)$$

Jadi akurasi yang dihasilkan oleh model dengan *input* gambar 64x64 piksel dengan jumlah *testing* sebanyak 1 kali didapatkan nilai akurasi sebesar 4.09%. Kemudian Fungsi *output Feature Map* dari input citra. Operasi konvolusi dapat dituliskan sebagai berikut:

$$s(t) = (x * t)(t) = \sum_{\alpha=-\infty}^{\infty} (\alpha) * w(t - \alpha) \quad (2)$$

Keterangan:

- S (t) : fungsi hasil operasi konvolusi
- X : input
- W : bobot (karnel)

Fungsi *s(t)* memberikan *output* tunggal berupa *feature map*. Argumen pertama adalah *input* yang merupakan *x* dan argumen kedua *w* sebagai kernel atau filter. Apabila dilihat *input* sebagai citra dua dimensi, maka bisa dikatakan *t* sebagai piksel dan menggantinya dengan *i* dan

j. Maka dari itu, operasi untuk konvolusi ke *input* dengan lebih dari satu dimensi dapat menulis sebagai berikut.

$$s(i, j) = s(i, j) = (k * i)(i, j) = \sum_{-\infty}^{\infty} \sum_{-\infty}^{\infty} I(i - m, j - n)K(m, n) \tag{3}$$

$$(k * i)(i, j) = (K * i)(i, j) = \sum_{-\infty}^{\infty} \sum_{-\infty}^{\infty} I(i + m, j + n)K(m, n) \tag{4}$$

Berdasarkan persamaan kedua diatas perhitungan dasar dalam konvolusi, *I, j* adalah sebuah piksel dari citra. Sebagai alternatif operasi konvolusi dapat dilihat sebagai perkalian matrik antara citra kernel dimana keluarannya dihitung dengan *dot product*[30].

III. HASIL DAN PEMBAHASAN

Proses awal dilakukan dengan menginputkan gambar dengan ukuran 64x64x3, yang mana tinggi dan lebar dari gambar yang diinput tersebut adalah 64 dan gambar tersebut memiliki 3 *channel* yaitu *red, green, blue* atau yang biasa disebut dengan RGB setiap *channel* tersebut memiliki matriks yang berbeda-beda. Pada sample di uji sebanyak 60 kali. Setelah di input gambar tersebut maka akan masuk pada tahap *convolution* yang mana pada konvolusi ini memiliki filter yang sudah ditentukan. Filter digunakan untuk menentukan pola apa yang dideteksi yang selanjutnya akan dikalikan dengan matriks dengan nilai matriks yang di input. Jumlah filter pada konvolusi ini sebanyak 64 piksel dengan ukuran kernel 3x3. Yang mana akan dihasilkan gambar sebanyak 64 *future map*. Agar dapat dipahami, peneliti mengambil contoh sebagian matriks pada *input image* yang berukuran 64x64.

23	25	0	21	22
23	25	0	0	25
25	25	0	20	20
20	25	24	0	25
25	24	0	25	24

1	1	1
-1	1	-1
1	-1	1

125	65	38
117	29	74
80	93	40

Gambar 2. Perhitungan Konvolusi

Pada gambar 2 menunjukkan proses konvolusi dengan menggunakan ukuran kernel 3x3 dengan menggunakan stride 1. Stride disini artinya jumlah pergeseran kernel terhadap matriks input berjumlah satu. Untuk mendapat kan hasil dari pehitungan tersebut digunakan fungsi dot product, perhitungannya dapat dilihat dibawah ini:

- a. *Position 1* = (23 x 1) + (23 x (-1)) + (25 x 1) + (25 x 1) + (25 x (-1)) + (25 x 1) + (0x1) + (0x1) + (0x(-1)) + (0x1) = 125.
- b. *Position 2* = (25 x 1) + (25 x (-1)) + (25x1) + (0x1) + (0x1) + (0x1) + (21x1) + (0 x (-1)) + (20x1) = 65.
- c. *Position 3* = (0x1) + (0 x (-1)) + (0x1) + (21x1) + (0x1) + (20x1) + (22x1) + (25x(-1)) + (20x1) = 38.
- d. *Position 4* = (23x1) + (25 x (-1)) + (20x1) + (25x1) + (25x1) + (25x1) + (0x1) + (0 x (-1)) + (24x1) = 117.
- e. *Position 5* = (25x1) + (25x(-1)) + (25x1) + (0x1) + (0x1) + (24x1) + (0x1) + (20x(-1)) + (0x1) = 29.
- f. *Position 6* = (0x1) + (0x(-1)) + (24x1) + (0x1) + (20x1) + (0x1) + (25x1) + (20x(-1)) + (25x1) = 74.
- g. *Position 7* = (25x1) + (20x(-1)) + (25x1) + (25x1) + (25x1) + (24x1) + (0x1) + (24x(-1)) + (0x1) = 80.
- h. *Position 8* = (25x1) + (25x(-1)) + (24x1) + (0x1) + (24x1) + (0x1) + (20x1) + (0x(-1)) + (25x1) = 93.

i. $Position\ 9 = (0x1) + (24x(-1)) + (0x1) + (20x1) + (0x1) + (25x1) + (20x1) + (25x(-1)) + (24x1) = 40.$

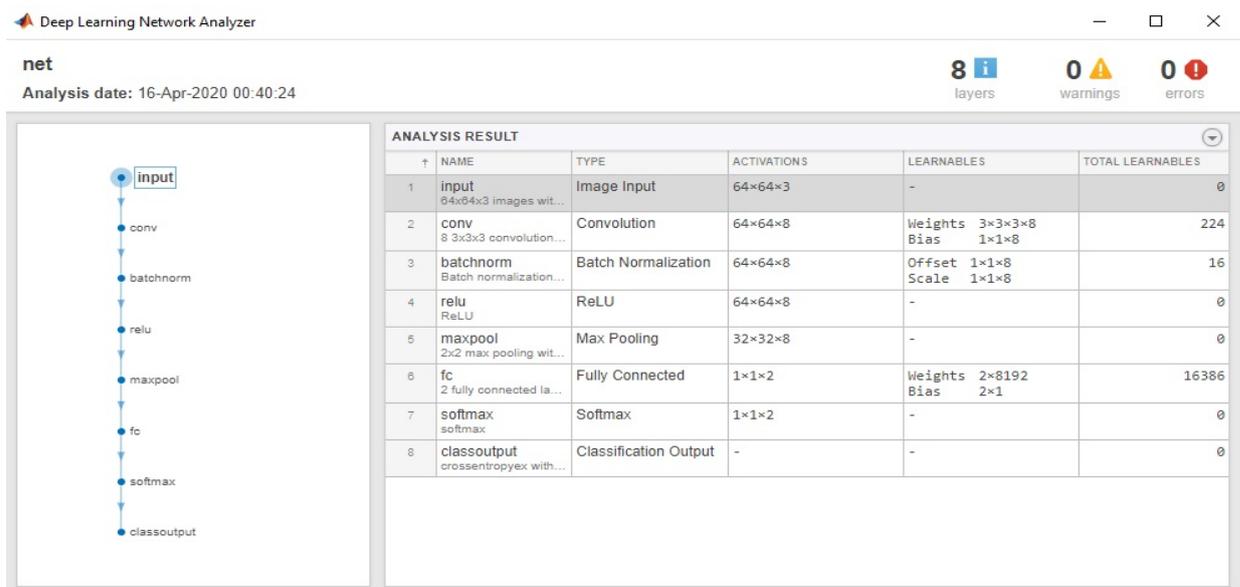
Setelah hasil perhitungan dengan konvolusi dapat, untuk menghilangkan nilai negatif pada hasil, maka digunakan fungsi aktivasi *ReLU (Rectified Linear Unit)* setelah proses konvolusi. Selanjutnya masuk kedalam proses *Pooling layers* dimana pada proses ini menggunakan fungsi *max pooling* yang mana dengan cara mengambil nilai maksimum matriks dari hasil konvolusi berikut gambaran proses pooling layer:

$$\begin{array}{|c|c|c|} \hline 125 & 65 & 38 \\ \hline 117 & 29 & 74 \\ \hline 80 & 93 & 40 \\ \hline \end{array} = \begin{array}{|c|c|} \hline 125 & 74 \\ \hline 117 & 93 \\ \hline \end{array}$$

Gambar 3. Poolin Layer

Softmax Layer, Image Input Layer, Convolutional Layer, ReLu Layer Fully Connected Layer, Classification Layer Batch Normalization Layer, Max Pooling Layer, dengan banyak ciri yang sudah di tentukan. konvolusi 1 kali di tunjukan pada *Convolutional Layer* yang di gunakan untuk mendapatkan model klasifikasi dengan filter sebanyak 8 dan kernel sebesar 3x3.

Sementara itu *Batch Normalization Layer* berfungsi mengurangi jumlah nilai terhadap pergeseran lapisan tersembunyi. *ReLu* dijadikan proses *training* lebih tepat. Sedangkan *Max Pooling* yaitu sebesar 2x2 dengan tujuan mengurangi dimensi dari *feature map*. *Softmax* pada nilai intuitif, pada klasifikasi interpretasi probabilistik dalam label di dihasilkan. *Classification Layer* digunakan untuk menunjukkan *output* bagus atau busuk.

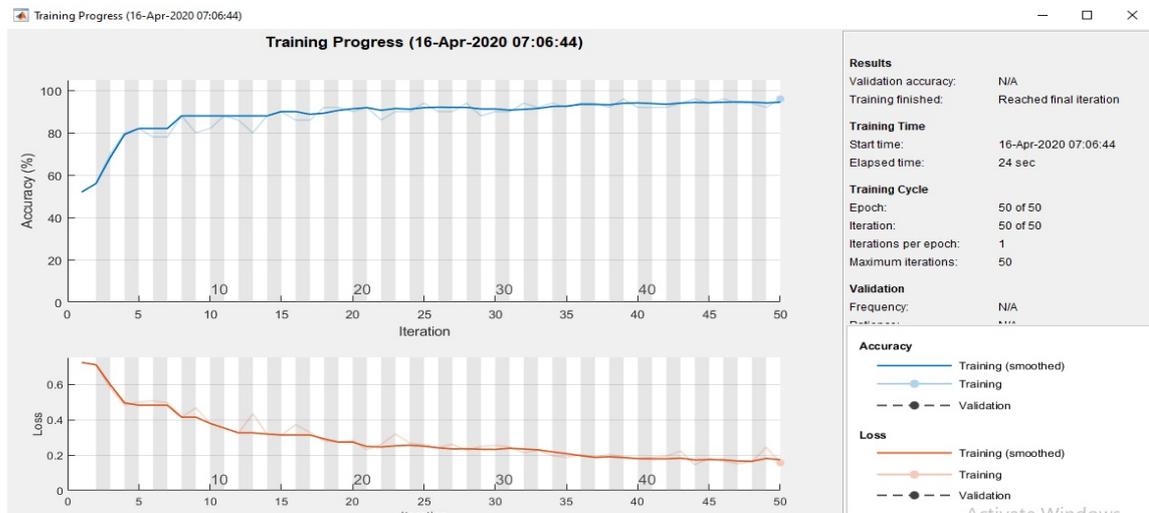


Gambar 4. Deep Learning Network Analyzer

Jumlah *learnables* yang di latih dalam model ini sebanyak 16.626 parameter. Berdasarkan hasil yang terbentuk, dapat diketahui yang ditunjukan pada *training* data. Adapun perbandingan *Loss* dan *accuracy* data terlihat di gambar 5 dan 6.

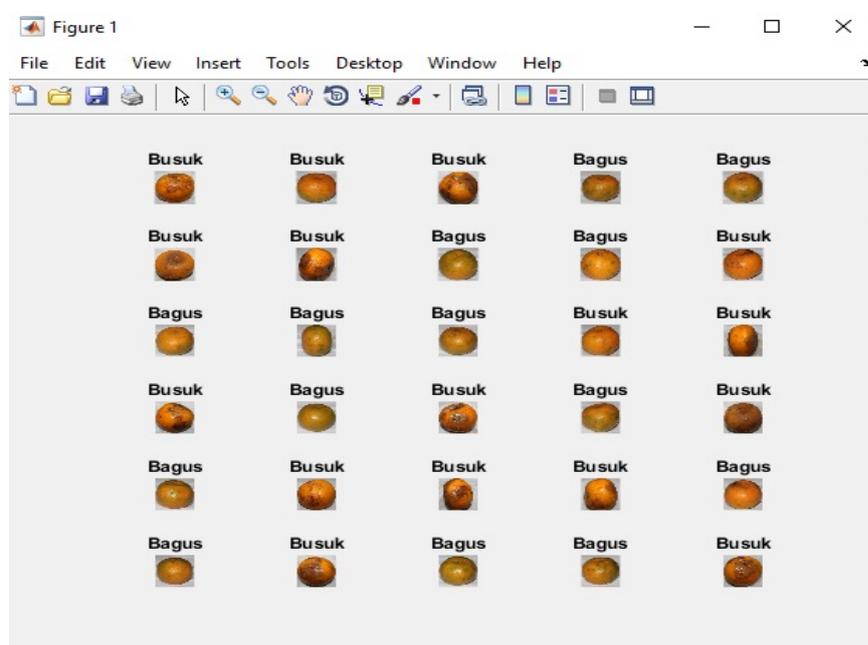
Epoch	Iteration	Time Elapsed (hh:mm:ss)	Mini-batch Accuracy	Mini-batch Loss	Base Learning Rate
1	1	00:00:02	52.00%	0.7250	1.0000e-04
50	50	00:00:24	96.00%	0.1563	1.0000e-04

Gambar 5. Akurasi data *Training*



Gambar 6. Data *training Plot loss dan accuracy*

Model yang terbentuk *Loss* dan *accuracy*, di tunjukan plot. *Loss function* adalah fungsi untuk menggambarkan kekurangan terkait pada kemungkinan yang akan diproduksi oleh model. *Loss Function* dikatakan baik ketika fungsi menghasilkan kesalahan sesuai dengan yang paling rendah. Untuk tingkat akurasi didefinisikan sebagai persentase dari data uji yang diklasifikasikan pada kelas yang benar. Nilai kehilangan yang dihasilkan dalam pelatihan adalah 0,1563. Nilai ini dapat dikatakan cukup rendah dan ini bagus untuk model. Nilai akurasi yang tinggi, yang mencapai 96% untuk pelatihan data dan 92% untuk tes data.



Gambar 7. Data *Training dan Test*

Untuk data training dan test dengan pemodelan yang diperoleh mampu mengklasifikasikan dengan benar. Proses training mendekati nol atau nilai rendah dan gerakan presisi menunjukkan hasil yang baik hingga 50 *epoch*. Proses training akan terhenti jika telah memenuhi syarat.

Dari hasil training didapatkan deteksi garis tepi yang dapat di simpulkan bahwa gambar *Laplace of Gaussian* (LOG terdapat banyak *noise*. Hasil dari proses pengklasifikasian model CNN tersebut berupa jeruk bagus dan jeruk busuk beserta skornya klasifikasi pada jeruk bagus.



Gambar 8 Input dan Klasifikasi pada Jeruk Bagus

Pada gambar di atas dapat di simpulkan bahwa hasil klasifikasi terdapat citra jeruk bagus, setelah dilakukan klasifikasi maka hasilnya sebesar 97.5184 %. Sedangkan proses klasifikasi pada jeruk busuk.



Gambar 9 Input dan Klasifikasi pada Jeruk Busuk

Pada gambar 9 terdapat citra jeruk busuk, setelah dilakukan klasifikasi maka hasilnya sebesar 67.8221 %. pengujian sebanyak 10 citra buah jeruk yang terbagi menjadi 5 citra jeruk bagus dan 5 citra jeruk busuk.

Hasil pengujian klasifikasi citra buah jeruk manis menggunakan metode CNN dengan 10 inputan yang dilakukan dengan data traing sebanyak 100 sample dan menghasilkan 50 epoch dapat terlihat dari tabel I.

TABEL I
HASIL KLASIFIKASI BUAH JERUK MANIS

No	Citra Jeruk Manis	Kelas	Skor	Nilai
1		Bagus	98.1564 %	Benar

No	Citra Jeruk Manis	Kelas	Skor	Nilai
2		Bagus	97.8437 %	Benar
3		Bagus	57.3703 %	Benar
4		Bagus	55.7629 %	Benar
5		Bagus	99.6214 %	Benar
6		Busuk	79.0748 %	Benar
7		Busuk	67.8221 %	Benar
8		Bagus	76.2958 %	Salah
9		Busuk	90.9586 %	Benar
10		Busuk	63.5467 %	Benar

Setiap gambar yang di *input*, hasilnya akan selalu berbeda tergantung pada bentuk dan warna jeruknya. Dari 10 citra jeruk manis dihasilkan 9 citra jeruk manis dengan nilai benar dan 1 citra jeruk manis dengan nilai salah. Pada pengujian nomor 8 terdapat kesalahan yaitu jeruk busuk terklasifikasi sebagai jeruk bagus dengan skor sebesar 76.2958 %. Masalah ini terjadi karena warna gelap pada citra jeruk manis tertutupi oleh bias cahaya. Itulah alasan mengapa pencahayaan pada citra yang akan di kelola harus diperhatikan

IV. KESIMPULAN

Penerapan *Convolutional Neural Network* (CNN) untuk klasifikasi buah jeruk Manis, berdasarkan tingkat kecerahan warna terhadap sampel data ini berupa *dataset* citra jeruk bagus dan jeruk busuk sebanyak 100 citra, dengan pelabelan menggunakan Bagus dan Busuk. Label bagus mewakili jeruk yang layak. Jumlah *learnables* yang di latih dalam model ini sebanyak 16.626 parameter. Pada nilai plot loss dan accuracy training dengan nilai *loss* yang di hasilkan pada *training* sebesar 0.1563. bisa di katakan rendah dan bagus untuk model. Tingkat akurasi dari testing data training sebanyak 250 data, didapatkan *accuracy* 96%, *training* 92% *testing*, terhenti pada 50 *epoch*. Proses pengambilan gambar dengan menggunakan smartphone, dengan pengklasifikasian model CNN yang dilakukan sudah layak mengenali dalam pengklasifikasin buah jeruk manis dengan kelas citra gambar jeruk manis terhadap tektur yang paling mirip. Dari segi kemamfaatan bagi masyarakat pengembangan dengan klasifikasi menggunakan smartphone yang terdeteksi langsung berdasarkan warna, bobot akan terlihat dan mampu memisahkan buah jeruk dengan baik.

REFERENSI

- [1] M. Lukmana and F. Sahab, Respon Pertumbuhan Bibit Jeruk Manis (*Citrus sinensis* L.) terhadap Pemberian Limbah Solid Industri Kelapa Sawit, *Agrisains J. Budid. Tanam. Perkeb. Politek. Hasnur*, vol. 6, no. 02, 2021, doi: 10.46365/agrs.v6i02.410.
- [2] D. Tobing, E. Bayu, and L. Siregar, Identifikasi Karakter Morfologi Dalam Penyusunan Deskripsi Jeruk Siam (*Citrus Nobilis*) Di Beberapa Daerah Kabupaten Karo, *J. Agroekoteknologi Univ. Sumatera Utara*, vol. 2, no. 1, pp. 72–95, 2013, doi: 10.32734/jaet.v2i1.5722.
- [3] Pakpahan, Rina Indriany P., Sarifuddin, and Supriadi Pemberian Bahan Amandemen Untuk Perbaikan Retensi Hara Tanaman Jeruk Manis (*Citrus Sinensis* L.) di Desa Talimbaru Kecamatan Barusjahe Kabupaten Karo, *Agroekoteknologi*, vol. 4, no. 1, 2015, doi: 10.32734/jaet.v4i1.12891.
- [4] M. Michiko, C. V. Manalu, and M. S. Mutia, Uji Efektivitas Ekstrak Etanol Kulit Jeruk Manis (*Citrus Sinensis*) Terhadap Bakteri *Propionibacterium Acnes*, (*Jurnal Ilm. Mhs. Kesehat. Masyarakat*), vol. 5, no. 1, 2020, doi: 10.37887/jimkesmas.v5i1.10552.
- [5] B. Surya Wibowo, I. Tazi, and K. Triyana, Pengembangan Sistem Sensor Rasa Berbasis Membran Selektif Ion untuk Klasifikasi Buah Jeruk (Halaman 9 s.d. 13), *J. Fis. Indones.*, vol. 17, no. 49, 2014, doi: 10.22146/jfi.24405.
- [6] R. A. W. Ramadhan, M. Baskara, and Agus Suryanto, Pengaruh Pemberian Pupuk Npk Terhadap Fruit Set Tanaman Jeruk Manis (*Citrus sinensis* Osb .) VAR . PACITAN, *J. Produksi Tanam.*, vol. 3, 2015.
- [7] A. R. K. Haba and K. C. Pelangi, Sistem Cerdas Dalam Klasifikasi Kematangan Buah Jeruk Berdasarkan Fitur Ekstraksi Gcm Dengan Metode Naïve Bayes, *J. Teknol. dan Manaj. Inform.*, vol. 5, no. 2, 2019, doi: 10.26905/jtmi.v5i2.3935.
- [8] K. Warman, L. A. Harahap, and P. Munir, Identifikasi Kematangan Buah Jeruk Dengan Teknik Jaringan Syaraf Tiruan, *J. Rekayasa Pangan dan Pertan.*, vol. 3, no. 2, pp. 248–253, 2015.
- [9] R. Utami, E. Widowati, and A. Rahayu, Screening dan Karakterisasi Pektinesterase Sebagai Enzim Potensial Dalam Klarifikasi Sari Buah Jeruk Keprok Garut (*Citrus Nobilis* Var.*Chrysocarpa*), *J. Agritech*, vol. 35, no. 04, 2015, doi: 10.22146/agritech.9326.
- [10] R. Rahmadewi, G. L. Sari, and H. Firmansyah, Pendeteksian Kematangan Buah Jeruk Dengan Fitur Citra Kulit Buah Menggunakan Transformasi Ruang Warna HSV, *JTEV (Jurnal Tek. Elektro dan Vokasional)*, vol. 5, no. 1.1, 2019.
- [11] S. Yudawati and A. P. Wibowo, Kajian Terhadap Beberapa Jenis Buah Sebagai Alternatif Peningkat Kadar HB, *Biomed Sci.*, vol. 2, no. 2, 2014.
- [12] M. F. Barkah, Klasifikasi Rasa Buah Jeruk Pontianak Berdasarkan Warna Kulit Buah Jeruk Menggunakan Metode K-Nearest Neighbor, *Coding Rekayasa Sist. Komput.*, vol. 08, no. 01, 2020.
- [13] B. Yanto, B. -, J. -, and B. H. Hayadi, Indentifikasi Pola Aksara Arab Melayu Dengan Jaringan Syaraf Tiruan Convolutional Neural Network (CNN), *JSAI (Journal Sci. Appl. Informatics)*, vol. 3, no. 3, 2020, doi: 10.36085/jsai.v3i3.1151.
- [14] K. Muchtar, Chairuman, Yudha Nurdin, and Afdhal Afdhal, Pendeteksian Septoria pada Tanaman Tomat dengan Metode Deep Learning berbasis Raspberry Pi, *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 5, no. 1, 2021, doi: 10.29207/resti.v5i1.2831.
- [15] N. Kasim and G. S. Nugraha, Pengenalan Pola Tulisan Tangan Aksara Arab Menggunakan Metode Convolution Neural Network, *J. Teknol. Informasi, Komputer, dan Apl. (JTika)*, vol. 3, no. 1, 2021, doi: 10.29303/jtika.v3i1.136.
- [16] T. Ito *et al.*, Deep Neural Network Incorporating CNN and MF for Item-Based Fashion Recommendation, in *Lecture Notes in Computer Science (including subseries Lecture*

- Notes in Artificial Intelligence and Lecture Notes in Bioinformatics*), 2021, vol. 12280 LNAI, doi: 10.1007/978-3-030-69886-7_4.
- [17] M. B. Bejiga, A. Zeggada, A. Nouffidj, and F. Melgani, A convolutional neural network approach for assisting avalanche search and rescue operations with UAV imagery, *Remote Sens.*, 2017, doi: 10.3390/rs9020100.
- [18] H. A. Pratiwi, M. Cahyanti, and M. Lamsani, Implementasi Deep Learning Flower Scanner Menggunakan Metode Convolutional Neural Network, *Sebatik*, vol. 25, no. 1, 2021, doi: 10.46984/sebatik.v25i1.1297.
- [19] M. F. Naufal, "Analisis Perbandingan Algoritma SVM, KNN, dan CNN untuk Klasifikasi Citra Cuaca," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 8, no. 2, 2021, doi: 10.25126/jtiik.2021824553.
- [20] A. Krizhevsky *et al.*, "ImageNet Classification with Deep Convolutional Neural Networks Alex," *Proc. 31st Int. Conf. Mach. Learn.*, 2012, doi: 10.1007/s13398-014-0173-7.2.
- [21] B. Yanto, J. Jufri, A. Lubis, B. H. Hayadi, and E. Armita, NST, Klarifikasi Kematangan Buah Nanas Dengan Ruang Warna Hue Saturation Intensity (HSI)," *INOVTEK Polbeng - Seri Inform.*, vol. 6, no. 1, 2021, doi: 10.35314/isi.v6i1.1882.
- [22] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning (Adaptive Computation and Machine Learning series)*. 2016.
- [23] S. Ye, Z. Zhang, X. Song, Y. Wang, Y. Chen, and C. Huang, A Flow Feature Detection Method For Modeling Pressure Distribution Around A Cylinder In Non-Uniform Flows By Using A Convolutional Neural Network, *Sci. Rep.*, Vol. 10, No. 1, 2020, doi: 10.1038/s41598-020-61450-z.
- [24] Rismiyati, Implementasi Convolution Neural Network untuk Sortasi Mutu Salak Ekspor Berbasis Citra Digital, 2016.
- [25] T. Zhi, L. Y. Duan, Y. Wang, and T. Huang, Two-Stage Pooling Of Deep Convolutional Features For Image Retrieval, 2016, doi: 10.1109/ICIP.2016.7532802.
- [26] T. Shafira, Implementasi Convolutional Neural Networks Untuk Klasifikasi Citra Tomat Menggunakan Keras, 2018.
- [27] Y. Yuan, Z. Xiong, and Q. Wang, Acm: Adaptive Cross-Modal Graph Convolutional Neural Networks For Rgb-D Scene Recognition, 2019, doi: 10.1609/aaai.v33i01.33019176.
- [28] J. Jebadurai, I. J. Jebadurai, G. J. L. Paulraj, and N. E. Samuel, "Learning Based Resolution Enhancement of Digital Images," *Int. J. Eng. Adv. Technol.*, vol. 8, no. 6, 2019, doi: 10.35940/ijeat.F9025.088619.
- [29] A. Coates, H. Lee, and A. Y. Ng, An Analysis Of Single-Layer Networks In Unsupervised Feature Learning, 2011.
- [30] K. Parthy, CS231n Convolutional Neural Networks for Visual Recognition, *Stanford Univ. Course cs231n*, 2018.