

Identifikasi Buah-buahan Menggunakan Metode Convolutional Neural Network

Verrell, Kartika Gunadi, Endang Setyati

Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Kristen Petra

Jl. Siwalankerto 121 – 131 Surabaya 60236

Telp. (031) – 2983455, Fax. (031) – 8417658

E-Mail: verrell.sby@gmail.com, kgunadi@petra.ac.id, endang@stts.edu

ABSTRAK

Buah-buahan memiliki jenis yang cukup banyak sehingga terkadang sulit untuk dibedakan oleh manusia. Dengan perkembangan teknologi yang pesat terutama dalam bidang *artificial neural network*, maka dikembangkanlah sebuah program untuk melakukan identifikasi buah-buahan dari gambar dengan bantuan *Tensorflow* dalam membuat sebuah model *artificial neural network*.

Metode yang digunakan adalah *convolutional neural network* untuk melakukan *training* pada model. Terdapat 4 proses utama dalam *convolutional neural network* yaitu *convolution layer*, *activation layer*, *pooling layer*, dan *fully connected layer*. Dalam pembuatan model *convolutional neural network* dapat dilakukan *tuning* dalam berbagai macam hal, beberapa diantaranya yaitu jumlah *epoch*, *activation function* yang digunakan, dan juga ukuran *learning rate*.

Penelitian ini menunjukkan bahwa jumlah *epoch* mempengaruhi akurasi pada model yang dibuat. Dalam hal *activation function*, ditunjukkan bahwa *rectified linear unit (RELU)* lebih baik dalam hal waktu yang dibutuhkan dalam membuat *neural network model* yang paling optimal dibandingkan dengan *exponential linear unit (ELU)*. Besar kecilnya *learning rate* juga mempengaruhi seberapa baik model yang dihasilkan. Dari berbagai pengujian yang dilakukan didapatkan akurasi terbaik yang dapat dicapai sebesar 99,31%.

Kata Kunci: *Artificial neural network, Tensorflow, convolutional neural network, rectified linear unit, exponential linear unit*

ABSTRACT

Fruits have so many variation, so sometimes its hard to distinguish by humans. Advancement in technology especially in artificial neural network, so a program is develop to identify fruits from pictures with the help of *Tensorflow* in making the neural network model.

The method used is convolutional neural network for training the model. 4 main processes in convolutional neural network are convolution layer, activation function layer, pooling layer, and fully connected layer. As for making the convolutional neural network model, tuning can be carried out in various ways, some of which are epoch size, activation function used, and learning rate size too.

The results shows that epoch size affect the accuracy of the model. In terms of activation function, rectified linear unit (RELU) is better in term of time needed to make the most optimal neural network model compared to the exponential linear unit (ELU). The size of learning rate also affect the result of the model. From

the various tests performed, the best accuracy can be achieved as big as 99,31%.

Keywords: *Artificial neural network, Tensorflow, convolutional neural network, rectified linear unit, exponential linear unit.*

1. PENDAHULUAN

Identifikasi objek buah-buahan melalui gambar sangat bermanfaat karena banyak sekali jenis-jenis buah yang ada dan membuat bingung dalam membedakannya. Untuk menyelesaikan masalah tersebut dibutuhkan aplikasi untuk memisahkan objek-objek yang akan diidentifikasi dari *background* sekaligus mengidentifikasi objek-objek tersebut.

Selective search algorithm adalah salah satu algoritma yang dapat digunakan untuk melakukan segmentasi gambar. Segmentasi gambar dilakukan untuk membagi gambar menjadi beberapa bagian. Hal ini sangat bermanfaat dalam melakukan deteksi objek tertentu pada gambar. Dengan segmentasi objek tersebut dapat dipisahkan dengan gambar *background*, sehingga dapat dimanfaatkan untuk analisa atau proses lebih lanjut[7].

Convolutional Neural Network (CNN) adalah salah satu metode yang efektif digunakan oleh komputer untuk melakukan *learning* dari gambar. Dengan metode ini, komputer nantinya bisa melakukan identifikasi objek dari gambar yang diberikan. Setiap objek pada gambar mempunyai fitur-fitur khusus (seperti warna, ukuran, tekstur), sehingga untuk mendapatkan fitur-fitur dari gambar tersebut sangat penting dilakukannya *convolution* dengan *filter* yang sesuai[1].

Pada penelitian yang dilakukan oleh Muresan, Horea, et. al[4] digunakan *convolutional neural network* untuk mengidentifikasi dengan batasan 1 buah. Sedangkan pada penelitian ini akan menggabungkan kedua metode diatas untuk mengidentifikasi buah-buah yang dapat ditemukan di dalam gambar.

2. DASAR TEORI

2.1 Selective Search Algorithm

Merupakan salah satu *region proposal algorithm* yang melakukan perhitungan dan pengelompokan berdasarkan warna, tekstur, ukuran, dan bentuknya. *Region proposal* berarti daerah pada gambar dengan probabilitas mengandung objek. *Selective search* dilakukan dengan melakukan *over* segmentasi ada gambar berdasarkan tingkat intensitas dari *pixel* menggunakan metode *Felzenszwalb* dan *Huttenlocher*. Ada 4 hal yang diperhatikan dalam algoritma ini :

1. Kemiripan warna

Diawali dengan membuat 25 *bins* histogram warna untuk masing-masing channel warna (*Red, Green, Blue*). Kemudian dilanjutkan

dengan menghitung kemiripan warna untuk setiap *region* yang ada pada gambar dengan menggunakan persamaan 1.

$$S_{color}(r_i, r_j) = \sum_{k=1}^n \min(c_i^k, c_j^k) \quad (1)$$

2. Kemiripan tekstur

Fitur tekstur akan dihitung menggunakan *gaussian derivative* dengan 8 orientasi untuk masing-masing *channel* (Red, Green, Blue). Lalu akan dibuat 10 *bins* histogram untuk setiap orientasi dan *channel*. Setelah itu akan dihitung juga untuk kemiripan tekstur dari setiap *region* yang ada pada gambar menggunakan persamaan 2.

$$S_{texture}(r_i, r_j) = \sum_{k=1}^n \min(t_i^k, t_j^k) \quad (2)$$

3. Kemiripan ukuran

Kemiripan ukuran membuat *region-region* dengan ukuran kecil agar bisa segera menyatu, dimana dalam melakukan penghitungan kemiripan ukuran akan digunakan persamaan 3.

$$S_{size}(r_i, r_j) = 1 - \frac{size(r_i) + size(r_j)}{size(im)} \quad (3)$$

4. Kemiripan bentuk

Persamaan 4 digunakan untuk menghitung kemiripan bentuk dengan mengukur seberapa kompatibel suatu *region* dengan *region* yang lain. Kedua *region* tidak akan digabungkan apabila tidak saling bersentuhan.

$$S_{fill}(r_i, r_j) = 1 - \frac{size(BB_{ij}) - size(r_i) - size(r_j)}{size(im)} \quad (4)$$

Pada penelitian yang dilakukan oleh Hou, Lei et. al[3], *selective search algorithm* digunakan untuk mengurangi waktu yang dibutuhkan dalam melakukan segmentasi gambar dan meningkatkan akurasi dibanding dengan metode *window search*[7].

2.2 Convolution Neural Network

Convolutional Neural Network merupakan salah satu kategori dalam *neural network* yang terbukti sangat efektif untuk *image recognition* dan *image classification*. *CNN* sukses dalam melakukan identifikasi wajah, objek seperti rambu-rambu lalu lintas yang digunakan juga untuk *self driving car*. Oleh karena itu *CNN* merupakan alat yang sangat penting untuk *machine learning research*. Ada 4 proses penting di dalamnya :

1. Convolution

Digunakan untuk mendapatkan fitur-fitur dari gambar (seperti warna, tekstur, garis tepi). Menggunakan *matrix image* akan dilakukan konvolusi menggunakan *matrix filter* seperti pada Gambar 1. Setelah dilakukan konvolusi hasil dari konvolusi tersebut merupakan sebuah *matrix* yang disebut dengan *matrix feature map* seperti pada Gambar 2.

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

Gambar 1. Konvolusi pada matriks gambar [2]

4	3	4
2	4	3
2	3	4

Gambar 2. Matrix feature map [2]

Penggunaan *filter* yang berbeda akan menghasilkan perbedaan gambar sehingga *feature* yang didapatkan dari gambar juga berbeda.

Dalam melakukan *convolution* terdapat 3 parameter utama, yaitu:

a. Depth

Menunjukkan jumlah *filter* yang digunakan dalam melakukan *convolution*, apabila jumlah *filter* yang digunakan 3 maka akan dihasilkan *feature map* untuk masing-masing *filter*.

b. Stride

Menunjukkan jumlah pergeseran *matrix filter* pada *matrix image* untuk dilakukan *convolution*.

c. Zero-padding

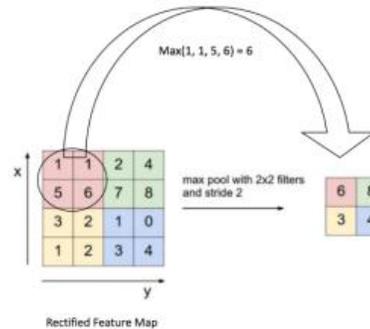
Menunjukkan apakah ketika dilakukan *convolution* pada bagian tepi gambar diberi angka 0. Parameter ini sekaligus mengatur ukuran dari *feature map* yang dihasilkan.

2. Activation function

Activation function digunakan untuk membuat hasil dari *feature map* menjadi *non-linear*. Pada RELU mengubah setiap nilai *pixel* pada *feature map* yang negatif menjadi 0.

3. Pooling

Pooling dilakukan untuk mengurangi dimensi dari *feature map* dengan hanya menyimpan informasi-informasi penting dari *feature map* tersebut (*pooling* bisa menggunakan *SUM*, *MAX*, *AVERAGE*, dll). Pada Gambar 3 ditunjukkan *MAX pooling* terhadap *feature map* yang didapatkan setelah melewati *activation function layer*.



Gambar 3. Contoh max pooling [2]

4. Fully connected layer

Fully connected layer mempunyai struktur seperti *multi layer perceptron*, dimana pada *output layer* biasanya digunakan *softmax activation function*. Hasil dari *pooling* akan menjadi *input* pada *layer* ini, dimana hasil tersebut merepresentasikan fitur-fitur yang sangat penting dari suatu gambar sehingga dapat digunakan untuk melakukan klasifikasi sesuai dengan kelas-kelas yang ada pada *dataset*. Digunakannya *softmax* agar hasil *output* dari *layer* ini

jumlahnya adalah 1, sehingga nilai-nilai dari masing-masing kelas sesuai yang ada pada *dataset* jika dijumlahkan nilainya adalah 1. Nilai terbesar yang dihasilkan pada *output layer* menunjukkan gambar yang menjadi *input* termasuk jenis buah apa[2].

Menurut hasil penelitian dari Shah, Anish, et. al[6] ditunjukkan bahwa *activation function ELU* dapat memberikan hasil yang lebih baik dan waktu yang lebih singkat dalam mencapai hasil tersebut dibanding *RELU*.

2.3 Python

Python adalah salah satu bahasa pemrograman yang dibuat oleh Guido van Rossum dan pertama kali dirilis pada tahun 1991. Fitur-fitur yang terdapat pada Python hampir sama dengan bahasa pemrograman pada umumnya dengan kelebihan pada library yang banyak dan komprehensif. Python mendukung hampir semua operating system yang sudah ada saat ini. Selain itu, Python sendiri merupakan open source software sehingga dapat didapatkan dengan mudah[5].

2.4 Tensorflow

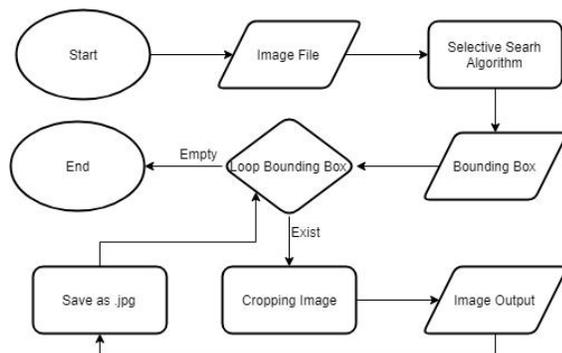
Tensorflow adalah salah satu open source library untuk pemrograman berbasis dataflow. Mempunyai banyak library untuk matematika dan sering digunakan untuk pengaplikasian machine learning seperti neural networks. Tensorflow sendiri dikembangkan oleh tim Google Brain untuk kegunaan internal[8].

3. ANALISIS SISTEM

Analisis sistem ini akan membahas proses-proses yang dilakukan pada data yang ada. Sistem meliputi *Preprocessing Image*, Model *Neural Network* untuk pengambilan fitur dan training, Sistem untuk *Testing*, dan Sistem untuk Aplikasi Identifikasi Buah-buahan.

3.1 Preprocessing Image dan Dataset

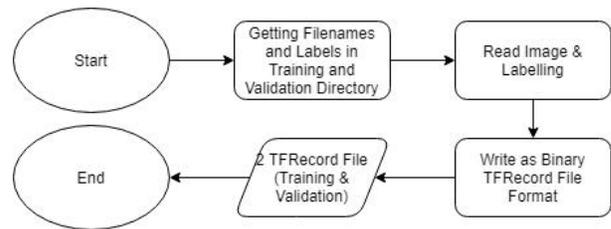
Preprocessing image dilakukan pada gambar inputan ke aplikasi akhir. Dimana akan dilakukan metode *selective search* terlebih dahulu untuk mendapatkan *bounding box* dari setiap objek yang ditemukan. Setelah itu akan dilakukan *cropping* pada gambar sebanyak *bounding box* dan setiap hasilnya akan disimpan dalam format .jpg/.jpeg. *Flowchart* proses ini dapat dilihat pada Gambar 4.



Gambar 4. Flowchart preprocessing image

Dataset yang digunakan berupa gambar dengan jumlah yang banyak. Oleh sebab itu untuk mempercepat proses pembacaan *file* dilakukan *preprocessing* dengan membaca seluruh *file* gambar *dataset* dari *folder dataset* untuk *training* dan juga *validation*. Gambar dan label yang dibaca akan ditulis menjadi sebuah *file*

biner sesuai format untuk TFRecord. *Output* berupa *file* TFRecord masing-masing untuk *training* dan juga *validation*. Proses *preprocessing dataset* dapat dilihat pada Gambar 5.



Gambar 5. Flowchart preprocessing dataset

3.2 Neural Network Model

Dataset berupa gambar berukuran 100x100 *pixel*. Neural Network sendiri terdiri dari 3 *layer* utama yaitu *input layer*, *hidden layer*, dan *output layer*. Pada setiap *layer* terdapat *weight* dan *bias* yang sangat berpengaruh kepada hasil *output* yang diberikan. Karena model yang digunakan adalah *convolutional neural network* maka *weight* pada setiap *layer* merupakan *filter* yang nantinya akan dikonvolusi pada gambar input.

Jaringan *neural network* yang akan dibuat terdiri dari 3 bagian *layer* yaitu *convolutional layer*, *fully connected layer*, dan *classification layer*.

Pada *convolutional layer*, dilakukan konvolusi dengan jumlah *filter* yang sudah ditentukan termasuk ukuran sisi *filter*, *stride* (jumlah pergeseran *filter*), *padding* (penggunaan angka 0 pada bagian tepi gambar atau tidak), dan ukuran serta metode *pooling* yang digunakan.

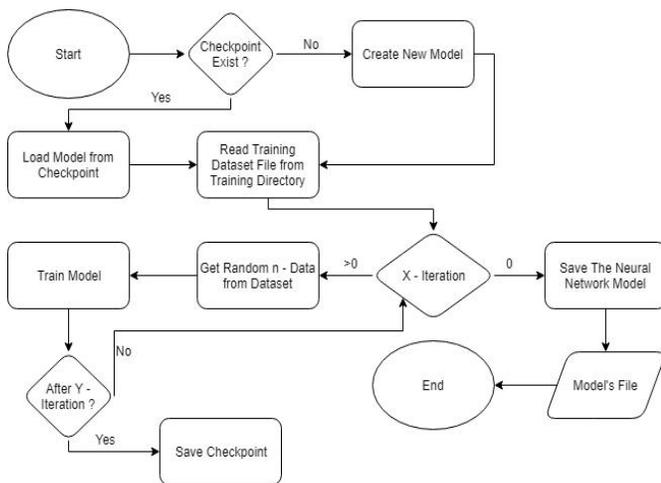
Pada bagian *fully connected layer* diterapkan *activation function* pada hasil gambar dari *output convolutional layer*. Dengan *activation function* akan dihasilkan *feature map* yang sifatnya *non linear*.

Terakhir pada *classification layer* diterapkan *softmax function* untuk memperoleh distribusi probabilitas sejumlah *class* yang diinginkan. Dari distribusi tersebut akan diambil probabilitas *class* yang tertinggi sebagai *output* dari *neural network*.

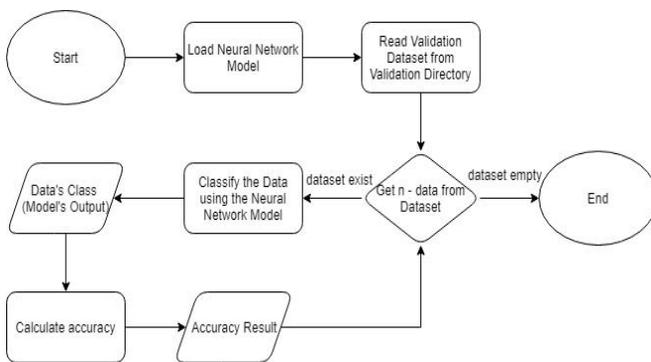
3.3 Sistem Training dan Validation

Sistem *training* dan *validation* dibuat menjadi dua bagian. Pada bagian *training* dilakukan *backpropagation* dengan *Adam Optimizer* untuk mendapatkan hasil *weight* dan *bias* yang optimum secara berulang sebanyak N kali iterasi. Pada Gambar 6 diberikan *flowchart* untuk proses *training*. Saat *training* dapat dilakukan pembuatan *neural network model* baru atau melakukan restorasi *neural network model* yang lama untuk dilakukan *training* lagi. Dimana setelah mencapai N - iterasi maka *neural network model* akan di *save*. Proses *training* dilakukan dengan membaca *dataset training* lalu dijadikan *input* pada *neural network model*. Pada setiap iterasi *neural network model* akan *learning* dengan melakukan perubahan pada nilai *weight* yang ada pada *neural network model* sehingga mampu menghasilkan *label* yang sesuai dengan jenis buah.

Proses *validation* ditunjukkan pada *flowchart* di Gambar 7. Proses *validation* diawali dengan mengecek apakah *neural network model* yang sudah ada. Lalu *model* itu akan direstorasi. Kemudian *dataset* untuk *validation* akan dibaca dan dijadikan *input* pada *neural network model*. Setiap gambar pada *dataset* akan diidentifikasi satu per satu dan dihitung jumlah salahnya sehingga menjadi nilai akurasi untuk *model* tersebut.



Gambar 6. Flowchart training



Gambar 7. Flowchart validation

3.4 Sistem Program Identifikasi Buah-Buahan

Pertama aplikasi akan menerima input berupa gambar. Lalu akan dilakukan *preprocessing image* seperti yang telah dijelaskan sebelumnya sehingga menghasilkan banyak gambar untuk masing-masing buah yang terdeteksi. Gambar hasil *preprocessing* akan di simpan pada suatu *folder*. Kemudian *model neural network* akan di-load lalu gambar hasil *preprocessing* pada *folder* akan dibaca satu per satu. Hasil output berupa *label* jenis buah yang akan dituliskan pada sebuah *file* beserta dengan nama *file* gambar yang diidentifikasi. Setelah semua gambar sudah selesai diidentifikasi maka *file* hasil dibaca terlebih dahulu lalu gambar dan *label* pada *file* akan ditampilkan pada aplikasi.

4. IMPLEMENTASI SISTEM

Implementasi sistem dilakukan pada computer dengan spesifikasi:

- RAM: 4GB, DDR3
- HDD: 1 TB
- CPU: Core i5
- OS Windows

Implementasi pengkodean sistem, menggunakan Bahasa pemrograman Python dengan versi 3.0 *Framework* yang digunakan untuk sistem ini adalah Flask *Micro-Framework*. Adapun beberapa *library* yang mendukung sistem ini adalah:

- Matplotlib
- Scikit-learn
- Flask-CORS
- Selective search
- Tensorflow

5. PENGUJIAN SISTEM

Pada bab ini akan dibahas tentang pengujian terhadap *model* yang telah dibuat sebelumnya pada saat proses *training*. Pengujian dilakukan menggunakan *dataset* yang sudah dipersiapkan sebelumnya.

5.1 Pengujian Sistem Neural Network

Pengujian sistem *neural network model* dilakukan menggunakan *model neural network* yang telah dibuat dengan struktur yang sama seperti pada Subbab 3.2. Adapun untuk *ELU* dan *RELU* akan menggunakan struktur *model* yang sama jadi perbedaannya hanya pada *activation function* yang digunakan.

5.1.1 Pengujian Berulang

Pembuatan *neural network model* diawali dengan *weight* yang nilainya *random* dengan standar deviasi tertentu. Oleh sebab itu akan menghasilkan tingkat akurasi yang berbeda-beda pula. Pengujian dilakukan sebanyak 3 kali untuk setiap jenis *activation function* yaitu *ELU* dan *RELU*. Akurasi pengujian berulang dapat dilihat pada Tabel 1.

Tabel 1. Akurasi pengujian berulang

Activation Function	Jumlah Testing	Jumlah Benar	Jumlah Salah	Akurasi (%)
ELU - 1	5906	5758	148	97,49
ELU - 2	5906	5752	154	97,39
ELU - 3	5906	5745	161	97,27
RELU - 1	5906	5768	138	97,66
RELU - 2	5906	5830	76	98,71
RELU - 3	5906	5778	128	97,83

Selain hasil akurasi yang berbeda jumlah iterasi dan waktu yang dibutuhkan suatu *model* dalam mencapai titik akurasi maksimum pertama kali juga berbeda karena inisialisasi *weight* yang berbeda. Pada Tabel 2 dapat dilihat perbedaan waktu yang dibutuhkan.

Tabel 2. Waktu training hingga maksimum pertama kali

Activation Function	Iterasi	Waktu (detik)	Rata-Rata Waktu (detik)
ELU - 1	225	95	0,422
ELU - 2	450	179	0,398
ELU - 3	275	97	0,353
RELU - 1	200	76	0,380
RELU - 2	175	61	0,349
RELU - 3	250	103	0,412

Waktu yang dibutuhkan dalam melakukan *training* juga berbeda-beda karena terpengaruh oleh beberapa faktor yang mungkin terjadi dalam proses yang sedang dikerjakan oleh komputer

ataupun hal lain. Oleh karena itu diberikan pula tabel waktu yang dibutuhkan dalam setiap pengulangan *training* pada Tabel 3.

Tabel 3. Waktu training hingga selesai

Activation Function	Iterasi	Waktu (detik)	Rata-Rata Waktu (detik)
ELU - 1	5000	1790	0,358
ELU - 2	5000	1610	0,322
ELU - 3	5000	1605	0,321
RELU - 1	5000	1598	0,320
RELU - 2	5000	1575	0,315
RELU - 3	5000	1592	0,318

5.1.2 Hasil Pengujian dengan ELU

Berikut akan diberikan beberapa contoh gambar yang dapat diidentifikasi dengan benar pada *ELU model* yang telah dibuat. Contoh gambar dapat dilihat pada Tabel 4.

Tabel 4. Contoh gambar benar

Jenis Buah	Apple Red	Avocado	Grape Red
Gambar			
Output	Apple Red	Avocado	Grape Red

Dari hasil pengujian terdapat pula beberapa jenis buah yang salah diidentifikasi oleh *neural network model ELU* yang telah dibuat. Beberapa gambar yang salah diidentifikasi dapat dilihat pada Tabel 5.

Tabel 5. Contoh gambar salah

Jenis Buah	Melon	Pear	Red Dragon Fruit
Gambar			
Output	Grape White	Apple Golden	Banana Red
Gambar Perbandingan			

5.1.3 Hasil Pengujian dengan RELU

Pada Tabel 6 diberikan beberapa contoh gambar buah yang berhasil diidentifikasi dengan benar oleh *RELU model*.

Tabel 6. Contoh gambar benar

Jenis Buah	Lemon	Orange	Pepino
Gambar			
Output	Lemon	Orange	Pepino

Model yang dibuat juga masih melakukan kesalahan dalam mengidentifikasi beberapa jenis buah. Beberapa *sample* gambar yang salah diidentifikasi oleh *RELU model* dapat dilihat pada Tabel 7.

Tabel 7. Contoh gambar benar

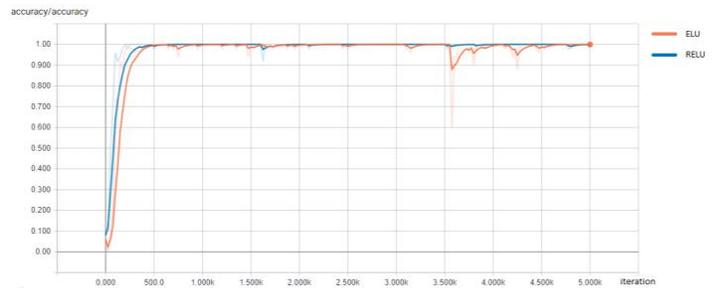
Jenis Buah	Apple Red	Banana	Cocos
Gambar			
Output	Apple Golden	Banana Red	Salak
Gambar Perbandingan			

5.2 Perbandingan Hasil ELU dan RELU

Pada bagian ini akan disajikan data-data perbandingan antara hasil uji pada *model ELU* dengan *model RELU*.

5.2.1 Perbandingan Akurasi Saat Training

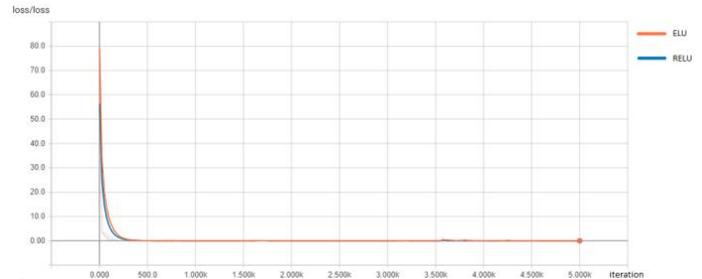
Pada saat *training* dilakukan pencatatan nilai akurasi setiap 25 iterasi. Grafik perbandingan nilai akurasi antara kedua *model* dapat dilihat pada Gambar 8.



Gambar 8. Gambar akurasi saat training

5.2.2 Perbandingan Loss Saat Training

Selain pencatatan nilai akurasi juga dilakukan pula pencatatan nilai *loss* setiap 25 iterasi. Grafik perbandingan nilai *loss* antara kedua *model* dapat dilihat pada Gambar 9.



Gambar 9. Gambar loss saat training

5.2.3 Perbandingan Rata-Rata Akurasi Testing

Dari hasil *testing* berulang yang dapat dilihat pada Tabel 1 akan dirata-rata untuk setiap *activation function*. Pada Tabel 8 dapat dilihat perbandingan rata-rata akurasi yang lebih baik dari hasil *testing* antara kedua *model*.

Tabel 8. Perbandingan akurasi testing

Activation Function	Ke 1	Ke 2	Ke 3	Rata-Rata Akurasi (%)
ELU	97,49	97,39	97,27	97,39
RELU	97,66	98,71	97,83	98,07

Dari tabel tersebut dapat dilihat bahwa *RELU* mempunyai akurasi yang lebih baik dibandingkan dengan *ELU*.

5.2.4 Perbandingan Rata-Rata Waktu Training

Pada Tabel 9 diberikan perbandingan waktu yang dibutuhkan dengan kondisi kedua model mencapai nilai akurasi maksimum untuk pertama kalinya.

Tabel 9. Perbandingan waktu mencapai akurasi maksimum

Activation Function	Ke 1	Ke 2	Ke 3	Rata-Rata Waktu
ELU	0,422	0,398	0,353	0,391
RELU	0,380	0,349	0,412	0,380

Pada Tabel 10 diberikan perbandingan waktu yang dibutuhkan dengan kondisi kedua model selesai melakukan *training*.

Tabel 10. Perbandingan waktu selesai training

Activation Function	Ke 1	Ke 2	Ke 3	Rata-Rata Waktu
ELU	0,358	0,322	0,321	0,334
RELU	0,320	0,315	0,318	0,318

Dari Tabel 9 dan Tabel 10 dapat dilihat bahwa *RELU* membutuhkan waktu yang lebih singkat daripada *ELU* dalam melakukan *training*.

6. KESIMPULAN DAN SARAN

Dari hasil pembuatan aplikasi dan *neural network model* baik dengan *activation function ELU* maupun *RELU* untuk identifikasi jenis buah-buahan, dapat diambil kesimpulan antara lain :

- *Selective search* yang diterapkan terbatas pada ukuran tertentu dalam mencari objek yang terdapat dalam gambar.
- Sistem *training* diuji dengan melihat perkembangan akurasi dari setiap iterasi.
- Sistem *testing* dilakukan menggunakan gambar-gambar yang sudah disisihkan sebelumnya (berbeda dengan data untuk *training*) dan diukur dengan melihat hasil akurasi.
- Tingkat akurasi saat *testing* pada pembuatan *convolutional neural network model* sangat bergantung pada *dataset* yang digunakan untuk *training*. Semakin banyak variasi pada *dataset training* maka semakin banyak pula variasi yang dapat dikenali namun juga akan menambah ambiguitas pada *convolutional neural network model*.
- Fitur-fitur seperti warna, ukuran, densitas, *noise* pada gambar sangat berpengaruh pada hasil *convolutional neural network model* yang akan dibuat.
- Pembuatan struktur *neural network model* terbatas pada *resource memory* yang terdapat pada komputer yang digunakan.

- Aplikasi berbasis *web* dibuat dengan *flask* yang terdiri dari 2 halaman, halaman utama untuk *input* gambar dan halaman kedua untuk menampilkan hasil.
- Aplikasi bersifat *local* sehingga tidak dapat diakses dari luar.
- Hasil *convolutional neural network model* yang dibuat mampu mencapai akurasi 99,31% dalam mengidentifikasi gambar-gambar pada *dataset testing*.

Saran yang diberikan untuk penyempurnaan dan pengembangan lebih lanjut untuk program ini adalah sebagai berikut :

- Penggunaan *selective search* untuk segmentasi gambar sehingga mendapatkan objek-objek yang terdapat di dalamnya dapat diganti dengan metode yang lebih baik yaitu *RPN (Region Proposal Network)* sehingga pengenalan objek tidak lagi dibatasi pada ukuran melainkan dengan probabilitas yang dihitung untuk setiap daerah pada gambar apakah mengandung objek atau tidak.
- Dapat dilakukan penambahan *dataset* sehingga akan memperluas jumlah variasi yang dapat dikenali *neural network model*.
- Dapat dilakukan modifikasi pada struktur *neural network model* (seperti penambahan *convolution layer*, pengurangan *pooling layer*, mengubah *activation function* yang digunakan) untuk meningkatkan akurasi *neural network model*.
- Dapat dibuat aplikasi berbasis *mobile* dimana gambar objek buah juga dapat diambil langsung melalui kamera untuk diidentifikasi.

7. DAFTAR PUSTAKA

- [1] Dertat, Arden. 2017. 8 November 2017. Applied Deep Learning Part - 4 : Convolutional Neural Network. URI = <https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2>
- [2] Gonzales, Juan. 2017. 20 November 2017. Use Of Convolutional Neural Network For Image Classification. URI = <https://www.apsl.net/blog/2017/11/20/use-convolutional-neural-network-image-classification/>
- [3] Hou, Lei , Wu, QingXiang , et.al. 2016. Fruit Recognition Based On Convolutional Neural Network. International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery.
- [4] Muresan, Horea , Oltean, Mihai. 2017. Fruit Recognition Using Deep Learning. Faculty of Computer Science.
- [5] Python. Python, Programming Language. URI = <https://www.python.org>
- [6] Shah, Anish , Shinde, Sameer , et.al. 2016. Deep Residual Networks with Exponential Linear Unit.
- [7] Singh, Vaibhaw. 2017. 18 September 2017. Selective Search for Object Detection. URI = <https://www.google.co.id/amp/s/www.learnopencv.com/selective-search-for-object-detection-cpp-python/amp/>
- [8] TensorFlow. TensorFlow, Open Source Machine Learning. URI = <https://www.tensorflow.org>