

Klasifikasi Citra Menggunakan Convolutional Neural Network dan K Fold Cross Validation

Ari Peryanto ^{1*}, Anton Yudhana ^{2**}, Rusydi Umar ^{3*}

* Program Studi Magister Teknik Informatika, Universitas Ahmad Dahlan

** Program Studi Teknik Elektro, Universitas Ahmad Dahlan

ari1907048002@webmail.uad.ac.id¹, eyudhana@ee.uad.ac.id², rusydi.umar@tif.uad.ac.id³

Article Info

Article history:

Received 2020-04-23

Revised 2020-05-12

Accepted 2020-05-13

Keyword:

Convolution Neural Network,
Image,
Clasifikasi,
Cross Validation.

ABSTRACT

Image classification is a fairly easy task for humans, but for machines it is something that is very complex and is a major problem in the field of Computer Vision which has long been sought for a solution. There are many algorithms used for image classification, one of which is Convolutional Neural Network, which is the development of Multi Layer Perceptron (MLP) and is one of the algorithms of Deep Learning. This method has the most significant results in image recognition, because this method tries to imitate the image recognition system in the human visual cortex, so it has the ability to process image information. In this research the implementation of this method is done by using the Keras library with the Python programming language. The results showed the percentage of accuracy with K = 5 cross-validation obtained the highest level of accuracy of 80.36% and the highest average accuracy of 76.49%, and system accuracy of 72.02%. For the lowest accuracy obtained in the 4th and 5th testing with an accuracy value of 66.07%. The system that has been made has also been able to predict with the highest average prediction of 60.31%, and the highest prediction value of 65.47%.



This is an open access article under the [CC-BY SA](https://creativecommons.org/licenses/by-sa/4.0/) license.

I. PENDAHULUAN

Pengklasifikasian objek/gambar adalah tugas yang cukup mudah bagi manusia, akan tetapi bagi komputer/mesin pengklasifikasian objek/gambar adalah tugas yang sangat kompleks, sehingga klasifikasi objek/gambar menjadi masalah utama dalam bidang *Computer Vision* yang sejak dahulu dicari pemecahannya. Memeriksa dan mengklasifikasikan gambar secara manual bisa menjadi tugas yang membosankan terutama ketika jumlahnya sangat banyak (misalnya 10.000) dan karenanya akan sangat berguna jika kita dapat mengotomatiskan seluruh proses ini menggunakan *Computer Vision*.

Dengan berkembangnya dunia komputasi dan dengan semakin meningkatnya kapasitas dan kecerdasan proses komputer saat ini muncul ilmu komputasi yang memungkinkan komputer dapat mengambil informasi dari suatu citra untuk keperluan pengenalan objek secara otomatis.

Metode yang paling banyak digunakan dalam pengolahan citra adalah metode *Convolutional Neural Network* (CNN). CNN merupakan pengembangan dari Multi Layer Perceptron (MLP) dan merupakan salah satu algoritma dari *Deep Learning*. Metode CNN memiliki hasil paling signifikan dalam pengenalan citra, hal tersebut dikarenakan CNN berusaha meniru sistem pengenalan citra pada visual cortex manusia, sehingga memiliki kemampuan mengolah informasi citra.

Ada banyak algoritma yang digunakan untuk pengklasifikasian gambar sebelum CNN menjadi sangat populer. Orang-orang biasa membuat fitur dari gambar dan kemudian memasukkan fitur-fitur itu ke dalam beberapa algoritma klasifikasi seperti SVM. Beberapa algoritma juga menggunakan nilai level piksel gambar sebagai vektor fitur. Sebagai contoh, melatih SVM dengan 784 fitur di mana setiap fitur adalah nilai piksel untuk gambar 28x28.

CNN mengikuti model hierarkis yang berfungsi untuk membangun jaringan, seperti corong, dan akhirnya memberikan lapisan yang sepenuhnya terhubung di mana

semua neuron terhubung satu sama lain dan output diproses. Konsep ini pertama kali disajikan oleh Yann le cun [1] untuk klasifikasi digit di mana ia menggunakan lapisan konvolusi tunggal. Hal itu kemudian dipopulerkan oleh Alexnet pada tahun 2012 yang menggunakan beberapa lapisan konvolusi pada imagenet sehingga menjadikan CNN suatu algoritma pilihan untuk tantangan klasifikasi gambar selanjutnya.

Penelitian sebelumnya yang dilakukan oleh [2] menyimpulkan bahwa kursi, kereta dan lemari pakaian diprediksi sempurna dengan 147 layer, sedangkan objek seperti mobil diprediksi sempurna dengan 177 layer. Selain itu juga disimpulkan bahwa semakin banyak jumlah layer dan semakin banyak pelatihan mengakibatkan semakin tinggi tingkat akurasi dalam prediksi.

Penelitian yang dilakukan oleh [3] pada basis data caltech 101 mendapatkan hasil bahwa klasifikasi citra objek dengan tingkat *confusion* yang berbeda menghasilkan nilai akurasi sebesar 20% - 50 %, sehingga disimpulkan bahwa metode CNN relatif handal terhadap perubahan parameter yang dilakukan.

Penelitian yang dilakukan oleh [4] menyimpulkan bahwa akurasi model CNN lebih baik daripada akurasi model MLP dalam pengenalan karakter Jawa tulisan tangan. Namun, model CNN perlu waktu lebih lama untuk dilatih dibandingkan dengan model MLP. Selain itu keakuratan model CNN untuk dataset karakter Jawa tulisan tangan tidak dapat mencapai 90%.

Penelitian yang dilakukan oleh [5] mendapatkan hasil bahwa untuk setiap klasifikasi gambar biner, fungsi aktivasi Relu dan kombinasi classifier Sigmoid memberikan akurasi klasifikasi yang lebih baik (90,54%), daripada kombinasi fungsi aktivasi dan classifier lainnya. Sehingga disimpulkan bahwa pada sistem CPU, fungsi aktivasi Relu dan Sigmoid classifier memberikan akurasi klasifikasi yang lebih baik untuk klasifikasi gambar biner.

Penelitian yang dilakukan oleh [6] mendapatkan hasil bahwa keakuratan sistem untuk mengidentifikasi dan mengenali citra jamur sangat dipengaruhi oleh parameter jaringan internal, yaitu laju pembelajaran, jumlah neuron layar tersembunyi, dan iterasi. Persentase akurasi tertinggi mencapai 93% itu berarti hasilnya bagus, dan untuk penelitian lebih lanjut dapat dicoba tanpa mengubah ke skala abu abu terlebih dahulu.

Penelitian ini dilakukan untuk melakukan klasifikasi citra dengan metode CNN dan *K-fold Cross Validation*. Beberapa topik pada penelitian sebelumnya dapat digunakan sebagai landasan alur pikir dalam perancangan dan pengembangan serta dalam penerapannya menyesuaikan dengan kebutuhan dan perkembangan teknologi terkini yang relevan.

II. METODE PENELITIAN

Untuk mencapai tujuan yang diharapkan sesuai dengan latar belakang, dilakukan beberapa tahapan yaitu antara lain, (1) Tahap Pengumpulan Data, (2) Preprocessing, (3) Pemodelan Klasifikasi dengan CNN, (4) Uji validasi hasil, dan (5) Kesimpulan.

A. Pengumpulan Data

Penelitian ini membutuhkan data untuk proses pengklasifikasian citra. Data yang dibutuhkan adalah citra atau objek tanaman yaitu bunga mawar, tulip dan matahari. Pengumpulan citra didapatkan dari pencarian acak di internet, melalui google image dan kaggle.com.

B. Pengolahan Data Awal (Preprocessing)

Agar diperoleh hasil akurasi yang maksimal maka perlu dilakukan pengolahan data awal (*preprocessing*) [7][8]. Pada penelitian ini ada dua proses yang dilakukan dalam pengolahan data awal, yaitu :

1. Cropping

Cropping digunakan untuk membuat gambar menjadi lebih fokus serta seragam. *Cropping* adalah penghilangan area terluar yang tidak diinginkan dari gambar. Proses ini biasanya terdiri dari penghapusan beberapa area periferal gambar untuk menghilangkan sampah asing dari gambar, untuk meningkatkan pembingkaiannya dan untuk mengubah rasio aspek. Proses ini dilakukan menggunakan software pengolah gambar dan dilakukan secara manual.

2. Pembersihan Noise

Noise adalah produk sampingan pengambilan gambar yang tidak diinginkan yang mengaburkan informasi yang diinginkan. Pembersihan *Noise* dilakukan untuk meningkatkan kualitas gambar sehingga nantinya akan menghasilkan nilai akurasi yang tinggi. Proses pembersihan *noise* ini juga dilakukan secara manual dengan menggunakan bantuan software pengolah gambar.

Contoh gambar yang telah dilakukan tahapan *preprocessing* dapat dilihat di gambar 1.



Gambar 1. Gambar setelah dilakukan tahapan *preprocessing*

C. Convolutional Neural Network

CNN adalah jenis jaringan saraf tiruan yang digunakan dalam pengenalan dan pemrosesan gambar. CNN meniru cara sel-sel saraf kita berkomunikasi dengan *neuron* yang saling berhubungan dan CNN memiliki arsitektur yang sama. Apa yang membuatnya unik dari jaringan saraf lain

adalah operasi konvolusional yang menerapkan filter pada setiap bagian dari input sebelumnya untuk mengekstraksi pola dan *features maps*.

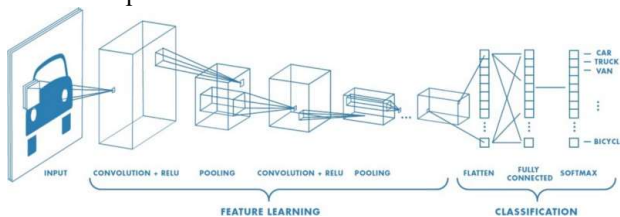
Ahli statistik dan Peneliti telah mencari ide-ide jaringan saraf di abad ke-20 untuk mengerjakan Pengenalan Pola. Salah satu perkembangan terkenal adalah *Neocognitron* oleh Fukushima pada tahun 1980 yang memiliki sifat unik tidak terpengaruh oleh perubahan posisi, untuk pengenalan pola.

Tetapi salah satu penelitian paling populer di bidang ini adalah pengembangan LeNet-5 oleh LeCun[1][3][9] dan kawan-kawan pada tahun 1997. Ini adalah salah satu dari CNN pertama yang digunakan di bank untuk membaca cek secara real-time, LeNet-5 membaca lebih dari satu juta cek. Meskipun ada algoritma lain seperti *Support Vector Machine* yang akurasi mendekati LeNet-5, namun banyak pendapat mengatakan bahwa kecepatan perhitungan CNN secara eksponensial lebih cepat daripada algoritma lain.

Pada tahun 2010, untuk mendukung penelitian di bidang visi Komputer, ImageNet lahir. ImageNet adalah tempat penyimpanan dataset gambar yang sangat besar dan memiliki kompetisi yang terbuka setiap tahun untuk mempromosikan penelitian. Pada 2012, pemenang kompetisi ImageNet adalah model AlexNet oleh Alex Krizhevsky. AlexNet adalah model CNN[10][11][12] yang mirip dengan LeNet-5 tetapi lebih signifikan dalam beberapa cara yang berdampak pada perkembangan *Artificial Intelligence*.

CNN menggunakan GPU (bukan CPU) untuk komputasinya. Dengan menggunakan platform Nvidia CUDA membuat CNN jauh lebih cepat daripada model yang dilatih dengan CPU. Beberapa komponen utama yang ada dalam CNN yaitu diantaranya adalah :

- Convolution Layer
- Pooling Layer
- Fully Connected Layer
- Dropout

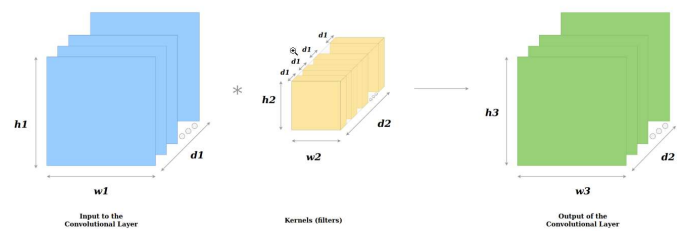


Gambar 2. Proses Convolutional Neural Network

(Sumber: <https://www.mathworks.com/discovery/convolutional-neural-network.html>)

1) Convolutional Layer

Blok bangunan utama CNN adalah *Convolution layer*. *Convolution* adalah operasi matematika untuk menggabungkan dua set informasi. *Convolution* diterapkan pada data input menggunakan filter *convolution* untuk menghasilkan *feature map*.



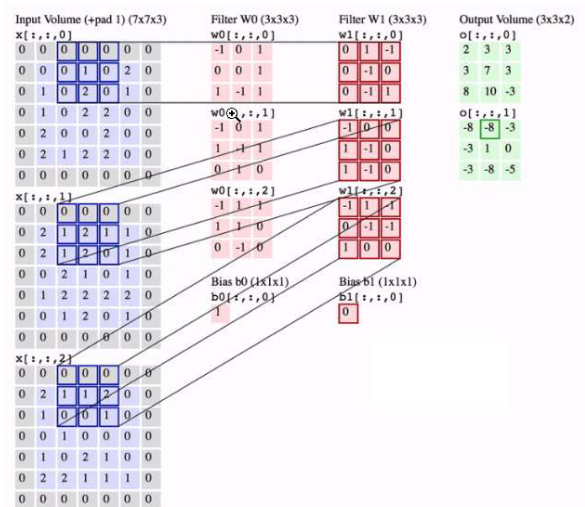
Gambar 3. Convolutional Layer

Matriks biru pada gambar 2 adalah input dengan dimensi $[h1 * w1 * d1]$. Kemudian *kernel* (filter) adalah matriks dengan dimensi $[h2 * w2 * d1]$, yang merupakan satu kuboid kuning dari beberapa kuboid (*kernel*) yang ditumpuk satu sama lain (dalam lapisan *kernel*) pada gambar di atas. Untuk setiap lapisan konvolusional, ada beberapa *kernel* bertumpuk di atas satu sama lain, inilah yang membentuk matriks 3 dimensi kuning pada Gambar 2, yang merupakan dimensi $[h2 * w2 * d2]$, di mana $d2$ adalah jumlah *kernel*. Untuk setiap *kernel*, memiliki bias masing-masing, yang merupakan jumlah skalar.

Matriks berwarna hijau pada Gambar 2, adalah output yang memiliki dimensi $[h3 * w3 * d2]$. Kedalaman ($d1$) dari *input* dan satu *kernel* adalah sama. Kedalaman ($d2$) dari *output* sama dengan jumlah *kernel* (yaitu kedalaman matriks 3 dimensi berwarna oranye).

Untuk setiap posisi *kernel* pada gambar, setiap angka dikalikan dengan angka yang sesuai pada matriks input (matriks biru) dan kemudian mereka semua dirangkum untuk nilai dalam posisi yang sesuai dalam matriks output (matriks hijau).

Dengan $d1 > 1$, hal yang sama terjadi untuk masing-masing saluran dan kemudian mereka ditambahkan bersama-sama dan kemudian disimpulkan dengan bias dari masing-masing filter dan ini membentuk nilai pada posisi yang sesuai dari matriks output. Seperti terlihat pada gambar 4.



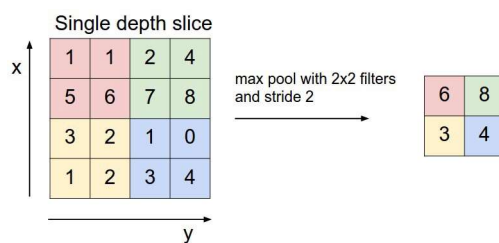
Gambar 4. Pemetaan Input ke Output

2) Pooling Layer

Tujuan utama dari *Pooling Layer* adalah untuk mengurangi jumlah parameter dari tensor input [13][14][15], sehingga dapat :

- Membantu mengurangi overfitting
- Ekstrak fitur representatif dari tensor input
- Mengurangi perhitungan dan dengan demikian membantu efisiensi

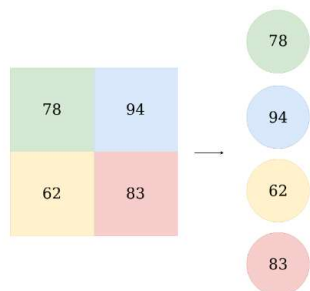
Ada dua jenis type dari *Pooling* yaitu *Max Pooling* dan *Average Pooling*. Contoh yang ditunjukkan pada Gambar 4, kernel ukuran $n * n$ (2×2) dipindahkan melintasi matriks dan untuk setiap posisi nilai maksimal diambil dan dimasukkan ke dalam posisi matriks keluaran yang sesuai, ini disebut *Max Pooling*. Dalam kasus *Average Pooling*, kernel dengan ukuran $n * n$ dipindahkan melintasi matriks dan untuk setiap posisi rata-rata diambil dari semua nilai dan dimasukkan ke dalam posisi yang sesuai dari matriks keluaran. Proses ini diulangi untuk setiap saluran dalam tensor *input*, hingga mendapatkan tensor *output*. Satu hal yang perlu diperhatikan adalah, *Pooling downsamples* gambar dalam tinggi dan lebar, tetapi jumlah *channels* (kedalaman) tetap sama.



Gambar 5. Max Pooling

3) Fully Connected Layer

Output dari *final pooling* atau *convolutional layer*, yang telah di *flaten* kemudian dimasukkan ke dalam *fully connected layer*. Hasil dari *final pooling* dan *convolutional layer* adalah matriks 3 dimensi, untuk melakukan *flaten* yaitu dengan merubah semua nilainya menjadi *vector*, seperti pada gambar 6.



Gambar 6. Flaten

Flatened vector ini kemudian dihubungkan ke beberapa *fully connected layer* yang sama dengan jaringan syaraf tiruan dan melakukan operasi matematika yang sama. Untuk setiap lapisan jaringan syaraf tiruan, digunakan perhitungan sebagai berikut : $\mathbf{g}(\mathbf{W}\mathbf{x} + \mathbf{b})$

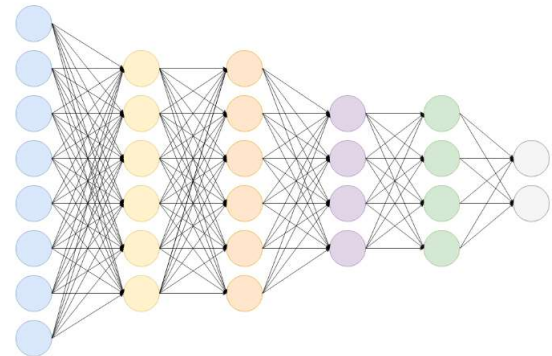
\mathbf{x} : vektor input dengan dimensi $[p_l, 1]$

\mathbf{W} : matriks bobot dengan dimensi $[p_l, n_l]$ di mana, p_l adalah jumlah *neuron* di lapisan sebelumnya dan n_l adalah jumlah *neuron* di lapisan saat ini.

\mathbf{b} : vektor bias dengan dimensi $[p_l, 1]$

\mathbf{g} : fungsi aktivasi, yang biasanya adalah *ReLU*.

Perhitungan diatas diulang untuk setiap lapisan. Setelah melewati *fully connected layer*, lapisan terakhir menggunakan fungsi aktivasi softmax (bukan ReLU) yang digunakan untuk mendapatkan probabilitas input berada di kelas tertentu (klasifikasi).

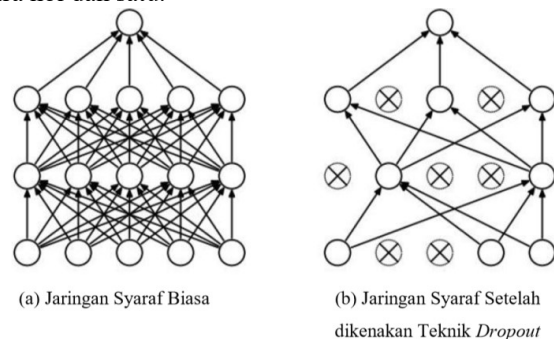


Gambar 7. Fully Connected Layer

4) Dropout

Dropout adalah teknik regularisasi jaringan syaraf dimana beberapa *neuron* akan dipilih secara acak dan tidak dipakai selama pelatihan. *Neuron* *neuron* ini dapat dibuang secara acak. Hal ini berarti bahwa kontribusi *neuron* yang dibuang akan diberhentikan sementara jaringan dan bobot baru juga tidak diterapkan pada *neuron* pada saat melakukan *backpropagation*.

Dropout merupakan proses mencegah terjadinya *overfitting* [16][17] dan juga mempercepat proses *learning*. *Dropout* mengacu kepada menghilangkan *neuron* yang berupa *hidden* maupun *layer* yang *visible* didalam jaringan. Dengan menghilangkan suatu *neuron*, berarti menghilangkannya sementara dari jaringan yang ada. *Neuron* yang akan dihilangkan akan dipilih secara acak. Setiap *neuron* akan diberikan *probabilitas* yang bernilai antara nol dan satu.



Gambar 8. Contoh Implementasi Dropout

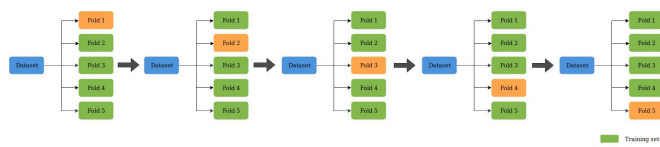
Pada gambar 8 jaringan syaraf (a) merupakan jaringan syaraf biasa dengan dua lapisan tersembunyi. Sedangkan

pada bagian (b) jaringan syaraf sudah diaplikasikan teknik *regularisasi dropout* dimana ada beberapa *neuron* aktivasi yang tidak dipakai lagi. Teknik ini sangat mudah diimplementasikan pada model CNN dan akan berdampak pada performa model dalam melatih serta mengurangi *overfitting*.

D. K Fold Cross Validation

Mengevaluasi model *Machine Learning* bisa sangat sulit. Biasanya, kita membagi set data menjadi set pelatihan dan pengujian. Kemudian menggunakan set pelatihan untuk melatih model dan set pengujian untuk menguji model. Kita kemudian mengevaluasi kinerja model berdasarkan matrik kesalahan untuk menentukan keakuratan model. Namun metode ini, tidak terlalu dapat diandalkan karena akurasi yang diperoleh untuk satu set tes bisa sangat berbeda dengan akurasi yang diperoleh untuk set tes yang berbeda. *K-fold Cross Validation* (CV) memberikan solusi untuk masalah ini dengan membagi data menjadi *fold* dan memastikan bahwa setiap *fold* digunakan sebagai set pengujian di beberapa titik CV.

K-Fold CV adalah kumpulan data yang diberikan dibagi menjadi sejumlah K bagian / *fold* di mana setiap *fold* digunakan sebagai set pengujian di beberapa titik [10][18][19]. Mari kita ambil skenario validasi silang 5-*fold* ($K = 5$). Kumpulan data dibagi menjadi 5 *fold*. Pada iterasi pertama, *fold* pertama digunakan untuk menguji model dan sisanya digunakan untuk melatih model. Pada iterasi kedua, *fold* kedua digunakan sebagai set pengujian sementara sisanya berfungsi sebagai set pelatihan. Proses ini diulangi sampai setiap *fold* dari 5 *fold* telah digunakan sebagai set pengujian.



Gambar 9. 5-Fold Cross Validation

III. HASIL DAN PEMBAHASAN

A. Dataset

Dataset yang digunakan dalam penelitian ini merupakan *sample* acak yang didapatkan dari google image sejumlah total 2100, dimana 700 data adalah kategori bunga mawar, 700 data adalah kategori bunga tulip dan 700 data lainnya adalah data kategori bunga matahari. Agar lebih jelas dapat dilihat seperti pada tabel 1.

TABEL I
PERSENTASE DATASET

Kategori	Jumlah Data	Training	Prosen %	Testing	Persen %
Mawar	700	560	80	140	20
Tulip	700	560	80	140	20
Matahari	700	560	80	140	20

B. Performa Metode CNN

Metode klasifikasi CNN yang dilakukan dengan cara membagi dataset menjadi data *train* dan data *testing*, dengan persentase 80% data *train* dan 20 % data *testing*, seperti di tabel I.

Training yang dilakukan dengan dilakukan dengan data gambar berukuran 32 x 32 dengan epochs sebanyak 30 dan jumlah kedalaman layer sebanyak 3 layer mendapatkan nilai akurasi 85%. Hal ini seperti yang ditunjukkan pada gambar 10.

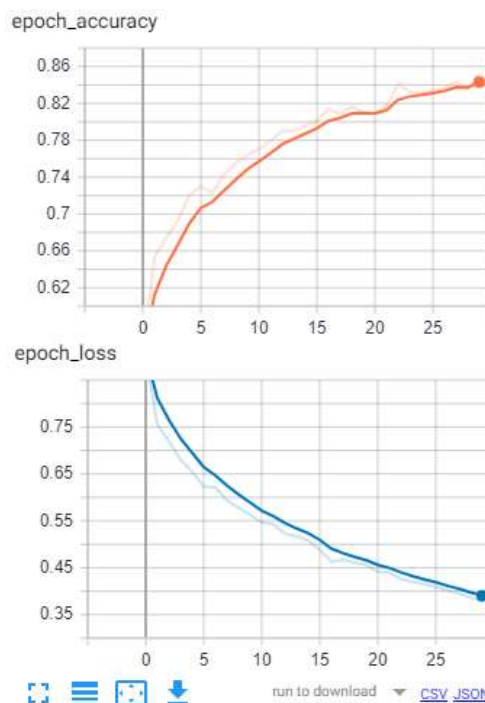
```

0.4263 - accuracy: 0.8411
Epoch 24/30
105/105 [=====] - 13s 121ms/step - loss:
0.4198 - accuracy: 0.8327
Epoch 25/30
105/105 [=====] - 13s 126ms/step - loss:
0.4154 - accuracy: 0.8315
Epoch 26/30
105/105 [=====] - 13s 126ms/step - loss:
0.4087 - accuracy: 0.8345
Epoch 27/30
105/105 [=====] - 13s 127ms/step - loss:
0.4013 - accuracy: 0.8369
Epoch 28/30
105/105 [=====] - 13s 127ms/step - loss:
0.3954 - accuracy: 0.8435
Epoch 29/30
105/105 [=====] - 13s 127ms/step - loss:
0.3858 - accuracy: 0.8363
Epoch 30/30
105/105 [=====] - 13s 122ms/step - loss:
0.3795 - accuracy: 0.8524

```

Gambar 10. Proses Training

Untuk mempermudah pengamatan nilai akurasi dan nilai *loss* ditampilkan dalam bentuk grafik *line* seperti pada gambar 11.



Gambar 11. Grafik Training Akurasi dan Loss

Untuk hasil *testing* yang dilakukan dengan menggunakan model hasil *training* didapatkan nilai prediksi sebesar 50 %, dengan prediksi benar sebanyak 210 dari total 420 data. Seperti terlihat pada gambar 12.

```
data/testing/tulip_train (96).jpg
Predicted: Tulip

data/testing/tulip_train (97).jpg
Predicted: Tulip

data/testing/tulip_train (98).jpg
Predicted: Tulip

data/testing/tulip_train (99).jpg
Predicted: Tulip

Execution Time: 5.704864978790283 seconds

Total data Testing adalah :420
Prediksi Bunga Matahari Benar : 93 Prediksi Salah : 47
Prediksi Bunga Mawar Benar : 81 Prediksi Salah : 59
Prediksi Bunga Tulip Benar : 36 Prediksi Salah : 104
```

Gambar 12. Akurasi Hasil Prediksi CNN

Agar lebih mudah diamati maka dapat dijabarkan seperti terlihat pada tabel 2.

TABEL III
AKURASI PREDIKSI CNN

Kategori Bunga	Prediksi Benar	Prediksi Salah	Persentase %
Matahari	93	47	22,14
Mawar	81	59	19,29
Tulip	36	104	8,57
Total			50

C. Uji Validasi Hasil

Algoritma yang digunakan untuk menguji validitas hasil akurasi adalah *k=5 Cross validation*. *Dataset* dibagi dalam 5 *fold* yang didalamnya terdapat 140 *dataset* dan pada setiap iterasi, diambil satu *fold* sebagai *dataset testing* dan yang lain dipakai sebagai *dataset training*. Untuk pemilihan *dataset testing* disesuaikan dengan urutan iterasi dengan urutan *fold*, yaitu iterasi ke-1 *fold 1*, iterasi ke-2 *fold 2* dan seterusnya. Proses *training* dilakukan sebanyak 3 kali dalam setiap iterasi. Setiap selesai melakukan *training* langsung dilakukan *testing* untuk mencari nilai prediksi, kemudian tingkat akurasinya dihitung secara rata-rata. Hasil pengujian menggunakan *k=5 cross validation* dapat dilihat pada tabel 3.

TABEL IIIII
AKURASI HASIL TRAINING DENGAN METODE K-FOLD CROSS VALIDATION

Iterasi ke	Training n-1 %	Training n-2 %	Training n-3 %	Rata-rata %
1	70,54	67,86	70,54	69,64
2	74,11	71,43	76,79	74,11
3	75	80,36	74,11	76,49
4	73,21	70,54	66,07	69,94
5	76,79	66,96	66,07	69,94
Rata-rata	73,93	71,43	70,71	72,02

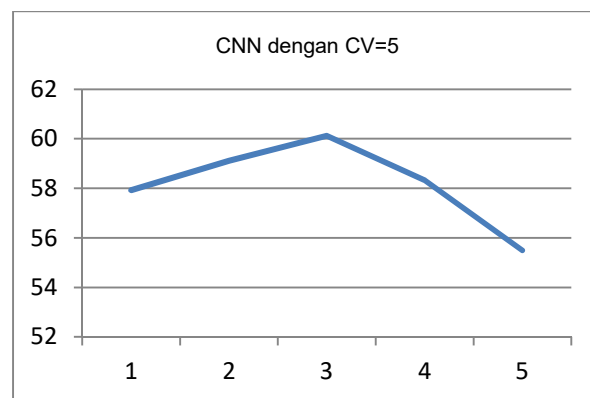
Hasil rata-rata akurasi tertinggi yaitu 76,49 %, didapat pada iterasi ke-3. Sedangkan untuk rata-rata nilai akurasi terendah didapat pada iterasi ke-1 yaitu 69,64 %. Iterasi ke-3 pada *training* ke-2 mendapatkan nilai akurasi tertinggi yaitu sebesar 80,36 %. Nilai akurasi sistem sebesar 72,02 didapatkan dari hasil rata-rata setiap iterasi.

Testing yang dilakukan setelah *training* disetiap iterasi menghasilkan prediksi seperti terlihat pada tabel 4.

TABEL IVV
AKURASI HASIL PREDIKSI DENGAN METODE K-FOLD CROSS VALIDATION

Iterasi ke	Testing n-1 %	Testing n-2 %	Testing n-3 %	Rata-rata %
1	58,33	57,14	58,33	57,93
2	60,71	54,76	61,9	59,12
3	55,96	65,47	59,52	60,31
4	59,52	54,76	60,71	58,33
5	54,76	53,38	58,33	55,49
Rata-rata	57,86	57,10	59,76	58,23

Dari tabel 4 dapat kita lihat hasil rata-rata prediksi tertinggi yaitu 60,31 %, didapat pada iterasi ke-3. Sedangkan untuk rata-rata nilai prediksi terendah didapat pada iterasi ke-5 yaitu 55,49 %. Iterasi ke-3 pada *testing* ke-2 mendapat nilai prediksi tertinggi yaitu sebesar 65,47 %. Kemudian untuk nilai prediksi sistem sebesar 58,23 didapatkan dari hasil rata-rata setiap iterasi. Agar mempermudah pengamatan hasil dari *k=5 cross validation* maka dapat digambarkan dalam bentuk grafik seperti terlihat pada gambar 13.



Gambar 13. Grafik Prediksi CNN dengan Cross Validation

IV. KESIMPULAN

Metode CNN berhasil diimplementasikan untuk pengklasifikasian citra menggunakan *library* keras dan *tensorflow* dengan bahasa pemrograman Python. Dari hasil penelitian didapatkan tingkat kecocokan / akurasi tertinggi sebesar 80,36% dan rata-rata akurasi tertinggi yaitu 76,49 %, serta akurasi sistem sebesar 72,02 %. Untuk akurasi terendah di dapat pada iterasi ke-4 dan 5 dengan nilai akurasi

sebesar 66,07 %. Sistem yang telah dibuat juga telah dapat memprediksi dengan rata-rata prediksi tertinggi yaitu 60,31 %, dan nilai prediksi tertinggi yaitu sebesar 65,47 %. Tahap penelitian selanjutnya dapat dilakukan dengan melakukan perbandingan algoritma CNN dengan algoritma klasifikasi lainnya seperti *Support Vector Machine* (SVM). Algoritma dengan kinerja terbaik dapat dikembangkan lebih lanjut sebagai sebuah sistem Android, sebagai alat prediksi.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada pihak yang telah banyak memberikan bantuan dan dukungan yang tiada terhingga baik secara langsung maupun tidak langsung. Kepada Tuhan YME, Ibu, Istri, Kedua Putriku, Universitas Ahmad Dahlan dan teman-teman MTI angkatan 8.

DAFTAR PUSTAKA

- [1] L. D. J. Le Cun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, "LeNet-5: Handwritten Digit Recognition with a Back-Propagation Network," *Dermatologic Surg.*, vol. 39, no. 1pt2, p. 149, 1998.
- [2] N. Sharma, V. Jain, and A. Mishra, "An Analysis of Convolutional Neural Networks for Image Classification," *Procedia Comput. Sci.*, vol. 132, no. Iccids, pp. 377–384, 2018.
- [3] W. S. Eka Putra, "Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101," *J. Tek. ITS*, vol. 5, no. 1, 2016.
- [4] C. K. Dewa, A. L. Fadhillah, and A. Afiahayati, "Convolutional Neural Networks for Handwritten Javanese Character Recognition," *IJCCS (Indonesian J. Comput. Cybern. Syst.)*, vol. 12, no. 1, p. 83, 2018.
- [5] K. Chauhan and S. Ram, "International Journal of Advance Engineering and Research Image Classification with Deep Learning and Comparison between Different Convolutional Neural Network Structures using Tensorflow and Keras," pp. 533–538, 2018.
- [6] A. Fadlil, R. Umar, and S. Gustina, "Mushroom Images Identification Using Orde 1 Statistics Feature Extraction with Artificial Neural Network Classification Technique Mushroom Images Identification Using Orde 1 Statistics Feature Extraction with Artificial Neural Network Classification Techn," 2019.
- [7] S. Saifullah, S. Sunardi, and A. Yudhana, "Perbandingan Segmentasi Pada Citra Asli Dan Citra Kompresi Wavelet Untuk Identifikasi Telur," *Ilk. J. Ilm.*, vol. 8, no. 3, p. 190, 2016.
- [8] I. Riadi, R. Umar, and F. D. Aini, "Analisis Perbandingan Detection Traffic Anomaly Dengan Metode Naive Bayes Dan Support Vector Machine (Svm)," *Ilk. J. Ilm.*, vol. 11, no. 1, p. 17, 2019.
- [9] H. Darmanto, D. Learning, T. Learning, and G. Descent, "PENGENALAN SPESIES IKAN BERDASARKAN KONTUR OTOLITH," vol. 2, 2019.
- [10] S. Ilahiyah and A. Nilogiri, "Implementasi Deep Learning Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan Convolutional Neural Network," vol. 3, no. 2, pp. 49–56, 2018.
- [11] E. N. Arrofiqoh and H. Harintaka, "Implementasi Metode Convolutional Neural Network Untuk Klasifikasi Tanaman Pada Citra Resolusi Tinggi," *Geomatika*, vol. 24, no. 2, p. 61, 2018.
- [12] D. C. K. Putu Aryasuta Wicaksana, I Made Sudarma, "Pengenalan Pola Motif Kain Tenun Gringsing Menggunakan Metode Convolutional Neural Network Dengan Model Arsitektur," vol. 6, no. 3, pp. 159–168, 2019.
- [13] K. O'Shea and R. Nash, "An Introduction to Convolutional Neural Networks," pp. 1–11, 2015.
- [14] B. J. Erickson, P. Korfiatis, Z. Akkus, T. Kline, and K. Philbrick, "Toolkits and Libraries for Deep Learning," *J. Digit. Imaging*, vol. 30, no. 4, pp. 400–405, 2017.
- [15] C. Y. Lee, P. W. Gallagher, and Z. Tu, "Generalizing pooling functions in convolutional neural networks: Mixed, gated, and tree," *Proc. 19th Int. Conf. Artif. Intell. Stat. AISTATS 2016*, pp. 464–472, 2016.
- [16] H. Abhirawan, Jondri, and A. Arifianto, "Pengenalan Wajah Menggunakan Convolutional Neural Networks (CNN)," *Univ. Telkom*, vol. 4, no. 3, pp. 4907–4916, 2017.
- [17] A. Santoso and G. Ariyanto, "Implementasi Deep Learning Berbasis Keras Untuk Pengenalan Wajah," *Emit. J. Tek. Elektro*, vol. 18, no. 01, pp. 15–21, 2018.
- [18] C. Applications, "Mathematical and Computational Applications," vol. 16, no. 3, pp. 702–711, 2011.
- [19] B. Hu, Z. Lu, H. Li, and Q. Chen, "Convolutional neural network architectures for matching natural language sentences," *Adv. Neural Inf. Process. Syst.*, vol. 3, no. January, pp. 2042–2050, 2014.
- [20] R. A. Surya, A. Fadlil, and A. Yudhana, "Ekstraksi Ciri Metode Gray Level Co-Occurrence Matrix (GLCM) dan Filter Gabor untuk Klasifikasi Citra Batik Pekalongan," *J. Inform. Pengemb. IT (JPIT)*, Vol. 02, No. 02, Juli 2017, vol. 02, no. 02, pp. 23–26, 2017.
- [21] S. Gustina et al., "Identifikasi Tanaman Kamboja menggunakan Ekstraksi Ciri Citra Daun dan Jaringan Syaraf Tiruan," vol. 2, no. 1, pp. 128–132, 2016.
- [22] S. Saifullah, S. -, and A. Yudhana, "Analisis Perbandingan Pengolahan Citra Asli Dan Hasil Cropping Untuk Identifikasi Telur," *J. Tek. Inform. dan Sist. Inf.*, vol. 2, no. 3, pp. 341–350, 2016.