



SUBMIT

(Jurnal Ilmiah Teknologi Informasi dan Sains)

Vol. I No. I (2021) 8 - 15

e-ISSN:-

p-ISSN:-

PENGGUNAAN *DEEP LEARNING* DENGAN METODE *CONVOLUTIONAL NEURAL NETWORK* UNTUK KLASIFIKASI KUALITAS SAYUR KOL BERDASARKAN CITRA FISIK

Ratna Dhamayanti¹, Mimin Fatchiyatur Rohmah², Soffa Zahara³

¹²³Universitas Islam Majapahit

e-mail: ¹ratna.dhama98@gmail.com, ²miminfr@gmail.com, ³soffa.zahara@gmail.com

Naskah masuk: 5 Februari 2021, diterima untuk diterbitkan: 15 Juni 2021

Abstrak

Sayur kol merupakan salah satu jenis sayuran yang umum dijumpai masyarakat Indonesia. Banyaknya sayur kol yang dipanen, membuat petani terkadang sedikit kerepotan dalam proses menyortir sayur kol mana yang layak ekspor dan tidak. Dan dengan seiring berkembangnya teknologi, maka hal ini menjadi sangat mungkin untuk membuat komputer mampu melakukan pekerjaan yang dianggap biasa oleh manusia termasuk proses sortir kualitas sayur kol. Hal semacam ini dapat dilakukan dengan menggunakan *Deep Learning* yang mengunakan *Convolutional Neural Network* (CNN) sebagai metode klasifikasi. Dengan memanfaatkan citra fisik sayur kol, CNN mampu melakukan klasifikasi dengan model yang telah disusun sebelumnya. Dalam penelitian ini, peneliti membangun sebuah model yang terdiri dari 4 *convolution layer*, 2 *pooling layer* yang berukuran 2×2, 3 *dropout layer*, 2 *dense layer* serta 1 *flatten layer*. Sedang untuk aktivasinya, digunakan *ReLU*, dengan *filter* sebanyak 32 dan 64 yang ukuran kernelnya 3×3. Model ini diuji dengan menggunakan 270 data yang dimana 210 digunakan sebagai data *train* dan 60 data digunakan sebagai data *test*. Dengan *learning rate* sebesar 0.004, 30 *epoch* dan tiga algoritma performansi berbeda yaitu; *Stochastic Gradient Descent* (SGD), *Adaptive Moment* (Adam), dan *Root Mean Square Propagation* (RMSProp) dengan hasil tertinggi berada pada algoritma Adam yang tingkat akurasi sebesar 80% untuk data *test* dan 73% untuk data *train* berdasarkan komposisi warna yang terdapat dalam citra.

Kata kunci: citra digital, deep learning, convolutional neural network (CNN), sayur kol.

USING DEEP LEARNING THROUGH CONVOLUTIONAL NEURAL NETWORK METHOD FOR QUALITY CLASSIFICATION OF CABBAGES BASED ON PHYSICAL IMAGES

Abstract

Cabbages are one of the common vegetables that Indonesian people can find. Because there are many cabbages that must be harvested, the farmers get a little bit inconvenience in the process of sorting out the cabbages that are worthily exported or not. As the era becomes technologically advanced, it is possible for the computer to be able to do works that people consider it as customary, including the process of sorting out the cabbages qualities. This kind of thing can be conducted using Deep Learning that carries Convolutional Neural Network (CNN) as the classification method. Seizing the opportunity of the cabbages' physical image, CNN can classify through the model that has been formerly designed. In this research, the research built a model that consists of 4 convolutional layers, 2 pooling layers of 2x2, 3 dropout layers, 2 dense layers, as well as 1 flatten layer. While for the activity, ReLu through 32 and 64 filters with its kernel of 3x3 were used. This model was examined using 270 data, which 210 data used as the data train, and 60 data were applied as the data test. By learning rate of 0.004, 30 epochs, and 3 algorithms with various performances which are; Stochastic Gradient Descent (SGD), Adaptive Moment (Adam), and Root Mean Square Propagation (RMSProp), with the highest result on the Adam algorithm with the accuracy level of 80% for the data test and 73% for the data train based on the color composition on the image.

Keywords: *digital image, deep learning, convolutional neural network (CNN), cabbages.*

1. PENDAHULUAN

Sebagai negara dengan predikat kepulauan serta wilayah yang tergolong luas, menjadikan Indonesia memiliki keragaman jenis tanah juga mata pencaharian yang kemudian memunculkan berbagai macam profesi sesuai dengan kenampakan alam tempat huni setiap masyarakat Indonesia. Salah satunya dengan melakukan perkembangan pertanian di bidang pangan khususnya hortikultura. Adapun salah satu hasil komoditas hortikultura di Indonesia adalah sayur kol atau yang akrab disapa kubis.

Namun, terkadang sayur kol yang ditemui memiliki beberapa kecacatan pada bentuk fisiknya, sehingga perlu dilakukan proses seleksi ulang dan ini akan memakan waktu lama apabila dilakukan hanya dengan menggunakan tenaga manusia saja. Dan dengan berkembangnya teknologi, maka hal itu bukan menjadi hal yang sulit lagi.

Dan salah satu metode yang diketahui banyak digunakan untuk mengatasi permasalahan ini yaitu metode *Convolutional Neural Network* (CNN). CNN ini sendiri adalah salah satu algoritma dari *Deep Learning* yang merupakan pengembangan dari *Multi Layer Perceptron* (MLP). Pada dasarnya, *Deep Learning* merupakan cabang ilmu dari *Machine Learning* yang berbasis jaringan saraf tiruan dengan konsep pengajaran kepada komputer untuk dapat melakukan Tindakan yang biasa dianggap wajar oleh manusia.

Convolutional Neural Network (CNN) merupakan salah satu metode dalam *Deep Learning* yang sedang berkembang saat ini dengan asumsi bahwa *input* yang digunakan yaitu data yang berupa

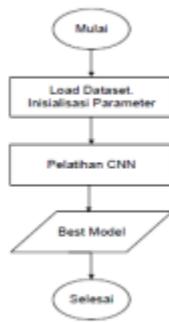
gambar. Dalam CNN, terdapat lapisan konvolusi yang dimana, pada lapisan ini akan dapat menghasilkan sebuah pola dari dari beberapa bagian citra yang nantinya akan lebih mudah untuk diklasifikasikan. Teknik ini dapat membuat fungsi pembelajaran citra menjadi lebih efisien untuk diimplementasikan. Oleh karenanya, dengan kelebihan yang dimiliki CNN dalam pengolahan data gambar, peneliti akan menjadikan CNN sebagai solusi untuk pengenalan sayur kol dengan harapan bahwa dengan adanya penelitian ini, dapat meringankan pekerjaan manusia dalam pengenalan kualitas sayur kol berdasarkan citra fisik.

Tujuan adanya penelitian ini yaitu untuk melakukan klasifikasi terhadap kualitas sayur kol berdasarkan citra fisik dengan menggunakan metode CNN. Serta mengetahui model yang paling tepat untuk klasifikasi sayur kol juga mengetahui hasil dari klasifikasi sayur kol yang menggunakan CNN sebagai metode penelitian.

Penelitian yang dilakukan oleh peneliti bukanlah penelitian yang pertama. Namun, penelitian lain juga telah dilakukan dan sudah ada terkait dengan *Deep Learning* yang mengungus metode CNN.

2. METODOLOGI

Secara umum, metodologi sistem Klasifikasi Kualitas Sayur Kol Berdasarkan Citra Fisik Menggunakan *Deep Learning* Dengan Metode *Convolutional Neural Network* (CNN) dapat dilihat pada diagram alir berikut:



Gambar 1. Perancangan Sistem

1) *Load Dataset*, Inisialisasi Parameter

Hal pertama yang dilakukan dalam perancangan sistem ini yaitu memasukan *dataset* (data *train*) yang sebelumnya telah dibuat. Kemudian, gambar-gambar ini akan dimasukkan ke dalam parameter yang telah dibuat.

2) Pelatihan CNN

Setelah model terbentuk, berikutnya dilakukan pelatihan CNN dengan menggunakan data *training* yang dimana dalam penelitian ini terdapat 210 data untuk *training*.

3) *Best Model*

Model terbaik bisa didapat dengan penambahan parameter, perubahan iterasi (*epoch*) juga penggunaan algoritma performasi. Dalam penelitian ini, peneliti menggunakan algoritma performasi yang berbeda untuk mendapat hasil model yang terbaik dengan nilai akurasi yang tergolong tinggi.

3. ANALISIS SISTEM

3.1. Populasi dan Sampel

Dalam penelitian ini, data gambar sayur kol dijadikan sebagai populasi. Yang mana, umumnya sayur kol dapat dengan mudah dijumpai oleh masyarakat Indonesia pada umumnya. Sedang sampel yang digunakan yaitu sayur kol yang banyak dibudidayakan di Indonesia. Gambar yang digunakan merupakan data yang diambil dengan menggunakan kamera ponsel (Samsung J5 2016) dengan resolusi 13MP. Pengambilan gambar dilakukan dengan memanfaatkan cahaya matahari dan *flash* yang terdapat pada ponsel.

Diambil dengan tiga waktu yang berbeda (pagi, siang, dan malam) membuat gambar yang dihasilkan memiliki presisi cahaya yang berbeda. Sayur kol yang telah dipotret dalam penelitian ini kemudian disimpan dalam format *.jpg dengan skala 1:1 dan setelahnya data dipecah menjadi data *train* (yang terdiri dari 210 gambar), serta data *test* yang terdiri dari 60 gambar.

3.2. Perancangan *Software*

Bagian ini menjelaskan tentang perangkat lunak apa saja yang digunakan atau yang dapat mendukung proses penelitian agar mampu berjalan dengan baik.

Tabel 1. Spesifikasi *Software* Yang Digunakan

Komponen	Spesifikasi
Sistem operasi	Windows 10 64-bit
Browser	Microsoft Edge versi 44.18362.449.0
Perancangan sistem	Python 3.6 64-bit Keras versi 2.1.5 Tensorflow versi 1.14.0 Matplotlib versi 3.2.1 Numpy versi 1.18.4

3.3. Perancangan *Hardware*

Bagian ini menjelaskan tentang perangkat keras apa saja yang digunakan atau yang dapat mendukung proses penelitian agar mampu berjalan dengan baik

Tabel 2. Spesifikasi *Hardware* Yang Digunakan

Komponen	Spesifikasi
Laptop	Acer Aspire 3 A315-21-948E
Prosesor	AMD A9
RAM (<i>Random Access Memory</i>)	4GB DDR4
Grafis	AMD Radeon R5 Graphics
Harddisk	1000GB

3.4. Arsitektur CNN

Pada model arsitektur yang dibuat peneliti, gambar akan masuk ke dalam 16 filter konvolusi dengan dengan ukuran 3x3. Selanjutnya, akan dilakukan *pooling* sebesar 2x2 *pixel* dan *dropout* sebesar 0,25. Pada bagian selanjutnya, dilakukan proses konvolusi yang lebih besar yaitu 64 filter konvolusi dengan *pooling* sebesar 2x2 *pixel* juga *dropout* sebesar 0,25. Setelah semua proses selesai, akan masuk ke dalam *hidden layer* dengan kedalaman *fully connected layer* sebesar 512 dan 3 dengan *dropout* sebesar 0,5 hingga menghasilkan *output*. Hasil dari model klasifikasi dapat dilihat pada tabel berikut.

Tabel 3. Arsitektur CNN

No.	Layer	Output Shape
1.	<i>Convolutional 2D</i>	32x3x3
2.	<i>Activation (ReLU)</i>	32x32x32
3.	<i>Pooling Layer</i>	2x2
4.	<i>Dropout</i>	15x15x32
5.	<i>Flatten</i>	2304
6.	<i>Dense</i>	512

4. PEMBAHASAN DAN IMPLEMENTASI

4.1. Proses *Training*

Proses ini dilakukan dengan *me-resize* gambar ke dalam ukuran 32x32 *pixel* dan diberi label yang baru. Pada bagian ini, intensitas *pixel* gambar juga diubah ke dalam range [0, 1]. Selanjutnya, citra sayur kol diubah warna ke dalam *grayscale*. Bila proses ini telah selesai, komputer kemudian akan melakukan penghilangan *noise* dan penentuan ukuran target menjadi 32x32. Saat *pixel* yang diambil lebih kecil dari yang ditentukan, maka akan berdampak pada kehilangan data *pixel* yang terlalu banyak. Dan jika terlalu besar, membuat proses komputasi memakan waktu yang lama. Untuk mengatasi hal ini, penulis merubah ukuran gambar menjadi 500x500 *pixel* terlebih dahulu agar tidak memakan waktu lama.

```

train_gen = aug.flow_from_directory('data/train',
    target_size = (32,32),
    color_mode = 'grayscale',
    batch_size = 32,
    class_mode = 'categorical',
    shuffle = True,
    subset = 'training')

val_gen = aug.flow_from_directory('data/train',
    target_size = (32,32),
    color_mode = 'grayscale',
    batch_size = 32,
    class_mode = 'categorical',
    shuffle = False,
    subset = 'validation')
    
```

Gambar 2. Script Merubah Citra RGB ke *Grayscale*

Setelah *dataset* didapat, selanjutnya dilakukan perancangan model klasifikasi. Prinsip dasar dari proses ini sendiri adalah dengan melakukan pelatihan pada CNN untuk menghasilkan model terbaik yang mampu memberikan akurasi tinggi. Secara keseluruhan, arsitektur yang direncanakan dalam penelitian ini dapat ditunjukkan pada Gambar 3 berikut. Pada bagian ini, data *train* yang ada *split* menjadi 80:20

layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 32, 32, 32)	328
activation_1 (Activation)	(None, 32, 32, 32)	0
batch_normalization_1 (Batch Normalization)	(None, 32, 32, 32)	128
conv2d_2 (Conv2D)	(None, 16, 16, 32)	8320
activation_2 (Activation)	(None, 16, 16, 32)	0
max_pooling2d_1 (MaxPooling2D)	(None, 15, 15, 32)	0
dropout_1 (Dropout)	(None, 15, 15, 32)	0
conv2d_3 (Conv2D)	(None, 15, 15, 64)	10496
activation_3 (Activation)	(None, 15, 15, 64)	0
batch_normalization_2 (Batch Normalization)	(None, 15, 15, 64)	256
conv2d_4 (Conv2D)	(None, 13, 13, 64)	5888
activation_4 (Activation)	(None, 13, 13, 64)	0
max_pooling2d_2 (MaxPooling2D)	(None, 6, 6, 64)	0
dropout_2 (Dropout)	(None, 6, 6, 64)	0
Flatten_1 (Flatten)	(None, 2304)	0
dense_1 (Dense)	(None, 512)	118656
activation_5 (Activation)	(None, 512)	0
batch_normalization_3 (Batch Normalization)	(None, 512)	2048
dropout_3 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 1)	2309
activation_6 (Activation)	(None, 1)	0
Total params: 1,249,129		
Trainable params: 1,247,087		
Non-trainable params: 1,120		

Gambar 3. Model Klasifikasi

Selanjutnya yaitu melakukan pelatihan data gambar sayur kol ke dalam model dengan *fit model*. Dalam melakukan *fit model*, digunakan *epoch* = 30, *batch_size* = 32 dan *validation_split* = 0,2. *Epoch*

berarti berapa kali jaringan akan melihat seluruh kumpulan data, sedang *batch_size* adalah jumlah contoh pelatihan dalam satu *forward/backward pass*. Semakin tinggi nilai *batch_size* maka akan semakin banyak memori yang dibutuhkan.

Namun, pada penelitian ini, peneliti menggunakan tiga jenis algoritma performansi. Sehingga, akan ada delapan hasil *fit model*, dan tiga akurasi untuk hasil iterasi yang telah dilakukan.

1) *Fit Model* Dengan Algoritma *Stochastic Gradient Descent* (SGD)

Bagian ini akan menampilkan hasil *fit model* yang didapat dengan *Stochastic Gradient Descent* (SGD) sebagai algoritma performansi.

Tabel 4. Hasil *Fit Model* Algoritma SGD

Epoch	Data Train		Data Test	
	Acc	Loss	Val Acc	Val Loss
1	0.5041	14.442	0.6098	0.8457
2	0.6047	10.630	0.5366	0.8293
3	0.6225	10.433	0.7561	0.6074
4	0.6052	0.9968	0.7317	0.5727
5	0.6131	10.050	0.7317	0.5213
⋮	⋮	⋮	⋮	⋮
26	0.6707	0.7562	0.6341	0.7991
27	0.6682	0.7416	0.6098	0.8590
28	0.6640	0.7481	0.7073	0.5869
29	0.6955	0.6486	0.6098	0.7989
30	0.7157	0.6678	0.6829	0.7173

Gambar 4 berikut merupakan grafik yang menunjukkan pergerakan nilai *loss* untuk data *train* dan data *test* yang dihasilkan dalam setiap iterasi (*epoch*). Garis warna merah menunjukkan pergerakan *loss* untuk data *train*, dan garis biru menunjukkan pergerakan data *test*.



Gambar 4. Grafik Nilai *Loss* Data *Train* dan Data *Test* Algoritma SGD

Dari gambar di atas, dapat dilihat bahwa pergerakan *loss* untuk data *test* kurang stabil. Sedangkan untuk data *train*, grafik menunjukkan penurunan hingga *epoch* yang terakhir.



Gambar 5. Grafik Nilai Akurasi Data Train dan Data Test Algoritma SGD

Berdasarkan gambar 5, dapat dilihat bahwa hasil iterasi yang dihasilkan untuk data *train* maupun data *test* tergolong rendah. Meski grafik untuk data *test* cenderung naik bahkan angka tertinggi hingga mencapai 85%, namun hasil di akhir mampu turun secara ekstrim menyentuh angka 68%. Dan untuk data *train*, hanya mampu menyentuh angka 70%.

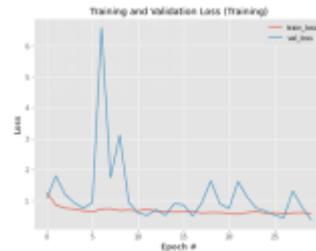
2) *Fit Model* Dengan Algoritma *Adaptive Moment* (Adam)

Bagian ini akan menampilkan hasil *fit model* yang didapat dengan *Adaptive Moment* (Adam) sebagai algoritma performasi.

Tabel 5. Hasil *Fit Model* Algoritma Adam

Epoch	Data Train		Data Test	
	Acc	Loss	Val Acc	Val Loss
1	0.5704	12.578	0.5854	10.656
2	0.6410	0.8446	0.5122	17.921
3	0.6749	0.8114	0.5610	11.984
4	0.6605	0.7483	0.5122	0.9292
5	0.6815	0.6982	0.6829	0.7490
⋮	⋮	⋮	⋮	⋮
26	0.7196	0.6655	0.6585	0.5257
27	0.7412	0.6111	0.8049	0.4377
28	0.7042	0.6474	0.5854	13.083
29	0.7277	0.6470	0.6829	0.7909
30	0.7335	0.5721	0.8049	0.3762

Berikut uraian grafik pergerakan nilai nilai *loss* untuk data *train* dan data *test* yang dihasilkan setiap iterasi (*epoch*) dapat dilihat pada Gambar 6 berikut. Garis warna merah menunjukkan pergerakan *loss* untuk data *train*, dan garis biru menunjukkan pergerakan data *test*. Dan dari gambar grafik di bawah, dapat dilihat bahwa pergerakan *loss* untuk data *test* kurang stabil, namun di akhir iterasi mengalami penurunan yang berarti menunjukkan bahwa tingkat *loss* untuk model ini menurun. Sedang untuk data *train*, grafik menunjukkan penurunan secara konstan hingga *epoch* yang terakhir.



Gambar 6. Grafik Nilai Loss Data Train dan Data Test Algoritma Adam

Pada gambar 6 ditunjukkan grafik pergerakan nilai akurasi dari kedua data. Berdasarkan gambar 14, dapat dilihat bahwa hasil iterasi yang dihasilkan untuk data *train* maupun data *test* tergolong tinggi, meski grafik untuk data *test* cenderung naik turun secara ekstrim, namun hasil tertinggi mampu menyentuh angka di atas 80%. Dan untuk data *train*, cenderung naik stabil dengan akurasi di *epoch* ke-30 menyentuh angka 73%.



Gambar 7. Grafik Nilai Akurasi Data Train dan Data Test Algoritma Adam

3) *Fit Model* Dengan Algoritma *Root Mean Square Propagation* (RMSProp)

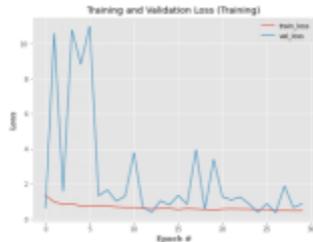
Bagian ini akan menampilkan hasil *fit model* yang didapat dengan *Root Mean Square Propagation* (RMSProp) sebagai algoritma performasi.

Tabel 6. Hasil *Fit Model* Algoritma RMSProp

Epoch	Data Train		Data Test	
	Acc	Loss	Val Acc	Val Loss
1	0.5909	13.137	0.7805	0.6317
2	0.6151	0.9554	0.3415	10.587
3	0.6609	0.8263	0.3171	15.765
4	0.6237	0.9004	0.3171	10.785
5	0.6804	0.7334	0.3171	88.237
⋮	⋮	⋮	⋮	⋮
26	0.7704	0.5799	0.6341	0.8686
27	0.8208	0.4628	0.9024	0.3615
28	0.8039	0.4859	0.5854	18.788
29	0.8306	0.4568	0.8293	0.6490
30	0.8023	0.4742	0.6585	0.8577

Berikut uraian grafik pergerakan nilai nilai *loss* untuk data *train* dan data *test* yang dihasilkan setiap iterasi (*epoch*) dapat dilihat pada Gambar 16 berikut. Garis warna merah menunjukkan pergerakan *loss* untuk data *train*, dan garis biru menunjukkan

pergerakan data *test*. Dan dari gambar grafik di bawah, dapat dilihat bahwa pergerakan *loss* untuk data *train* maupun data *test* memiliki sedikit kemiripan dengan hasil perhitungan *loss* oleh algoritma Adam.



Gambar 1. Grafik Nilai Loss Data Train dan Data Test Algoritma RMSProp

Pada gambar 9 ditunjukkan grafik pergerakan nilai akurasi dari kedua data. Berdasarkan gambar 8, dapat dilihat bahwa hasil iterasi yang dihasilkan untuk data *train* maupun data *test* tergolong tinggi, meski di akhir iterasi perhitungan untuk akurasinya menjorok ke bawah dengan masing-masing hasil untuk data *train* turun di angka 80% dan data *test* turun hingga hamper menyentuh angka 60%.



Gambar 9. Grafik Nilai Akurasi Data Train dan Data Test Algoritma RMSProp

4) Uji Coba-1

Uji coba-1 ini merupakan uji coba yang dilakukan oleh peneliti terhadap model yang telah dibangun. Menggunakan model kedua yang menggunakan Adam sebagai algoritma performansi yang memiliki nilai akurasi tertinggi sebesar 80%, uji coba ini dilakukan pada citra sayur kol yang memiliki komposisi cahaya lebih sedikit, dengan hasil citra yang memiliki bayangan lebih tebal (gelap).



Gambar10. Data Test Percobaan Pertama

Berikut merupakan hasil yang didapat setelah klasifikasi menggunakan model CNN yang telah dibuat sebelumnya.

Tabel 7. Uji Coba-1

No Urut Data	Kualitas Citra Sesungguhnya	Hasil Klasifikasi	Status
1	Buruk	Buruk	True
2	Buruk	Buruk	True
3	Buruk	Buruk	True
4	Cukup	Cukup	True
5	Cukup	Cukup	True
6	Cukup	Cukup	True
7	Cukup	Cukup	True
8	Cukup	Cukup	True
9	Cukup	Cukup	True
10	Cukup	Cukup	True
11	Cukup	Cukup	True
12	Cukup	Cukup	True
13	Cukup	Layak	False
14	Cukup	Cukup	True
15	Cukup	Cukup	True
16	Cukup	Cukup	True
17	Layak	Cukup	False
18	Layak	Layak	True
19	Layak	Layak	True
20	Layak	Layak	True
Total Klasifikasi Benar			18
Persentase			90%

Dari tabel di atas, dapat disimpulkan bahwa model mampu mengklasifikasi data citra yang disediakan sebanyak 90% data dengan benar.

6) Uji Coba-2

Uji coba-2 ini merupakan uji coba yang dilakukan oleh peneliti terhadap model yang telah dibangun. Menggunakan model kedua yang menggunakan Adam sebagai algoritma performansi yang memiliki nilai akurasi tertinggi sebesar 80%, uji coba ini dilakukan pada citra sayur kol yang memiliki komposisi cahaya lebih banyak, sehingga citra yang dihasilkan memiliki bayangan lebih tipis (terang).



Gambar 11. Data Test Percobaan Kedua

Berikut merupakan hasil yang didapat setelah klasifikasi menggunakan model CNN yang telah dibuat sebelumnya.

Tabel 8. Uji Coba-2

No Urut Data	Kualitas Citra Sesungguhnya	Hasil Klasifikasi	Status
1	Layak	Layak	True
2	Layak	Layak	True
3	Layak	Layak	True
4	Buruk	Buruk	True

5	Buruk	Buruk	True
6	Cukup	Cukup	True
7	Cukup	Cukup	True
8	Cukup	Cukup	True
9	Layak	Layak	True
10	Layak	Layak	True
11	Layak	Cukup	False
12	Buruk	Cukup	False
13	Buruk	Buruk	True
14	Buruk	Buruk	True
15	Layak	Layak	True
16	Layak	Layak	True
17	Layak	Layak	True
18	Layak	Layak	True
19	Layak	Layak	True
20	Layak	Layak	True
Total Klasifikasi Benar			18
Persentase			90%

Dari tabel di atas, dapat disimpulkan bahwa model mampu mengklasifikasi data citra yang disediakan sebanyak 90% data dengan benar.

7) Uji Coba-3

Uji coba-3 ini merupakan uji coba yang dilakukan oleh peneliti terhadap model yang telah dibangun. Menggunakan model kedua yang menggunakan Adam sebagai algoritma performansi yang memiliki nilai akurasi tertinggi sebesar 80%, uji coba ini dilakukan pada citra bukan sayur kol yang memiliki komposisi cahaya yang baik, sehingga citra yang dihasilkan memiliki kualitas yang baik dengan bayangan yang minim dan persebaran cahaya pada benda yang merata.



Gambar 12. Data Test Percobaan Ketiga

Berikut merupakan hasil yang didapat setelah klasifikasi menggunakan model CNN yang telah dibuat sebelumnya.

Tabel 9. Uji Coba-3

No Urut Data	Jenis Citra Sesungguhnya	Hasil Klasifikasi	Status
1	Paprika	Cukup	False
2	Anggur	Buruk	False
3	Apel	Buruk	False
4	Apel	Buruk	False
5	Belimbing	Buruk	False

6	Botol	Buruk	False
7	Brokoli	Buruk	False
8	Bunga	Cukup	False
9	Karakter Kartun	Cukup	False
10	Jeruk	Buruk	False
11	Kangkung	Buruk	False
12	Kaos	Buruk	False
13	Kemangi	Buruk	False
14	Kresek	Buruk	False
15	Laptop	Buruk	False
16	Melon	Buruk	False
17	Parfum	Buruk	False
18	Pansil	Buruk	False
19	Sayur	Buruk	False
20	Sepatu	Buruk	False
Total Klasifikasi Benar			0
Persentase			0%

Dari tabel di atas, dapat disimpulkan bahwa model belum mampu mengklasifikasi objek selain sayur kol. Hal ini terjadi karena pada penelitian ini, peneliti hanya menggunakan warna sebagai parameter untuk seleksi kualitas sayur kol. Sehingga, meski objek yang diuji bukan sayur kol, namun model akan tetap melakukan klasifikasi sebagaimana yang dilakukan pada sayur kol.

4. SIMPULAN DAN SARAN

5.1. Simpulan

Simpulan yang dapat ditarik dari hasil penelitian serta pembahasan implementasi *Convolutional Neural Network* (CNN) dalam melakukan klasifikasi terhadap kualitas sayur kol yaitu dari tiga algoritma performansi yang digunakan, untuk proses akurasi tertinggi ada pada algoritma Adam dengan hasil akurasi tertinggi berada di angka 80% untuk data *test* dan 73% untuk data *train*. Sehingga, dapat disimpulkan bahwa, metode *Convolutional Neural Network* (CNN) mampu melakukan klasifikasi kualitas sayur kol berdasarkan citra fisik dengan baik.

Namun, model ini hanya mampu mengklasifikasi citra sayur kol dan melakukan klasifikasi berdasarkan warna pada citra. Jumlah fitur yang digunakan hanya satu (klasifikasi berdasarkan warna pada citra).

5.2. Saran

Tentunya, penelitian ini jauh dari kata sempurna. Sehingga, beberapa saran perlu dimasukkan dengan segala aspek pertimbangan yaitu sistem ini dapat dikembangkan lagi dengan adanya penambahan ekstraksi fitur untuk klasifikasi. Penambahan *interface*. Diharapkan, untuk kedepannya akan banyak data yang digunakan untuk proses penelitian. Karena semakin banyak data, maka tidak menutup kemungkinan hal ini mempengaruhi tingkat akurasi

yang dihasilkan model itu sendiri. Agar sistem mampu mendeteksi citra sayur kol lebih baik, penambahan *preprocessing* diperlukan agar tidak terjadi kesalahan dalam klasifikasi.

Penelitian ini dapat dikembangkan lagi dengan menambahkan perhitungan akurasi salah satunya dengan menggunakan *Root Mean Square Error* (RMSE).

DAFTAR PUSTAKA

- A'LA, F. Y (2016). Deteksi Retak Permukaan Jalan Raya Berbasis Pengolahan Citra Menggunakan Metode Ekstraksi Ciri Wavelet. Yogyakarta: Program Studi Teknik Informatika Fakultas Teknik Universitas Muhammadiyah Yogyakarta.
- BERND, J., & H. HORST. (2000). *Computer Vision and Application*. San Diego: Academic Press.
- <http://deeplearning.net/tutorial/lenet.html>. Diakses 5 Juni 2020.
- DZULQARNAIN, M. F., SUPRAPTO, S., & MAKHRUS, F. (2019). Improvement of Convolutional Neural Network Accuracy on Salak Classification Based Quality on Digital Image. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, 13(2), 189. <https://doi.org/10.22146/ijccs.42036>.
- FUKUSHIMA, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4), 193–202. <https://doi.org/10.1007/BF00344251>
- ILAHYAH, S., & NILOGIRI, A. (2018). Implementasi Deep Learning Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan Convolutional Neural Network. *JUSTINDO (Jurnal Sistem Dan Teknologi Informasi Indonesia)*, 3(2), 49–56. <https://doi.org/10.32528/JUSTINDO.V3I2.2254>
- MELYDAILY.(2019). Kubis hias atau Ornamen cabbage menghiasi musim dingin di Austria. Diakses 5 Juni 2020, dari <http://www.melydaily.com>.
- PURNAMASARI, R. W. (2013). *Backpropagation Sebagai Sistem Deteksi Penyakit Tuberculosis (Tbc)*. 134.
- PUTRA, D. (2010). *Pengolahan citra digital*. Penerbit Andi.
- PUTRA, I. W. S. E., WIJAYA, A. Y., & SOELAIMAN, R. (2016). Klasifikasi Citra Menggunakan Convolutional Neural Network (Cnn) Pada Caltech 101 Image Classification Using Convolution Neural Network (Cnn) on Caltech 101. *Institut Teknologi Sepuluh November*, 5(1), 1–76.
- RAINA, G. (2016). *Deep Convolutional Network evaluation on the Intel Xeon Phi : Where Subword Parallelism meets*.
- RAWLA, P., SUNKARA, T., GADUPUTI, V., JUE, T. L., SHARAF, R. N., APPALANENI, V., ANDERSON, M. A., BEN-MENACHEM, T., DECKER, G. A., FANELLI, R. D., FUKAMI, N., IKENBERRY, S. O., JAIN, R., KHAN, K. M., KRINSKY, M. L., MALPAS, P. M., MAPLE, J. T., FISHER, D., HWANG, J. H., ... ABBAS KS, MADBOULY KM, ABBAS MAS, E. D. A. (2018). Title. *Gastrointestinal Endoscopy*, 10(1), 279–288. <https://doi.org/10.1542/peds.2006-2099>
- RISMIYATI. 2016. Implementasi convolutional neural network untuk sortasi mutu salak ekspor berbasis citra digital (Tesis). Universitas Gadjah Mada: Yogyakarta.
- wildml.(2015). Understanding Convolutional Neural Networks For NLP. Diakses 5 Juni 2020, dari <http://www.wildml.com>.