

Implementasi Convolutional Neural Network Untuk Klasifikasi Varietas Pada Citra Daun Sawi Menggunakan Keras

Implementation of Neural Network Convolutionals For Classification of Variety on Image of Collards Meat Leaves Using The Keras

Ahmad Kurniadi¹, Kusrini², Moh. Fal Sadikin³

^{1,2} Magister Teknik Informatika, Fakultas Ilmu Komputer, Universitas AMIKOM Yogyakarta
E-mail: ¹ahmad.kurniadi@students.amikom.ac.id, ²kusrini@amikom.ac.id, ³fal_sadikin@yahoo.com

Abstrak

Perkembangan pertanian di Indonesia membuat Indonesia memiliki banyak varietas sayuran salah satunya sayuran yang banyak diminasi dikalangan masyarakat yaitu sayur sawi. Tapi sayangnya banyak masyarakat kita yang masih sulit untuk membedakan atau mengklasifikasi berbagai macam sawi. Hal ini akan menjadi masalah karena ketidaktahuan jika terus berlanjut maka akan tidak ada yang melestarikan. Permasalahan ini bisa diatasi dengan menggunakan metode deep learning salah satunya yaitu metode convolutional neural network untuk mengklasifikasi citra gambar sawi yang mampu melakukan proses pembelajaran secara sendiri dalam pengenalan sebuah gambar. Penelitian ini menghasilkan data uji coba dari klasifikasi citra pada sayuran varietas sawi yaitu sawi pakcoy, sawi putih, dan sawi caisim menghasilkan nilai akurasi sebesar 83%, recall 80 % dan presisi 89%. Ketika data training dilakukan perbedaan dengan jumlah perbandingannya maka hasil yang didapatkan adalah sama yaitu 83%

Kata Kunci — Klasifikasi, Sawi, CNN

Abstract

The development of agriculture in Indonesia makes Indonesia has many varieties of vegetables, one of which is the most popular vegetable among the people, namely mustard greens. But unfortunately many of our society is still difficult to distinguish or classify various kinds of mustard. This will be a problem because ignorance if it continues then there will be nothing to preserve. This problem can be overcome by using deep learning methods, one of them is the convolutional neural network method to classify mustard images that are able to do the learning process by themselves in the introduction of an image. This research resulted in trial data from image classification on mustard vegetable varieties namely mustard greens, mustard white, and mustard caisim produce accuracy values of 83%, 80% recall and 89% precision. When the training data is different from the number of comparisons, the results obtained are the same, namely 83%

Keywords — Classification, Collards, CNN

PENDAHULUAN

Menurut data statistik holtikura Daerah Istimewa Yogyakarta luas panen sayuran potensi tanaman sawi pada tahun 2016 adalah 539 hektar dan tahun 2017 adalah 438 hektar artinya luas panen sawi mengalami penurunan sebesar 101 hektar (18,74%). Sedangkan pada tahun 2018

menjadi 397 hektar artinya luas panen sawi mengalami penurunan lagi sebesar 41 hektar (9,36%).

Pada tahun 2017, secara umum produksi tanaman sayur-sayuran mengalami penurunan yang sangat signifikan. Diantaranya sawi dari jumlah produksi tahun 2016 sebesar 39.105,

ditahun 2017 turun menjadi 36.305, dan ditahun 2018 turun menjadi 32.047 (BPS, 2018). Kondisi dimana semakin berkurangnya lahan pertanian terutama di daerah perkotaan mendorong berbagai macam teknologi dan inovasi dan dalam mengembangkan bidang pertanian.

Banyak masyarakat yang masih belum mengetahui apa saja jenis dan macam sayuran yang ada di Indonesia ini dikarenakan terdapat beberapa macam jenis sayuran yang hanya mampu tumbuh atau hidup di beberapa bagian wilayah saja atau metode penanaman tertentu saja. Dengan persoalan tersebut, diharapkan para petani melakukan penanaman dalam skala besar sebagai langkah untuk melestarikan keanekaragaman ini. Penanaman skala besar ini nanti juga akan menjadi permasalahan baru jika tidak ditambah dengan teknologi untuk membantu dalam penanaman dalam skala besar tersebut. Maka dari itu, harus diimbangi dengan suatu teknologi menggunakan teknik *deep learning* dengan metode *Convolutional Neural Network* yang mampu digunakan sebagai solusi untuk mengklasifikasikan macam-macam jenis tanaman sayuran sebagai langkah untuk pendeteksian tanaman sayuran khususnya tanaman sawi. Dalam melakukan proses klasifikasi pada tumbuhan dapat dilakukan dengan cara gambar dan bentuk daun pada tumbuhan dilakukan identifikasi. Pola daun dapat dikenali dengan mengenali karakteristik struktural dan seperti tekstur dan bentuk pada daun. Karakteristik struktural daun dapat dilakukan analisa dengan memanfaatkan sebuah memanfaatkan sebuah metode pemrosesan terhadap citra dengan memanfaatkan pengelolaan citra digital. Pesatnya perkembangan teknologi terutama untuk teknik pengelolaan citra memunculkan berbagai pengembangan berbagai tehnik untuk mempermudah manusi baik dari segi pengolahan citra, analisis citra untuk berbagai tujuan dan keperluan. Dalam melakukan pengolahan citra untuk memperoleh hasil yang ideal sering dihadapkan dengan berbagai

masalah yang bisa mempengaruhi hasil interpelasi dan akan mempengaruhi analisa dan perancangan yang akan dilakukan diantaranya seringkali citra yang digunakan tidak dalam kondisi yang ideal untuk dikaji dikarenakan banyaknya gangguan diantaranya foto atau gambar kabur, kurang jelasnya kenampakan obyek dan juga dapat berupa bayangan,

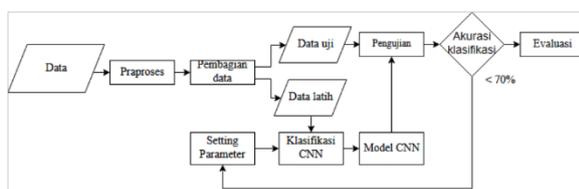
Pengklasifikasian tumbuhan yang cepat akan didapatkan dari Sistem pengenalan dan pengklasifikasian daun tumbuhan yang otomatis. Daun merupakan salah satu ciri tumbuhan yang mudah diamati, unik dan cukup representatif sehingga bisa dijadikan obyek untuk ekstraksi fitur tumbuhan. Hasil klasifikasi tumbuhan dapat dipengaruhi oleh baik buruknya dari hasil ekstraksi fitur objek. Beberapa penelitian yang sudah dilakukan umumnya menggunakan metode jaringan saraf tiruan untuk mengelompokkan bentuk tumbuhan, seperti yang dilakukan Kadir dkk tahun 2011, Husin dkk tahun 2012, Stephen dkk tahun 2011, dan Chaki dkk tahun 2011. Metode jaringan saraf tiruan dikenal lebih cepat secara substansial sehingga banyak yang menggunakan. Akan tetapi dalam menentukan jumlah hidden layer yang digunakan akan berpengaruh pada hasil, serta dibutuhkan parameter jumlah epoch yang besar sehingga membutuhkan komputasi yang lebih tinggi. Penelitian yang dilakukan oleh Arunpriya pada tahun 2012, tentang pengenalan tumbuhan menggunakan metode *Support Vector Machine* (SVM). Metode *Support Vector Machine* (SVM) secara teoritik hanya dapat dikembangkan untuk permasalahan dua kelas dan bersifat *linear classifier*.

Dalam melakukan pengolahan citra banyak metode yang dapat digunakan untuk melakukannya salah satunya adalah menggunakan metode *Convolutional Neural Network* (CNN). Penelitian Suartika tahun 2016 menyebutkan kemampuan dalam mengolah informasi citra dengan CNN berusaha meniru sistem pengenalan citra pada visual cortex yang dimiliki oleh manusia. Beberapa penelitian sebelumnya

mengenai pengolahan citra dengan menggunakan metode CNN mendapatkan hasil akurasi yang bagus, diantaranya penelitian yang dilakukan oleh Budi Setiyono dan Muhammad Zufar pada tahun 2016 untuk pengenalan wajah secara *realtime*. Hasil akurasi yang didapatkan yaitu sebesar 89%. Penelitian terbaru juga dilakukan oleh Kevin Pudi Danokusumo tahun 2017 untuk klasifikasi citra candi berbasis GPU. Hasil pengujian terhadap citra candi menunjukkan akurasi sebesar 98,99% pada training set dan pada test sebesar 85,57% dengan waktu pelatihan mencapai 389,14 detik. Dari beberapa penelitian yang sudah dilakukan menyebutkan bahwa klasifikasi citra dalam melakukan pengenalan tanaman menghasilkan akurasi yang sangat baik dengan teknik *Deep Learning* dengan metode CNN.

Metode Penelitian

Dalam melakukan tahapan penelitian ini ada beberapa tahap yaitu, akuisisi data, praproses data, pembagian data, klasifikasi CNN, pengujian, dan evaluasi yang bisa dilihat pada Gambar 1.



Gambar 1. Alur kerja dari klasifikasi varietas sawi.

2.1. Akuisisi Data/Citra

Proses akuisisi citra daun sawi diambil dengan menggunakan kamera digital. Pengambilan gambar daun sawi dilakukan di dalam ruangan yang sistem pencahayaannya cenderung lebih stabil. Pengambilan citra daun sawi dilakukan dengan menempatkan kamera diatas objek daun dan jarak kamera dengan objek ditentukan dikarenakan ukuran daun dari masing-masing varietas sawi berbeda-beda. Sebuah kertas berwarna putih digunakan sebagai latar belakang dari objek daun. Varietas sawi yang digunakan dalam penelitian ini berjumlah 3 varietas yaitu sawi pakchoy, sawi putih,



dan sawi caisim. Adapun contoh citra tanaman sawi pada Gambar 2.

(a)SawiPutih (b)Sawi Caisim (c)Sawi Pakcoy
Gambar 2. Citra daun sawi

2.2. Setting Parameter

Pada tahap ini nilai parameter telah ditentukan pada saat pembuatan model CNN. Parameter yang digunakan meliputi jumlah *epoch*, jumlah *batch*, ukuran dimensi citra, dimensi citra, jumlah filter, ukuran *convolution layer*, dan ukuran *pooling layer*.

2.3. Praproses

Sebelum dilakukan proses klasifikasi CNN, citra terlebih dahulu dilakukan praproses. Tahapan praproses yang pertama ialah mengubah ukuran piksel citra asli menjadi 32 x 32 piksel. Proses analisis dalam penelitian ini menggunakan bantuan *software Java Script, Python 3.6.4* dan *R Studio 1.1.419*.

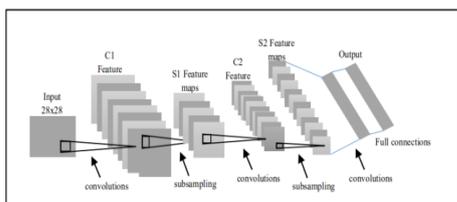
2.4. Pembagian Data

Tahapan selanjutnya setelah dilakukan praproses, semua data citra daun dibagi menjadi dua yaitu data latih dan data uji sesuai prosentase yang sudah ditentukan.

2.5. Klasifikasi CNN

Dalam pemrosesan data struktur kotak (*grid*) hanya jaringan syaraf CNN yang mampu, contohnya yaitu gambar citra yang berbentuk dua dimensi. Layer konvolusi merupakan operasi dari aljabar linear yang mengalikan matriks dari filter pada citra yang akan diproses. Salah satu jenis dari banyak lapisan yang bisa dimiliki dalam suatu jaringan biasanya disebut dengan proses lapisan konvolusi. Lapisan konvolusi ini merupakan lapisan

utama yang paling penting digunakan. Jenis lapisan yang lain yang tak kalah pentingnya yang digunakan adalah *Pooling Layer*, yaitu lapisan yang digunakan untuk mengambil nilai rata-rata atau mengambil nilai maksimal dari bagian-bagian lapisan piksel pada sebuah citra. Penelitian ini menggunakan arsitektur yang dapat dilihat pada Gambar 3.



Gambar 3. Arsitektur CNN

2.5. Pengujian

Pengujian dilakukan dengan memasukkan sebuah citra data uji ke dalam model klasifikasi CNN yang telah dibuat. Citra ini akan masuk ke dalam jaringan model melalui tahapan *fit model* dan akan menghasilkan sebuah output yang dihitung menggunakan bobot yang telah dioptimasi. Sehingga model klasifikasi yang telah dibuat diharapkan dapat mengklasifikasi data uji tersebut ke kelas yang benar. Pengujian ini dilakukan untuk menghitung nilai akurasi yang dihasilkan pada model klasifikasi yang telah dibuat.

2.6. Evaluasi

Output dari penelitian adalah nilai akurasi dimana nilai tersebut yang akan diujikan. Jumlah data uji yang terklasifikasikan benar dibagi dengan jumlah keseluruhan data uji dihitung untuk memperoleh nilai akurasi. Untuk melihat tingkat keberhasilan klasifikasi terhadap model hasil pelatihan dilakukan pengujian pada data uji. Banyaknya data uji yang diprediksi secara benar dan tidak benar oleh model bisa didapatkan dari proses Evaluasi dari kinerja model. Hal ini dapat dihitung menggunakan akurasi yang diformulasikan pada persamaan (1).

$$\text{Akurasi} = \frac{\text{Jumlah data uji yang teridentifikasi benar}}{\text{Jumlah data uji keseluruhan}} \times 100 \%$$

Untuk menganalisis akurasi yang diperoleh, akurasi juga dapat direpresentasikan menggunakan *Confusion matrix*. *Confusion matrix* adalah suatu metode yang biasanya digunakan untuk melakukan perhitungan akurasi. Tabel ini sering digunakan untuk menggambarkan *performance* dari sebuah model klasifikasi. Tabel ini terdiri dari baris dan kolom sebanyak jumlah kelas yang menunjukkan nilai *false positives*, *false negatives*, *true positives*, and *true negatives*. *Confusion matrix* digunakan jika nilai akurasi yang diperoleh tidak tepat akibat dari data yang tidak seimbang. Tabel 1 merupakan sebuah contoh *confusion matrix*.

Tabel 1. *Confusion matrix* dengan 3 kelas

Aktual	C1	C2	C3	Akurasi
C1				
C2				
C3				

HASIL DAN PEMBAHASAN

Pada bagian ini akan diinterpretasikan hasil dari klasifikasi gambar dengan 3 kategori objek citra, yaitu citra sawi pakcoy, sawi putih, dan sawi caisim dengan menggunakan teknik *deep learning* dengan salah satu metode yang digunakan *convolutional neural network* (CNN). Hasil dan pembahasan akan dibagi menjadi 4 bagian, yaitu pengumpulan data citra, histogram citra, *preprocessing* citra, dan pengolahan citra.

3.1. Data Citra

Data citra adalah citra yang digunakan oleh peneliti adalah sawi pakcoy, sawi putih, sawi pagoda, sawi caisim dan sawi samhong. Dalam menggunakan citra, peneliti terlebih dahulu mengetahui karakteristik dari citra yaitu format citra, ukuran dimensi, ukuran *frame* dari citra dll.



Gambar 4. Citra data sebelum di proses

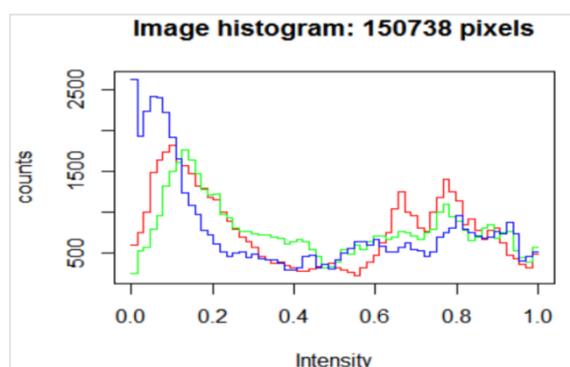
Gambar 4. adalah citra data yang belum mengalami proses *resize*, *combine* dll. Citra diatas memiliki keterangan pada setiap piksel dari citra. Setiap citra memiliki format warna, dimensi dan ukuran *frame*. Dapat dilihat dalam gambar 5.

```
> print(train[[1]])
Image
colorMode : color
storage.mode : double
dim : 259 194 3
frames.total : 3
frames.render : 1

imageData(object)[1:5,1:6,1]
[,1] [,2] [,3] [,4] [,5] [,6]
[1,] 0.6470588 0.6470588 0.6470588 0.6549020 0.6549020 0.6666667
[2,] 0.6470588 0.6470588 0.6509804 0.6549020 0.6627451 0.6666667
[3,] 0.6509804 0.6509804 0.6549020 0.6627451 0.6666667 0.6705882
[4,] 0.6549020 0.6549020 0.6627451 0.6666667 0.6705882 0.6705882
[5,] 0.6627451 0.6627451 0.6666667 0.6705882 0.6705882 0.6823529
```

Gambar 5. Keterangan citra

Berdasarkan gambar 5. pada citra sawi pakcoy diatas yang menjadi sample, citra tersebut memiliki keterangan bahwa berwarna, dengan ukuran dimensi 259 piksel x 194 piksel x 3 adalah format RGB yang dimiliki oleh citra dan data dari citra yang menjelaskan piksel pada citra.



Gambar 6. histogram citra

Histogram citra mendeskripsikan tentang intensitas warna yang ada dalam citra. Format dari *Red*, *Green* and *Blue* (*RGB*) memiliki intensitas yang berbeda, walaupun intensitasnya hampir sama. Dalam format *RGB* masing-masing warna

dinyatakan sebagai komponen dari spektrum dasar.

3.2. Pembuatan Data Latih dan Data Uji

Dalam memahami pola data agar *machine* semakin baik maka dibutuhkan sebuah data latih yang besar. Model *classifier*, dibentuk dari algoritma klasifikasi menggunakan data latih dimana model ini merupakan gambaran pengetahuan yang akan digunakan untuk data baru yang belum pernah ada atau memprediksi kelas data baru. Untuk mengukur sejauh mana *classifier* berhasil melakukan klasifikasi dengan benar maka menggunakan data uji. Data yang digunakan untuk data latih dan data uji adalah data yang telah memiliki label kelas, dengan jumlah perbandingan data latih 80% dan data uji 20%. Perbandingan tersebut merupakan acuan dari berbagai penelitian sebelumnya dalam penentuan jumlah data latih dan data uji pada penelitian analisis citra menggunakan metode *convolutional neural network*.

Data latih dan data uji yang telah dibuat selanjutnya diaplikasikan dalam *software R Studio*. Pada folder dataset citra yang akan digunakan setiap benda diberi nama berdasarkan kategori benda masing-masing untuk mempermudah pengurutan dan pembuatan data uji dan data latih. Untuk pembuatan data latih, *script* yang digunakan adalah *train* dan memasukkan urutan dari dataset citra dari masing-masing benda yang digunakan.

3.3. Resize dan Combine

Proses *resize* dan *combine* dilakukan untuk menyamakan ukuran dari seluruh citra yang akan diproses, dengan dilakukan proses *resize* dan *combine* maka proses analisis akan lebih mudah dan cepat dikarenakan ukuran citra sesuai dengan kemampuan dari perangkat yang digunakan.

Proses *resize* membuat ukuran dari data citra latih dan uji masing-masing berukuran 32 x 32, ukuran citra yang dianjurkan adalah nilai pangkat dua. Nilai yang lebih kecil dari 32 adalah 16 dan yang lebih besar adalah 64. Penggunaan piksel 16x16 membuat informasi yang

diberikan oleh citra akan tidak maksimal, karena bentuk citra terlalu kecil dan banyak informasi yang hilang, sedangkan ukuran 64x64 menjadikan proses citra menjadi lambat dan berat pada perangkat. Sehingga digunakanlah piksel sebesar 32x32 piksel. Selain itu dataset citra dibuat dalam format RGB (*Red, Green, Blue*) yang artinya seluruh dataset citra yang digunakan memiliki warna, bukan *grayscale*. Perubahan ukuran citra menandakan bahwa setiap citra siap untuk dilakukan analisis menggunakan metode *convolutional neural network (cnn)*.

Fungsi dari *script combine* digunakan untuk mengkombinasi dataset latih maupun dataset uji dari hasil perubahan citra, sehingga dapat mengetahui bahwa perubahan ukuran citra dilakukan dengan sempurna pada seluruh dataset citra.

3.4. Reorder Dimension dan Response

Reorder Dimension berfungsi agar penempatan dimensi citra dapat dibaca oleh program dalam urutan yang digunakan, program menginginkan kombinasi jumlah dataset citra latih dan uji, ukuran piksel dan format RGB yang digunakan. Sehingga dilakukan lah *reorder dimension* agar data citra dapat dianalisis.

Response berfungsi untuk memberikan pelabelan dataset citra latih dan uji yang digunakan berdasarkan klasifikasi dataset citra agar dalam pembentukan matrik nantinya dapat dibaca dengan mudah dan benar.

One hot encoding berfungsi untuk melakukan kategori dataset citra latih dan uji berdasarkan label yang telah dibuat.

Pada *script* diatas menggunakan *reorder dimension* untuk membuat dataset citra latih sebanyak 225 citra, dengan piksel ukuran 32 x 32 dan format citra yang digunakan adalah RGB. Dataset citra uji sebanyak 45 citra, dengan piksel ukuran 32x32 dan format citra yang digunakan adalah RGB.

Response melakukan klasifikasi berdasarkan citra dataset uji dan latih, pada data latih klasifikasi sawi pakcoy dengan label 0, sawi putih dengan label 1,

sawi pagoda dengan label 2, sawi caisim dengan label 3 dan sawi samhong dengan label 4 dengan masing-masing klasifikasi sebanyak 36 citra. Pada data uji klasifikasi label yang digunakan sama dengan data training dengan masing-masing klasifikasi sebanyak 9 citra. Setelah melakukan pelabelan maka dilakukan kategorisasi berdasarkan data latih dan data uji.

3.5. Model

Model memuat beberapa *layer* yang berbeda-beda yaitu *layer* konvolusi, *layer pooling*, *layer dropout*, *layer flatten* dan *layer dense*. Selain layer-layer yang digunakan dalam proses *cnn* terdapat pula fungsi aktivasi, dalam studi kasus ini menggunakan fungsi aktivasi *Relu*.

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 30, 30, 32)	896
conv2d_2 (Conv2D)	(None, 28, 28, 32)	9248
max_pooling2d_1 (MaxPooling2D)	(None, 14, 14, 32)	0
dropout_1 (Dropout)	(None, 14, 14, 32)	0
conv2d_3 (Conv2D)	(None, 12, 12, 64)	18496
conv2d_4 (Conv2D)	(None, 10, 10, 64)	36928
max_pooling2d_2 (MaxPooling2D)	(None, 5, 5, 64)	0
dropout_2 (Dropout)	(None, 5, 5, 64)	0
Flatten_1 (Flatten)	(None, 1600)	0
dense_1 (Dense)	(None, 256)	409856
dropout_3 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 3)	771

Total params: 476,195
Trainable params: 476,195
Non-trainable params: 0

Gambar 7. Model Convolutional Neural Network

Model yang dibentuk pada gambar 7. dapat dijelaskan bahwa pada *input* gambar dengan dimensi 32 x 32 x 3 *channel* RGB kemudian dilakukan *convolution* untuk lapisan 1 dengan kernel 3 x 3 dan *filter* 32 lapisan 5 menggunakan aktivasi 'relu' untuk mengambil *nodes* paling tinggi dan meneruskannya untuk *convolution* berikutnya, sehingga *output* gambar menjadi dimensi 30 x 30 x 3 RGB, sehingga keseluruhan parameternya menjadi ((3x3x3)+1 bias) x 32 filter = 896 parameter.

Kemudian, dilanjutkan dengan melakukan *convoluion* untuk lapisan 2 menggunakan *filter* 32 lapisan dan kernel 3 x 3 menggunakan aktivasi 'relu' sehingga *output* dimensi menjadi 28 x 28 x 32 *channel*, sehingga keseluruhan parameternya menjadi ((3x3x32)+1 bias) x 32 filter = 9.248 parameter.

Setelah dilakukan *convolution* 1 dan 2, maka dilanjutkan dengan melakukan proses *pooling* sehingga parameter tidak perlu dihitung karena setiap 2 kotak atau 2 piksel x 2 piksel menjadi 1 piksel x 1 piksel sehingga dari *input shape* 28 piksel x 28 piksel menjadi 14 piksel x 14 piksel.

Proses setelah *pooling*, dilanjutkan dengan proses *dropout* dimana dengan batas nilai *rate* = 0,01 dan untuk nilai yang kurang dari 0,01 maka tidak akan dilanjutkan ke dalam proses selanjutnya untuk didalam nilai pikselnya. Setelah dilakukan proses *dropout* maka dilanjutkan dengan proses *convolution* kembali yaitu dengan *input shape* 14 x 14 x 32 *channel* dengan *filter* 64 lapisan dan kernel 3 x 3 menghasilkan *output* berdimensi 12 x 12 x 64 *channel*, sehingga keseluruhan parameternya menjadi $((3 \times 3 \times 32) + 1 \text{ bias}) \times 64 = 18.496$ parameter.

Kemudian, dilakukan kembali proses *convolution* ke 4 dengan *input shape* 12 x 12 x 32 *channel* RGB menggunakan kernel 3 x 3 dan filter sebanyak 64 lapisan menghasilkan *output* berdimensi 10 x 10 x 64 *channel*, sehingga keseluruhan parameternya menjadi $((3 \times 3 \times 64) + 1 \text{ bias}) \times 64 = 36.928$ parameter.

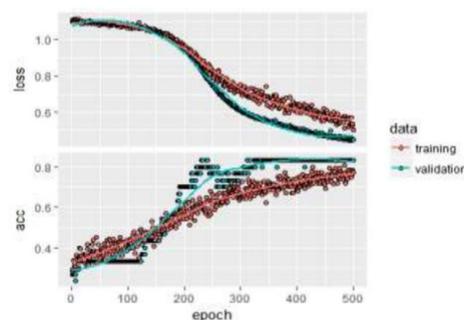
Setelah dilakukan *convolution* 2 kali kemudian dilanjutkan untuk proses *pooling* yang kedua yaitu dengan *input shape* 10 x 10 menjadi 5 x 5 dan tidak ada parameter pada proses ini. Kemudian dilanjutkan dengan proses *dropout* dan dihasilkan sama dengan hasil pada *pooling*. Setelah *pooling* dan *dropout* maka akan terbentuk *layer flatten* yaitu mengubah menjadi nilai vektor dengan 64 *channel* dan ukuran 5 x 5 menjadi bentuk vektor maka tinggal dikalikan menjadi $(5 \times 5 \times 64) = 1.600$, sehingga vektor yang terbentuk ada 1.600 nilai yang masuk ke jaringan syaraf tiruan dimana setiap nilai dari *layer flatten* berasal dari hasil proses konvolusi (*nodes*).

Kemudian, hasil *layer dense* dengan menetapkan sebanyak 256 unit. Hasil vektor sebanyak 1600 nilai akan masuk satu-persatu ke dalam 256 unit maka akan menghasilkan $(1600 \times 256) + 256 \text{ bias} = 409.856$ parameter. Kemudian,

dilakukan *dropout* untuk 256 unit tadi yang nantinya akan diproses kembali untuk *layer dense* sebanyak 3 klasifikasi, sehingga parameter yang akan dihasilkan menjadi $(256 \times 3) + 3 \text{ bias} = 771$ parameter. Total parameter keseluruhan yang dikerjakan oleh *deep learning* adalah 476.195 perhitungan.

3.6. Fit Model

Fit model adalah proses penentuan model dari data citra latih dan data uji untuk mengetahui hasil dari nilai akurasi data citra latih dan data citra uji, selain nilai akurasi, akan terdapat nilai *loss*. Model ini terbentuk dari arsitektur algoritma dalam *cnn* yang telah disiapkan sebelumnya.



Gambar 8. Fit Model

Model arsitektur yang akan membentuk proses dan *output* yang akan didapatkan dalam penelitian ini. Setelah terbentuk model, dilakukan *fit model* untuk menentukan nilai akurasi dari data *training* dan data *Testing*. Peneliti menggunakan *epoch* atau iterasi sebanyak 500 iterasi, yang hasilnya bahwa data mendapatkan hasil akurasi sebesar 81% untuk data *training*. Dari data *training* yang telah diklasifikasikan dan diketahui nilai akurasinya, sehingga terbentuk hasil dalam *confusion matrix* untuk data *training*:

Tabel 2. Hasil Klasifikasi pada Data *Training*

	Jumlah	Pakcoy	Putih	Pagoda
Pakcoy	10	8	0	1
Putih	10	0	9	1
Pagoda	10	2	1	8

Hasil klasifikasi yang didapatkan pada tabel 2, terbentuk menjadi sebuah *confusion matrix* yang artinya bahwa data yang hasil klasifikasi tidak seluruhnya mampu terbaca sama oleh program sesuai dengan kategori. Pada objek citra sawi pakcoy hanya 45 citra yang mampu terdeteksi kategorinya dari 50 objek citra yang diujikan, objek citra sawi putih sebanyak 34 objek citra dari 43 objek citra yang diujikan, dan objek citra sawi caisim sebanyak 34 objek citra dari 47 objek citra yang diujikan pada data *training*.

Hasil nilai prediksi pada data *training* yang benar menghasilkan nilai yang cukup besar dibandingkan dengan prediksi data *training* yang salah, maka dapat disimpulkan perhitungan dengan arsitektur jaringan dapat dikatakan baik dengan mendapatkan nilai *precision* sebesar 85%, nilai *recall* sebesar 90% dan nilai akurasi sebesar 81%.

Sedangkan, pada data *testing* program mampu mengklasifikasikan objek citra sebesar 83% dari keseluruhan data *testing* yang diklasifikasikan.

```
> #Evaluation&Prediction-test data (2)
> model %>% evaluate(test, testLabels)
30/30 [-----] - 0s 3ms/step
$loss
[1] 0.4465474

$acc
[1] 0.8333333

> pred<-model %>% predict_classes(test)
> table(Predicted = pred, Actual = testy)
      Actual
Predicted 0 1 2
         0 8 0 1
         1 0 9 1
         2 2 1 8
```

Gambar 9. Hasil Evaluasi Training dan Prediksi Test Data

Sehingga, didapatkan hasil klasifikasi yang terbentuk dalam *confusion matrix* :

Tabel 3. Hasil Klasifikasi pada Data Testing

	Jumlah	Pakcoy	Putih	Pagoda
Pakcoy	50	45	2	6
Putih	43	0	34	7
Caisim	47	5	7	34

Hasil klasifikasi yang didapatkan pada tabel 3. memberikan penjelasan bahwa pada objek citra sawi pakcoy hanya 8 citra yang mampu terdeteksi kategorinya dari 10 objek citra yang diujikan, objek citra sawi putih hanya 9 objek citra dari 10 objek citra yang diujikan, dan objek citra sawi caisim hanya 8 objek citra dari 10 objek citra yang diujikan.

Hasil dari nilai prediksi pada data *testing* yang benar menghasilkan nilai yang cukup besar dibandingkan dengan prediksi data *testing* yang salah, maka dapat disimpulkan perhitungan dengan arsitektur jaringan dapat dikatakan baik dengan mendapatkan nilai *precision* sebesar 89%, nilai *recall* sebesar 80% dan nilai akurasi sebesar 84%.

Tingkat akurasi 84% artinya kinerja sistem terhadap metode yang digunakan mampu mengerjakan dengan tingkat persentase sebesar 84% dari 100%. Sistem mampu mengklasifikasikan dengan tepat data citra dengan baik sebesar 84%, kemudian sisanya artinya 16% sistem tidak mampu mengklasifikasikan dengan tepat.

Dengan hasil tersebut maka dapat dikatakan bahwa penggunaan metode *convolutional neural network* menggunakan *library Keras* relevan diimplementasi terhadap citra sawi yaitu sawi pakcoy, sawi putih dan sawi caisim.

KESIMPULAN

Berdasarkan hasil penelitian yang sudah dilakukan, maka dapat ditarik kesimpulan hasilnya adalah sebagai berikut:

1. Berdasarkan model yang telah dibuat, hasil validasi dari Prediksi data *testing* memiliki nilai *loss* 0,4465474 dan nilai akurasi sebesar 0,83. Nilai tersebut dihasilkan dari epoch sebanyak 250, batch size 32 dan validation split 0.2. Pada tabel prediksi dapat dilihat pada prediksi klasifikasi citra pakcoy mengalami 1 kesalahan dan hanya 8 citra yang diprediksi sesuai dengan yang diinginkan, pada citra sawi putih mengalami 1 kesalahan dan hanya 9 citra yang sesuai dengan yang

diinginkan, pada klasifikasi sawi caisim mengalami 3 kesalahan dari 10 citra yang diharapkan. Untuk akurasi model uji berasal dari hasil data testing.

2. Tingkat akurasi model yang didapatkan dari hasil pendeteksian klasifikasi citra sawi pakchoy, sawi putih dan sawi caisim pada suatu citra digital menggunakan *Convolutional Neural Network* yaitu sebesar 84% untuk tingkat akurasi, presisi 89% , dan recall 80%.

SARAN

Untuk menyempurnakan penelitian selanjutnya maka ada beberapa saran yang bisa dikembangkan yaitu:

1. Tahapan *preprocessing* yang dilakukan juga dengan menambahkan proses segmentasi serta penghilangan *noise* pada citra dengan bantuan komputer untuk mendapatkan hasil kualitas citra yang lebih baik.
2. Data citra yang digunakan diperbanyak untuk melatih model dan mencapai tingkat akurasi yang tinggi.
3. Spesifikasi komputer yang digunakan sebaiknya lebih tinggi, yaitu dengan menggunakan komputer dengan *Graphics Processing Unit (GPU)* dan *Random Access Memory (RAM)* yang tinggi .
4. Perbandingan antara nilai jumlah filter, *learning rate*, *epoch*, *layer dropout*, serta fungsi aktivasi yang digunakan bisa menjadi penelitian lebih lanjut untuk melihat bagaimana pengaruhnya terhadap tingkat akurasi yang diperoleh.

DAFTAR PUSTAKA

- Arun Priya C, Balasaravanan T. 2012. *An Efficient Leaf Recognition Algorithm for Plant Classification Using Support Vector Machine*, *Proceedings of the International Conference on Pattern Recognition, Informatics and Medical Engineering*. Hlm. 21-23
- Chaki J, Parekh R. 2011. *Plant Leaf Recognition using Shape based Features and Neural Network classifiers*, *International journal of Advanced Computer Science and Applications*. Vol 2, no 10
- Danukusumo, K. P. 2017. *Implementasi Deep Learning Menggunakan Convolutional Neural Network untuk Klasifikasi Citra Candi Berbasis GPU*. Fakultas Teknologi Industri Universitas Atma Jaya Yogyakarta
- Kadir, Lukito E. N, Adhi N. 2011. *Leaf Classification Using Shape, Color, and Texture Features*, *International Journal of Computer Trends and Technology*
- Stephen G. W., Forrest S. B., Eric Y. Xu, Yu-Xuan W., Yi-F. C. and Qiao-Liang X. 2007. *A Leaf Recognition Algorithm for Plant Classification Using Probabilistic Neural Network*. *IEEE International Symposium*. pp 11-16
- Suartika, I Wayan E. P , Arya Yudhi W. 2016. *Klasifikasi Citra Menggunakan Convolutional Neural Network (Cnn) pada Caltech 101*. *Jurnal Teknik ITS* Vol. 5, No. 1
- Theodoridis, Sergios. and Koutroumbas, Kontantinos. 2003. *Pattern Recognition*. 2nd Edition. New York. USA: Academic Press
- Z. Husin, A. Y. M. Shakaff, A. H. A. Aziz, R. S. M. Farook, M. N. Jaafar, U. Hashim, A. Harun. 2012. *Embedded Portable Device For Herb Leaves Recognition Using Image Processing Techniques And Neural Network Algorithm*, *Science Direct on Computers and Electronics in Agriculture*. Hlm. 18-29
- Zufar, Muhammad, Budi Setiyono. 2016. *Convolutional Neural Networks untuk Pengenalan Wajah Secara Real-Time*. *Jurnal Sains dan Seni ITS* Vol. 5 No. 2