

juti dimas

by Dimas Fanny Hebrasianto Permadi

Submission date: 27-Aug-2022 10:30PM (UTC-0700)

Submission ID: 1885368745

File name: jurnal_juti.pdf (594.64K)

Word count: 5692

Character count: 33070

IMPLEMENTASI PENGENALAN CITRA WAJAH BINATANG DENGAN ALGORITME *CONVOLUTIONAL NEURAL NETWORK* (CNN)

Dimas Fanny Hebrasianto Permadi¹⁾ dan Moch Zawaruddin Abdullah²⁾

¹⁾ Program Studi Sistem Komputer, Fakultas Teknologi Informasi, Universitas Islam Balitar Jl. Mojopahit No 4, Sananwetan, Kec. Sananwetan, Kota Blitar, Jawa Timur, Indonesia 6613

²⁾ Jurusan Teknologi Informasi, Politeknik Negeri Malang Jl. Soekarno Hatta No.9, Jatimulyo, Kec. Lowokwaru, Kota Malang, Jawa Timur, Indonesia 65141
e-mail: dimas.fanny92@gmail.com¹⁾, zawaruddin@polinema.ac.id²⁾

ABSTRAK

Penelitian ini dimaksudkan untuk melakukan implementasi pada Algoritme Convolutional Neural Network (CNN) terhadap cabang ilmu Computer Vision. Algoritme CNN merupakan pengembangan dari Neural Network yang dikombinasikan dengan Multilayer Perceptron yang menerapkan teknik konvolusi pada proses fitur ekstraksinya. Algoritma CNN ini diimplementasikan pada dataset Citra wajah Binatang sebanyak 16.130 citra yang terdiri dari 3 kelas berbeda, yaitu Cat, Dog, dan Wild. Tujuan penelitian ini untuk mendeteksi hasil citra wajah binatang mana yang termasuk kelas Kucing, Anjing, atau Binatang Liar yang menyerupai kucing ataupun Anjing seperti Singa, Harimau, Hyena, Serigala, dll. Hasil penelitian ini berupa analisis pengujian Algoritme CNN dengan membandingkan hasil learning rate dan epoch yang terbaik secara berturut-turut yaitu $10e-4$ dan 60. Dari learning rate dan epoch tersebut didapatkan hasil akurasi sebesar 97.22% atau 116.33 dari 120 citra yang menggunakan citra acak dari dataset tersebut. Jika dengan menggunakan citra selain dari dataset, maka mendapatkan nilai akurasi sebesar 77.78% atau 93.33 dari 120 citra.

Kata Kunci: Convolutional Neural Network (CNN); Deep Learning; Machine Learning; Multilayer perceptron (MLP), Klasifikasi Wajah Binatang.

IMPLEMENTATION OF ANIMAL FACE'S RECOGNITION BY CONVOLUTIONAL NEURAL NETWORK (CNN) ALGORITHM

Dimas Fanny Hebrasianto Permadi¹⁾ dan Moch Zawaruddin Abdullah²⁾

¹⁾ Program Studi Sistem Komputer, Fakultas Teknologi Informasi, Universitas Islam Balitar Jl. Mojopahit No 4, Sananwetan, Kec. Sananwetan, Kota Blitar, Jawa Timur, Indonesia 6613

²⁾ Jurusan Teknologi Informasi, Politeknik Negeri Malang Jl. Soekarno Hatta No.9, Jatimulyo, Kec. Lowokwaru, Kota Malang, Jawa Timur, Indonesia 65141
e-mail: dimas.fanny92@gmail.com¹⁾, zawaruddin@polinema.ac.id²⁾

ABSTRACT

This research is intended to implement the Convolutional Neural Network (CNN) algorithm for the branch of Computer Vision. CNN algorithm is a development of Neural Network combined with Multilayer Perceptron which applies convolution technique to its feature extraction process. CNN algorithm is implementing animal dataset which contains 16.130 images that consisting 3 different classes that is Cat, Dog, and Wild. The purpose of this research is detection of facial images from animals belong to the class of Cats, Dogs, or Wild Animals that resemble descending of cats or dogs likes Lion, Tiger, Hyena, Wolf, etc. The result of this research is comparing onto learning rate and epoch respectively are $10e-4$ and 60. From that learning rate and epoch can reach accuracy about 97.22% or 116.33 from 120 images using random images from the dataset. If we are using images outside the dataset, it can reach 77.78% accuracy or 93.33 from 120 images.

Keywords: Convolutional Neural Network (CNN); Deep Learning; Machine Learning; Multilayer perceptron (MLP), Animal's faces classification.

I. PENDAHULUAN

COMPUTER VISION salah satu cabang ilmu kecerdasan buatan yang menerapkan komputasi pada data berupa citra digital. Cabang ilmu ini dimaksudkan untuk dapat mengolah dan mengenali sebuah objek dari citra digital. Computer vision juga dapat dikatakan sebagai mata atau indera penglihatan pada komputer untuk mengenali objek citra digital. Beberapa kerangka kerja yang umum dilakukan dalam computer vision adalah: image acquisition, pengenalan citra, preprocessing, feature extraction, deteksi objek, klasifikasi, dan segmentasi [1].

Computer vision juga dapat melakukan klasifikasi dari suatu citra secara akurat. Pengklasifikasian citra termasuk salah satu bagian dari *Computer Vision* yang berfokus pada penerapan Algoritme untuk menentukan citra kedalam kategori tertentu dimana dataset berasal dari berbagai sumber citra. Proses klasifikasi ini dapat mempermudah pekerjaan dalam mengidentifikasi suatu citra walaupun jumlah citra yang diproses berjumlah besar. Dalam melakukan pengklasifikasian secara otomatis diperlukan teknik komputasi yang handal, salah satunya adalah teknik dengan menerapkan Algoritme *Convolutional Neural Network (CNN)* [2], [3].

CNN merupakan Algoritme pengembangan dari *Neural Network* dengan kombinasi dari *Multilayer Perceptron (MLP)* yang menggunakan teknik konvolusi pada proses ekstraksi fiturnya. Teknik konvolusi ini cocok untuk pengklasifikasian dan pengenalan gambar yang mana konvolusi juga termasuk salah satu teknik fitur ekstraksi pada gambar berupa vektor [4], [5].

Pengoperasian CNN mirip dengan MLP dimana setiap neuron direpresentasikan dalam dua dimensi sedangkan setiap neuron pada MLP hanya satu dimensi. Di CNN, operasi konvolusi digunakan untuk operasi linier sementara bobot empat dimensi adalah kumpulan inti konvolusi. Karena sifat dari proses konvolusi, CNN hanya dapat digunakan untuk data dengan struktur dua dimensi, seperti gambar dan suara [6].

Berdasarkan penelitian sebelumnya telah dilakukan pengenalan hewan berdasarkan masing-masing ras tiap jenisnya. Seperti contoh, untuk kucing dideteksi mana yang termasuk ras persia, ras angora, lokal, british, atau persilangannya. Pengenalan ras kucing yang sudah dilakukan menggunakan Algoritme CNN [7] dan beberapa juga menggunakan library OpenCV dan gabungan *Haarcascade* dengan *Viola* [8]. Sedangkan pada penelitian yang untuk mendeteksi ras anjing seperti Airedale, Appenzeller, Beagle, Pug, Chihuahua, dll menggunakan Algoritme CNN [9] dan penelitian lainnya masih menggunakan CNN juga. Hanya saja, berbeda perlakuan pada *preprocessing*-nya pada pemotongan gambar pada focus objek yang dideteksi menggunakan *You Only Look Once (YOLO)* yang selanjutnya diproses untuk menjadi model CNN-nya [10].

Jika pada penelitian sebelumnya hanya pada satu binatang yang sama tetapi, yang diklasifikasikan adalah ras masing-masing [6]–[10]. Sedangkan pada penelitian ini secara keseluruhan ras yang dimasukkan pada dataset.

Penelitian ini berkontribusi untuk melakukan klasifikasi citra dari dataset yang telah ada, yaitu dataset citra wajah binatang dengan menggunakan Algoritme CNN. Dataset terdiri dari tiga kelas citra wajah binatang, yaitu citra kelas anjing (*dog*), kelas kucing (*cat*), dan kelas binatang liar (*wild*) yaitu yang menyerupai kucing atau anjing seperti singa, harimau, serigala, hyena, puma, dan lain-lain.

II. METODE PENELITIAN

A. Dataset

Dataset yang digunakan dalam penelitian ini adalah dataset yang diambil dari *repository* website www.kaggle.com yang berisikan tentang data citra binatang dengan jumlah 16.130 citra. Dataset ini terdiri dari 3

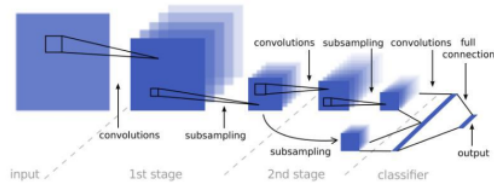
TABEL I
SAMPEL CITRA BINATANG PADA DATASET

No	CAT	Dog	Wild
1			
2			
3			
4			
5			

kelas, antara lain kelas Kucing (*Cat*) sebanyak 5653 citra, Anjing (*Dog*) sebanyak 5239 citra, dan Binatang Liar (*Wild*) sebanyak 5238 citra. Secara keseluruhan citra pada dataset menampilkan ketiga citra binatang dengan tampak dari depan (muka) binatang dan citra memiliki resolusi 512 x 512 piksel. Detail *sample* citra dengan tiga kelas dapat diketahui seperti pada Tabel 1.

B. Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) merupakan pengembangan dari *multilayer perceptron* (MLP) yang didesain untuk mengolah data dua dimensi dalam bentuk citra [11]. CNN termasuk kedalam jenis *Deep Neural Network* karena kedalaman jaringan yang tinggi dan banyak diaplikasikan pada data citra. Pada dasarnya klasifikasi



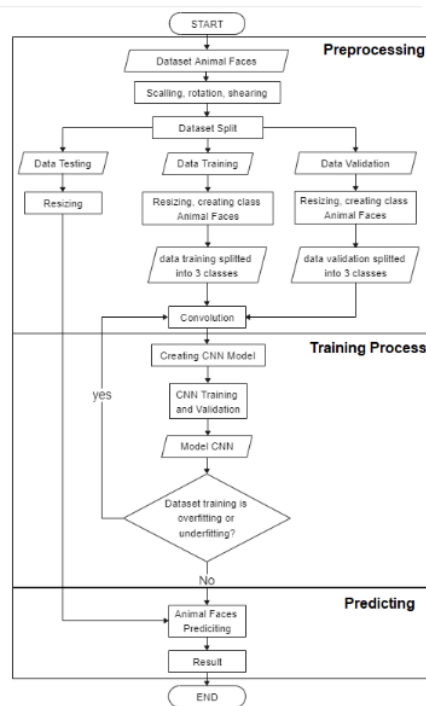
Gambar 1. Arsitektur Convolutional Neural Network (CNN)

citra dapat digunakan dengan MLP, akan tetapi dengan metode MLP kurang sesuai untuk digunakan karena tidak menyimpan informasi spasial dari data citra dan menganggap setiap piksel adalah fitur yang independen sehingga menghasilkan hasil yang kurang baik [12], [13].

CNN merupakan operasi konvolusi yang menggabungkan beberapa lapisan pemrosesan, menggunakan beberapa elemen yang beroperasi secara paralel dan terinspirasi oleh sistem saraf biologis. Pada CNN setiap neuron dipresentasikan dalam bentuk 2 dimensi, sehingga metode ini cocok untuk pemrosesan dengan input berupa citra [14].

Secara teknis, CNN adalah sebuah arsitektur yang dapat dilatih dan terdiri dari beberapa tahap. Masukan (input) dan keluaran (output) dari setiap tahap adalah terdiri dari beberapa *array* yang biasa disebut *feature map*. Setiap tahap terdiri dari tiga layer yaitu konvolusi, fungsi aktivasi layer dan *pooling layer* [12].

Gambar 1 [15] menunjukkan tahapan yang ada pada metode CNN. Tahap pertama ialah tahap konvolusi, yaitu tahap yang dilakukan dengan menggunakan kernel dengan ukuran tertentu. Perhitungan jumlah kernel yang dipakai



Gambar 2. Flowchart Penelitian

tergantung dari jumlah fitur yang dihasilkan. Kemudian dilanjutkan menuju tahap fungsi aktivasi. Fungsi aktivasi merupakan fungsi yang digunakan pada jaringan saraf untuk mengaktifkan atau tidak mengaktifkan *neuron*. Fungsi aktivasi yang sering digunakan adalah fungsi aktivasi ReLU (*Rectifier Linear Unit*), Setelah keluar dari tahap fungsi aktivasi selanjutnya melalui proses pooling. Proses ini diulang beberapa kali sampai didapatkan peta fitur yang cukup untuk dilanjutkan ke *fully connected neural network*, dan dari *fully connected network* adalah *output class*.

C. Alur Penelitian

Penelitian ini terbagi menjadi 3 sub bagian alur proses, yaitu *Preprocessing*, *Training Process*, dan *Predicting*. Untuk detail alur penelitian dapat dilihat pada Gambar 2.

1) Preprocessing

Pada proses ini dilakukan pengumpulan dataset citra binatang yang terdiri dari 3 kelas. Selanjutnya, masing-masing citra dalam dataset akan dilakukan proses *scaling*, *rotation*, dan *shearing*. Pada proses ini secara berurutan dilakukan penyesuaian resolusi citra keseluruhan dengan skala tertentu. Langkah selanjutnya dilakukan proses rotasi citra untuk citra yang tidak sesuai atau objek pada citra terbalik. Kemudian dilakukan proses *cropping* atau pemotongan beberapa bagian citra yang dibutuhkan.

Dataset split digunakan untuk membagi *dataset* menjadi 3 bagian, yaitu *Data Testing*, *Data Training*, dan *Data Validation*. *Data Testing* secara khusus terpisah dari dataset dan akan digunakan untuk uji coba model CNN yang telah dibangun. *Data Training* dan *Data Validation* akan dibagi dengan proporsi 4:1 yang mana *Data Training* mempunyai porsi 80% dan *Data Validation* mempunyai porsi 20% dari dataset secara keseluruhan. Pembagian data training dan data validation dengan porsi 4:1 dilakukan untuk mendapatkan model CNN yang memiliki akurasi yang lebih baik [16], [17].

Proses selanjutnya yaitu *Resizing* atau mengubah ukuran dimensi citra secara menyeluruh pada dataset citra. Proses *resizing* dilakukan agar citra memiliki dimensi yang sama. Ukuran dalam dimensi yang sama akan memudahkan proses, dan menyeragamkan input citra kedalam *training process*. Dalam penelitian ini, ukuran dimensi citra yang digunakan adalah 150x150 piksel.

Proses *Convolution* menjadi proses yang kritical dalam penelitian ini. Pada tahapan ini dilakukan beberapa pembagian proses konvolusi dua-dimensi yang terbagi menjadi tiga lapisan konvolusi berbeda. Kelima lapisan tersebut secara berurutan menggunakan hasil *output* dengan dimensi 16, 32, dan 64 yang menerapkan ukuran kernel 3x3. Semua lapisan tersebut juga diproses menggunakan aktivasi ReLU. ReLU tersedia pada *library* KERAS yang berguna untuk mengubah nilai negatif menjadi 0 pada hasil matriks konvolusi atau bisa disebut sebagai normalisasi nilai matriks konvolusi [4], [11], [13], [18], [19].

Proses pada masing-masing lapisan konvolusi ini juga dilakukan operasi penyatuan yang menghitung nilai maksimum setiap *patch*/bagian kecil dari citra. Penghitungan menggunakan fungsi *MaxPooling2D* pada *library* KERAS dengan ukuran *sampling* matriks 2x2 sebagai pereduksi ukuran citra dan *node* sebanyak mungkin dengan nilai *pixel loss* seminimal mungkin. Hal ini untuk menghindari data yang *overfitting* atau *underfitting* pada citra yang akan diproses. [18]

Proses pada *convolution* selanjutnya adalah melakukan proses *flattening* yang berguna untuk mengkonversi menjadi data 1 dimensi vektor tunggal. Selanjutnya dilakukan proses *dense* untuk menghubungkan lapisan-lapisan konvolusi tersebut dengan nilai 512 dan 256 secara berurutan dengan menggunakan aktivasi ReLU. Akhir proses *convolution* ini dengan diakhiri pembuatan model *sequential* yang nanti akan diproses *training* oleh *Neural Network*.

2) Training Process

Tahapan awal pada *Training Process* adalah membuat Model CNN dari hasil konvolusi. Kemudian dilakukan proses CNN *training and validation* dengan mengkompilasi data hasil konvolusi sebelumnya dengan menggunakan optimasi Adam. Optimasi Adam merupakan metode optimasi yang bekerja dengan baik pada hasil empiris pada *Machine Learning* yang diterapkan pada Tensorflow [20].

Pengecekan *data loss* dilakukan dengan pengujian *categorical cross-entropy*. Pengujian *categorical cross-entropy* digunakan karena lebih baik daripada *binary cross-entropy*. Hal ini dikarenakan *categorical cross-entropy* dapat mengklasifikasikan lebih dari 2 jenis kelas dan memiliki nilai akurasi yang tinggi [21]. Penerapan *categorical cross-entropy* dengan pembobotan standard dapat dilihat pada Persamaan 1.

$$J_{wccc} = \frac{-1}{M} \sum_{k=1}^K \sum_{m=1}^M w_k \times y_m^k \times \log(h_{\theta}(x_m, k)) \quad (1)$$

Berdasarkan Persamaan 1, dapat dijelaskan bahwa variable M adalah jumlah sampel untuk *training* dan K sebagai banyaknya kelas. w_k merupakan nilai bobot dari kelas k dan y_m^k adalah target label sampel *training* dari variable m untuk kelas k . x_m sebagai inputan dari *data training* dari m , dan h_{θ} untuk model dengan *Neural Network* dari pembobotan θ [21].

Model CNN sudah terbentuk akan dilakukan proses pengujian dengan *Data Testing*. Hasil dari proses *training* dan *validation* dilakukan pengecekan terhadap nilai akurasi, *data loss*, nilai validasi akurasi, dan validasi *data loss* sehingga tidak dinyatakan sebagai *underfitting* atau *overfitting* [22]. Pada penelitian ini iterasi pada proses *training* dan *validation* akan terus dilakukan sampai mendekati *threshold* yang ditentukan, atau ketika *epoch* mencapai target pengujian. *Threshold* yang digunakan yaitu

- nilai maksimum untuk nilai akurasi dan validasi akurasi adalah 5%
- nilai maksimal *loss* dan validasi *loss* adalah 15%
- nilai akurasi dan validasi akurasi mendekati 95%

3) C. Predicting

Proses *predicting* digunakan untuk melakukan prediksi inputan sebuah citra. Citra yang dilakukan prediksi adalah citra yang tidak termasuk dari hasil *training* dengan menggunakan CNN sebelumnya. *Data Testing* ini terdiri dari masing-masing 20 citra untuk setiap kelas yang dipilih secara acak dari berbagai sumber dan memiliki resolusi berbeda. Contoh citra yang digunakan untuk *testing* bisa dilihat pada Tabel 2.

TABEL II
 BEBERAPA RAW-SAMPLE CITRA BINATANG YANG DIGUNAKAN UNTUK DATA TESTING

No	Cat	Dog	Wild
1			
2			
3			
4			
5			

Data citra untuk *testing* diinputkan satu per satu dan dilakukan proses *resizing* citra. Setelah proses *resizing* citra, dilakukan proses pembuatan *array* untuk citra yang nantinya akan dilakukan prediksi dengan hasil model yang sudah dibuat oleh CNN.

Setelah proses prediksi, dilakukan proses pengidentifikasian kelas dari hasil prediksi. Pengidentifikasian dilakukan untuk menentukan hasil prediksi termasuk pada kelas yang sudah ada (kelas *Cat*, *Dog*, *Wild*). Dalam pengidentifikasian diperlukan pernyataan ke dalam bentuk matriks yang nantinya akan ditampilkan hasil cetak dalam bentuk *String*. Penentuan matriks berdasarkan hasil dari *compiling* dari model CNN dengan kriteria *categorical cross-entropy*. Kelas *Cat* diidentifikasi sebagai matriks [1,0,0], kelas *Dog* [0,1,0], dan kelas *Wild* [0,0,1]. Jika dilihat pola matriks dapat dinyatakan bahwa nilai hasil dari prediksi *output* yang mendekati nilai 1, maka citra inputan akan teridentifikasi sesuai dengan kelas yang sesuai. Hal ini dikarenakan nilai hasil *training* terbagi menjadi 3 kelas tersebut yang mana total nilai dari matriks tersebut adalah 1. Sehingga salah satu nilai tertinggi dari ketiga kelas maka akan dinyatakan menjadi kelas tersebut.

III. HASIL DAN PEMBAHASAN

Bab ini menjabarkan pembahasan bagaimana teknik dan skenario pengujian data, dan hasil prediksi dari skenario pengujian tersebut.

A. Hasil Model CNN

Hasil Model CNN pada penelitian ini, menghasilkan sejumlah parameter untuk hasil relasi modelling pada Machine Learning. Untuk detail hasil parameter yang didapatkan, dapat dilihat pada Tabel 3.

TABEL III
HASIL PARAMETER RELASI MODELLING DATA TRAINING DARI CNN YANG MENGGUNAKAN MODEL SEQUENTIAL

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 148, 148, 16)	448
Max_pooling2d (MaxPooling2D)	(None, 74, 74, 32)	0
conv2d_1 (Conv2D)	(None, 72, 72, 16)	4640
Max_pooling2d_1 (MaxPooling2D)	(None, 36, 36, 32)	0
conv2d_2 (Conv2D)	(None, 17, 17, 64)	18496
Max_pooling2d_2 (MaxPooling2D)	(None, 17, 17, 64)	0
dropout (Dropout)	(None, 17, 17, 64)	0
flatten (Flatten)	(None, 18496)	0
dense (Dense)	(None, 512)	9470464
dense_1 (Dense)	(None, 256)	131328
dense_2 (Dense)	(None, 3)	771

Tabel 3 menunjukkan hasil pembuatan model CNN yang mendapatkan hasil total parameter sebesar 9.626.147. Hanya saja, dari total parameter tersebut terdapat nilai parameter yaitu *Non-trainable params* yang tidak dapat di-*training* sebesar 448 berasal dari layer pertama yang dilakukan oleh konvolusi 2D. Hal ini dikarenakan, *output shape* yang dihasilkan terlalu besar sebesar 148x148 piksel dengan 16 lapisan konvolusi. Sedangkan, pada *Trainable params* sebagai nilai data yang dapat di-*training* yang berdasarkan pada *output node* jaringan saraf tiruan sebesar 9.625.699 parameter. Nilai parameter tersebut didapatkan berdasarkan akumulasi dari banyaknya tiga kali proses konvolusi 2D menggunakan fungsi Conv2D secara berturut-turut yang masing-masing menghasilkan *output shape kernel* 148x148 dengan filter 16 lapisan yang menghasilkan nilai parameter 448, kernel 72x72 dengan filter 32 lapisan yang menghasilkan 4640, dan kernel 34x34 dengan lapisan 64 lapisan yang menghasilkan 18.496 parameter. Pada masing-masing proses setelah proses Conv2D, juga dilakukan perhitungan nilai maksimum *patch* dengan fungsi MaxPooling2D dengan kernel 2x2 yang menghasilkan nilai parameter 0.

Selanjutnya proses perhitungan *flatten* yang diambil dari hasil dari fungsi Conv2D dan MaxPooling2D terakhir. Proses *flatten* sudah tidak melakukan *learning* sehingga nilai parameternya adalah 0. Proses *dense* dilakukan dengan mengambil nilai dari proses Conv2D terakhir yaitu 18.496 parameter yang dikalikan terlebih dahulu dengan 512 *output channel*, kemudian ditambah nilai 512 *output channel*. sehingga nilai parameter yang didapatkan adalah 9.470.464. Selanjutnya proses *dense* yang kedua dari nilai lapisan 512 dikalikan dengan 256 *output channel*, lalu ditambahkan dengan 256 *output channel*. Nilai parameter yang dihasilkan pada *dense* yang kedua ini adalah 131.328. Terakhir, *dense* yang ketiga, menggunakan nilai 256 yang dikalikan dengan 3 *output channel* dan ditambahkan dengan 3 *output channel* menghasilkan nilai parameter 771. Untuk hasil pemodelan CNN dapat dilihat pada Gambar 3.

```
Epoch 56/60
807/807 - 353s - loss: 0.0145 - accuracy: 0.9953 - val_loss: 0.1263 - val_accuracy: 0.9646
Epoch 57/60
807/807 - 354s - loss: 0.0144 - accuracy: 0.9957 - val_loss: 0.1196 - val_accuracy: 0.9662
Epoch 58/60
807/807 - 358s - loss: 0.0102 - accuracy: 0.9970 - val_loss: 0.1575 - val_accuracy: 0.9566
Epoch 59/60
807/807 - 352s - loss: 0.0124 - accuracy: 0.9957 - val_loss: 0.1321 - val_accuracy: 0.9653
Epoch 60/60
807/807 - 355s - loss: 0.0131 - accuracy: 0.9957 - val_loss: 0.1439 - val_accuracy: 0.9609
```

Gambar 3. Hasil pemodelan CNN

B. Skenario pengujian data

Pada tahap ini dilakukan pengujian data hasil pengenalan citra binatang dari dataset yang digunakan. Data diuji berdasarkan hasil *training* model CNN yang menyatakan data tersebut tidak *overfitting* atau *underfitting*. Sehingga dapat dikatakan dataset tersebut, tingkat akurasi dapat dinyatakan tepat dan sesuai kevalidannya. Untuk menguji tingkat akurasi setelah tidak dinyatakan *overfitting* atau *underfitting*, diperlukan citra data testing diluar dari dataset yang digunakan untuk membuktikan hasil *training* dan *validation*. Selain itu, pengecekan dataset juga dapat dilihat hasil selisih yang didapatkan dari *val_accuracy* terhadap *accuracy* dan *loss* dan *val_loss* yang tidak memiliki selisih tidak signifikan yaitu maksimal 0.25 [23].

Pengujian juga dilihat pada jumlah *epoch* dan *learning rate* yang digunakan. *Epoch* merupakan banyaknya perulangan dari iterasi dari pembuatan model *neural network*. *Learning rate* adalah beban intensitas *learning* pada *neural network* untuk mencapai nilai konvergen. Hal ini dikarenakan jika terlalu rendah nilai *learning rate* maka, akan memperlambat untuk mencapai nilai konvergen dan jika terlalu tinggi nilainya maka model CNN tidak pernah mencapai nilai konvergen [24].

Skenario pengujian dibagi berdasarkan nilai *learning rate* sebanyak 4 jenis, diantaranya dengan nilai 10^{-3} , 10^{-4} , 10^{-5} , 10^{-6} . Masing-masing *learning rate* tersebut dibatasi dengan menggunakan *epoch* = 100 [7], [25]–[27]. Pengujian pada pembagian masing-masing *learning rate* ini dilakukan perulangan sebanyak 3 kali. Sehingga mendapatkan total terdapat 12 kali perulangan pengujian data untuk mencapai hasil maksimal yang diinginkan. Hal ini dilakukan untuk mencari nilai rata-rata dari masing-masing hasil *learning rate* mana yang terbaik untuk menerapkan Algoritme CNN pada studi kasus ini. Pengujian terakhir adalah penentuan *epoch* dan *learning rate* yang terbaik dari pengujian sebelumnya. Dengan pengujian ini diharapkan mendapatkan hasil yang lebih optimal dari nilai akurasi dan lama komputasi pembuatan model CNN dari sebelumnya yang menggunakan *epoch* = 100 dan *learning rate* yang masih belum ditentukan.

C. Hasil pengujian model CNN

Pengujian model CNN dengan menerapkan dengan menggunakan *optimizers* Adam dan untuk penerapan *data loss* menerapkan *categorical_crossentropy*. Pengujian ini dengan menerapkan *epoch* = 100 pada semua *learning rate* yang sesuai dengan skenario pengujian tersebut yang dilakukan perulangan pengujian sebanyak 3 kali [7], [25]–[27]. Detail skenario pengujian model CNN dapat dilihat seperti pada Tabel 4.

TABEL IV
HASIL RATA-RATA PENGUJIAN MODEL CNN DENGAN DATA LOSS CATEGORICAL_CROSSENTROPY, EPOCH = 100

<i>Learning rate</i>	<i>Loss</i>	<i>Accuracy</i>	<i>Val_loss</i>	<i>Val_accuracy</i>
10e-3	0.0224	0.9931	0.1361	0.9682
10e-4	0.0079	0.9974	0.1631	0.9646
10e-5	0.0571	0.9799	0.1360	0.9533
10e-6	0.2795	0.8991	0.3203	0.8782

Tabel 4 menunjukkan bahwa pengujian berdasarkan *learning rate* yang terbaik adalah dengan menggunakan *learning rate* = 10^{-4} . Hal ini dilihat berdasarkan tingginya nilai *accuracy* dan *val_accuracy* secara berturut-turut mencapai nilai 0.9974 dan 0.9646, serta memiliki selisih 0.1552. Sedangkan untuk *loss* sebesar 0.0079 dan *val_loss* sebesar 0.1631 dengan selisih 0.0328. Jika dilihat dari nilai tertinggi tersebut, peneliti juga menguji dengan mengguncari *epoch* mana yang lebih optimal berdasarkan dengan patokan *learning rate* dan jumlah maksimal *epoch* yang digunakan. Hasil pengujian ini dapat dilihat pada Tabel 5.

TABEL V
HASIL RATA-RATA PENGUJIAN MODEL CNN DENGAN MELIHAT EPOCH TERBAIK PADA MASING-MASING LEARNING RATE

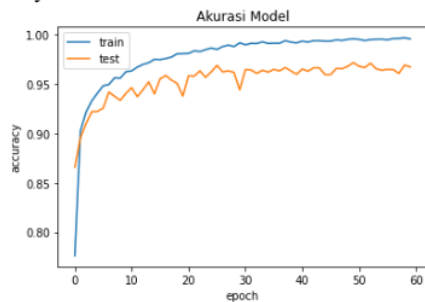
<i>Learning rate</i>	<i>Loss</i>	<i>Accuracy</i>	<i>Val_loss</i>	<i>Val_accuracy</i>	<i>Epoch</i>
10e-3	0.0261	0.9925	0.1382	0.9673	80
10e-4	0.0131	0.9954	0.1248	0.9680	60
10e-5	0.0571	0.9799	0.1359	0.9532	100
10e-6	0.2795	0.8991	0.3203	0.8782	100

Tabel 5 menunjukkan bahwa dengan *epoch* kecil sudah mendapatkan hasil tingkat akurasi dan presisi yang terbaik. *Epoch* yang didapatkan adalah 60 dengan *learning rate* = 10^{-3} atau 10^{-4} . Nilai terbaik ini dinyatakan oleh tingginya *Accuracy* sebesar 0.9954, *val_accuracy* sebesar 0.9680. Bahkan, nilai *loss* dan *val_loss* juga dinyatakan sebagai nilai terkecil dari percobaan ini secara berturut-turut sebesar 0.0131 dan 0.1248. Dapat dibandingkan pula pada Tabel 4 dan Tabel 5 bahwa jika menggunakan *learning rate* yang semakin kecil dan *epoch* semakin tinggi, maka nilai *accuracy* dan *val_accuracy* semakin menurun. Tentunya, nilai *loss* dan *val_loss* juga semakin meningkat yang artinya tingkat akurasi prediksi semakin banyak menunjukkan kesalahan prediksi.

Pada hasil prediksi, akan dibandingkan hasilnya dengan menggunakan *epoch* sebesar 100 dan *learning rate* 10e-4 juga dengan perulangan pengujian sebanyak 3 kali. Penentuan nilai *epoch* dan *learning rate* ini dimaksudkan untuk mengurangi lama komputasi pembuatan model CNN. Jika semakin banyak *epoch* yang digunakan, maka semakin lama komputasinya. Dalam penelitian ini, rata-rata komputasi tiap *epoch* sebesar 355 detik dan banyaknya iterasi pada tiap *epoch* adalah 807 seperti pada Gambar 3.

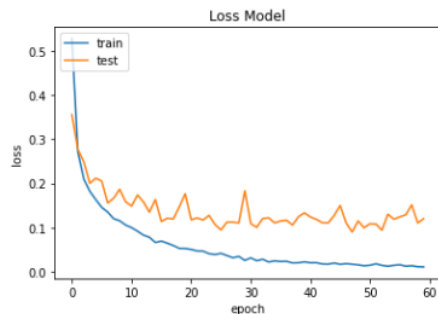
Pengujian model CNN dilakukan dengan membandingkan hasil akurasi berdasarkan nilai *epoch* dan *learning rate*. Berdasarkan Tabel 3 dan Tabel 4, grafik hasil akurasi pemodelan dan *loss* berturut-turut ditampilkan pada Gambar 4 dan Gambar 5.

Gambar 4 menampilkan grafik dari akurasi untuk *data train* dan *data test*. Untuk grafik *data train*, nilai akurasi stabil di sekitar nilai 0.97 hingga 0.99 yang dimulai pada *epoch* 30. Sedangkan untuk grafik *data test* stabil di antara nilai 0.94 hingga 0.97 yang juga dimulai pada *epoch* 30. Hal ini menunjukkan bahwa nilai yang didapatkan semakin mendekati angka 1.00, maka akurasinya akan semakin baik.



Gambar 4. Grafik Hasil Akurasi Pemodelan CNN

Gambar 5 menampilkan grafik dari *data loss* terhadap *data train* dan *data test*. Untuk grafik *data train*, nilai akurasi stabil di sekitar nilai 0.05 hingga 0.01 yang dimulai pada *epoch* 30. Sedangkan untuk grafik *data test* stabil di antara nilai 0.15 hingga 0.1 yang juga dimulai pada *epoch* 30. Hal ini menunjukkan bahwa jika kedua nilai tersebut mendekati angka 0, maka semakin kecil kemungkinan data yang akan diuji terhindar dari kesalahan prediksi.



Gambar 5. Grafik Hasil Loss Pemodelan CNN

D. Hasil prediksi

Hasil prediksi adalah sebagai pengujian data dari sebuah masukkan citra sebanyak 120 citra, dibagi dengan 3 kelas yang berbeda masing-masing sebanyak 40 citra yaitu kelas Cat, Dog, dan Wild. Untuk sampel data yang digunakan dapat dilihat pada Tabel 2. Skenario pengujian ini dilakukan dengan 2 skenario yang berbeda yang juga dilakukan sama dengan proses sebelumnya [7], [25]–[27]. Pertama berdasarkan *epoch* 100 dengan patokan *learning rate* yang digunakan. Untuk hasil pengujian skenario pertama dapat dilihat pada Tabel 6 dan Tabel 7.

TABEL VI
HASIL PERSENTASE PREDIKSI DATA TESTING (EPOCH=100)

Learning rate	Cat (%)	Dog (%)	Wild (%)	Rata-rata (%)
10e-3	73.33	79.17	76.67	76.39
10e-4	85.83	77.50	75.00	79.44
10e-5	73.33	79.17	73.33	75.28
10e-6	67.50	59.17	81.67	69.44

Berdasarkan Tabel 6 dapat diambil kesimpulan bahwa, persentase prediksi yang terbaik adalah menggunakan $learning\ rate = 10e-4$ dengan hasil akurasi rata-rata keseluruhan kategori kelas adalah 79,44%. Tabel 7 menunjukkan bahwa banyak citra yang dapat diprediksi pada $learning\ rate = 10e-4$ sebanyak rata-rata 31,78 citra dari 40 dari tiap kelas. Sedangkan rata-rata total citra keseluruhan adalah 95,33 dari 120 citra.

TABEL VII
HASIL BANYAK RATA-RATA CITRA PREDIKSI DATA TESTING (EPOCH=100)

Learning rate	Cat	Dog	Wild	Rata-rata	Total Citra
10e-3	29.33	31.67	30.67	30.56	91.67
10e-4	34.33	31	30	31.78	95.33
10e-5	29.33	31.67	29.33	30.11	90.33
10e-6	27	23.67	32.67	27.78	83.33

Skenario kedua dibandingkan dengan $epoch$ dari hasil paling optimal dari nilai $accuracy$, $val_accuracy$, $loss$, dan val_loss pada skenario pertama yang menggunakan $epoch = 100$ yaitu dengan menggunakan $epoch = 60$ dan $learning\ rate = 10e-4$. Penentuan patokan ini berdasarkan hasil dari pengujian Tabel 5 dan Tabel 6 [28]. Maka, hasil prediksi dengan menggunakan data Tabel 2 dapat dilihat pada Tabel 8 untuk persentase rata-rata tingkat akurasinya. Sedangkan pada Tabel 9 adalah banyaknya citra yang dapat terprediksi dengan akurat.

TABEL VIII
HASIL PERSENTASE PREDIKSI DARI DATA TESTING (EPOCH = 60 DAN LEARNING RATE = 10E-4)

Pengujian ke	Cat (%)	Dog (%)	Wild (%)	Rata-rata (%)
1	77.50	75.00	77.50	76.67
2	92.50	75.00	70.00	79.17
3	82.50	72.50	77.50	77.50
Rata-rata	84.17	74.17	75.00	77.78

TABEL IX
HASIL BANYAK RATA-RATA CITRA PREDIKSI DATA TESTING (EPOCH=100)

Pengujian ke	Cat	Dog	Wild	Rata-rata	Total
1	31	30	31	30.67	92
2	37	30	28	31.67	95
3	33	29	31	31	93
Rata-rata	33.67	29.67	30	31.11	93.33

Pada Tabel 8 dan Tabel 9 dapat disimpulkan bahwa pada tiap kategori kelas, hasil prediksi yang didapatkan dari $data\ testing$ dari Tabel 2 rata-rata mendapatkan sebanyak 31,11 citra yang sesuai. Jika diakumulasi secara keseluruhan, maka didapatkan rata-rata sebesar 93,33 citra prediksi yang sesuai dari 120 citra dan didapatkan nilai persentase akurasi sebesar 77,78%.

Dengan hasil yang sudah didapatkan pada Tabel 8 dan Tabel 9 meskipun menggunakan $epoch$ lebih rendah daripada Tabel 6 dan Tabel 7, maka peneliti memiliki permasalahan yaitu nilai akurasinya masih kurang dari 95%. Padahal, ketika proses pembuatan model CNN, mendapatkan hasil $accuracy$ dan $val_accuracy$ secara berturut-turut adalah 99,50% dan 96,60% dari Tabel 4 dan Tabel 5.

Berdasarkan hasil prediksi yang sudah didapatkan sebelumnya, peneliti menguji lagi dengan menggunakan dataset yang dipilih secara acak seperti pada Tabel 1 untuk membuktikan dengan hasil yang seharusnya menyesuaikan hasil Tabel 4 dan Tabel 5. Pada pengujian ini juga menggunakan $epoch\ 60$ dan $learning\ rate\ 10-4$. Sehingga hasil yang didapatkan dapat dilihat pada Tabel 10 dan Tabel 11.

TABEL X
HASIL PERSENTASE PREDIKSI DARI DATASET (EPOCH = 60 DAN LEARNING RATE = 10E-4)

Pengujian ke	Cat (%)	Dog (%)	Wild (%)	Rata-rata (%)
1	100.00	97.50	95.00	97.50
2	100.00	97.50	87.50	95.00
3	100.00	100.00	97.50	99.17
Rata-rata	100.00	98.33	93.33	97.22

Tabel 10 dan Tabel 11 menunjukkan bahwa pengujian yang telah dilakukan berhasil dan mendapatkan hasil yang memuaskan. Hasil prediksi ini mendapatkan rata-rata citra dari setiap kelas mendapatkan 38,89 citra yang sesuai. Jika diakumulasi secara keseluruhan, maka mendapatkan 116,33 citra yang sesuai dari 120 citra yang digunakan. Sehingga nilai persentase akurasinya mencapai 97,22%.

TABEL XI
HASIL BANYAK CITRA PREDIKSI DARI DATASET (EPOCH = 60 DAN LEARNING RATE = 10E-4)

Pengujian ke	Cat	Dog	Wild	Rata-rata	Total
1	40	39	38	39	116
2	40	39	35	38	114
3	40	40	39	39.67	119
Rata-rata	40	39.33	37.33	38.89	116.33

24 Dari hasil Tabel 10 dan Tabel 11 juga dapat membuktikan bahwa, nilai akurasi yang didapatkan tersebut juga sesuai dengan hasil pengujian pada Tabel 4 yang menunjukkan bahwa dengan *learning rate* = $10e-4$ dan *epoch* = 60 mendapatkan nilai validasi akurasi sebesar 96.80%. Nilai ini tidak terlalu signifikan perbedaan antara rata-rata akurasi yang didapatkan pada Tabel 9 yaitu 97.22% dengan selisih perbedaan adalah 0.42%. Dengan selisih perbedaan ini dataset yang digunakan tidak mengalami *overfitting* ataupun *underfitting*.

15 IV. KESIMPULAN DAN SARAN

Hasil penelitian ini dapat diambil kesimpulan bahwa, Algoritme CNN dapat melakukan identifikasi terhadap dataset *animal faces* dengan baik. Hasil terbaik yang didapatkan adalah dengan menerapkan *learning rate* = $10e-4$ dan *epoch* = 60 yang menghasilkan rata-rata nilai akurasi paling tinggi sebesar 77.78% dengan 93.33 citra yang teridentifikasi secara tepat dari 120 citra *testing*. Untuk pengujian yang citra dipilih secara acak mendapatkan rata-rata nilai akurasi sebesar 97.22% dengan citra yang teridentifikasi secara tepat sebanyak 116.33 dari 120 citra yang terpilih.

Jika dibandingkan dengan hasil prediksi dengan menggunakan antara citra prediksi yang menggunakan sampel dari dataset dan citra prediksi yang di luar dari dataset, maka hasil akurasinya menurun hingga di bawah 80%. Hal ini disebabkan pada proses prediksinya hanya sekadar melakukan *resize* secara manual dengan mengikuti ukuran piksel yang ditentukan di awal. Seharusnya, pada proses ini juga dilakukan proses *pre-processing* yang lebih kompleks lagi untuk penyesuaian resolusi dan *cropping* gambar pada *data testing* yang sesuai dengan resolusi model CNN yang dipakai. Untuk proses tambahan ini, dapat dengan menggunakan seperti library OpenCV atau menggunakan model arsitektur VGG-16 untuk CNN. Sehingga, diharapkan hasil akurasi dari *data testing* yang dari luar dataset akan lebih meningkat dibandingkan dengan penelitian ini.

DAFTAR PUSTAKA

- [1] N. Shahdadpuri, "Real Image of Computer Vision Application and its Impact : Future and Challenges," *Wesley. J. Res.*, vol. 13, no. 53, pp. 62–75, 2020.
- [2] A. Peryanto, A. Yudhana, and R. Umar, "Klasifikasi Citra Menggunakan Convolutional Neural Network dan K Fold Cross Validation," *J. Appl. Informatics Comput.*, vol. 4, no. 1, pp. 45–51, 2020.
- [3] J. Gross, J. Breitenbach, H. Baumgartl, and R. Buettner, "High-Performance Detection of Corneal Ulceration Using Image Classification with Convolutional Neural Networks," *Proc. 54th Hawaii Int. Conf. Syst. Sci.*, pp. 3416–3425, 2021.
- [4] W. Rahmaniar and A. Hernawan, "Real-Time Human Detection Using Deep Learning on Embedded Platforms : A Review," *J. Robot. Control*, vol. 2, no. 6, pp. 462–468, 2021, doi: 10.18196/jrc.26123.
- [5] M. R. Alwanda, R. Putra, K. Ramadhan, and D. Alamsyah, "Implementasi Metode Convolutional Neural Network Menggunakan Arsitektur LeNet-5 untuk Pengenalan Doodle," *J. Algoritm.*, vol. 1, no. 1, 2020.
- [6] A. S. Riyadi, I. Puspa, and S. Widayati, "KLASIFIKASI CITRA ANJING DAN KUCING MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK (CNN)," in *Seminar Nasional Teknologi Informasi dan Komunikasi*, 2021, vol. 5, pp. 2–6.
- [7] N. Azahro Choirunisa, T. Karlita, and R. Asmara, "Deteksi Ras Kucing Menggunakan Compound Model Scaling Convolutional Neural Network," *Technomedia J.*, vol. 6, no. 2, pp. 236–251, 2021, doi: 10.33050/tmj.v6i2.1704.
- [8] M. R. Effendi, "Sistem Deteksi Wajah Jenis Kucing Dengan Image Classification Menggunakan Opencv," *J. Teknol. Inform. dan Komput.*, vol. 4, no. 1, pp. 27–35, 2018, doi: 10.37012/jtik.v4i1.283.
- [9] M. A. Pangestu and H. Bunyamin, "Analisis Performa dan Pengembangan Sistem Deteksi Ras Anjing pada Gambar dengan Menggunakan Pre-Trained CNN Model," *J. Tek. Inform. dan Sist. Inf.*, vol. 4, pp. 337–344, 2018.

- [10] K. O. Lauw *et al.*, “Identifikasi Jenis Anjing Berdasarkan Gambar Menggunakan Convolutional Neural Network Berbasis Android,” *J. Infra*, vol. 8, no. 2, pp. 37–43, 2020.
- [11] A. Hossain and S. Sajib Alam, “Classification of Image using Convolutional Neural Network (CNN),” *Glob. J. Comput. Sci. Technol. D Neural Artif. Intell.*, vol. 19, no. 2, 2019.
- [12] T. Nurhikmat, “IMPLEMENTASI DEEP LEARNING UNTUK IMAGE CLASSIFICATION MENGGUNAKAN ALGORITMA CONVOLUTIONAL NEURAL NETWORK (CNN) PADA CITRA WAYANG GOLEK,” YOGYAKARTA, 2018.
- [13] D. Jha, A. Yazidi, M. A. Riegler, D. Johansen, and D. Johansen, “LightLayers : Parameter Efficient Dense and Convolutional Layers for Image Classification,” *arXiv Prepr. arXiv2101.02268*, no. ML, pp. 1–12, 2021.
- [14] 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, doi: 10.24895/jig.2018.24-2.810.
- [15] C. K. Dewa and A. L. Fadhilah, “Convolutional Neural Networks for Handwritten Javanese Character Recognition,” *IJCCS (Indonesian J. Comput. Cybern. Syst.*, vol. 12, no. 1, pp. 83–94, 2018, doi: 10.22146/ijccs.31144.
- [16] B. Boehmke and B. Greenwell, *Hands-On Machine Learning with R*. 2019.
- [17] P. Singh and A. Manure, *Learn TensorFlow 2.0*. 2020.
- [18] S. Ilahiyah and A. Nilogiri, “Implementasi Deep Learning Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan Convolutional Neural Network,” *JUSTINDO (Jurnal Sist. Teknol. Inf. Indones.*, vol. 3, pp. 49–56, 2018.
- [19] WULAN ANGGRAINI, “Deep Learning Untuk Deteksi Wajah Yang Berhijab Menggunakan Algoritma Convolutional Neural Network (Cnn) Dengan Tensorflow,” *Deep Learn. Untuk Deteksi Wajah Yang Berhijab Menggunakan Algoritma Convolutional Neural Netw. Dengan Tensorflow*, vol. 28, no. 2, pp. 1–43, 2020, [Online]. Available: <http://repositorio.unan.edu.ni/2986/1/5624.pdf>.
- [20] I. K. M. Jais, A. R. Ismail, and S. Q. Nisa, “Adam Optimization Algorithm for Wide and Deep Neural Network,” *Knowl. Eng. Data Sci.*, vol. 2, no. 1, p. 41, 2019, doi: 10.17977/um018v2i12019p41-46.
- [21] Y. Ho and S. Wookey, “The Real-World-Weight Cross-Entropy Loss Function: Modeling the Costs of Mislabeling,” *IEEE Access*, vol. 8, pp. 4806–4813, 2020, doi: 10.1109/ACCESS.2019.2962617.
- [22] A. Krizhevsky and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” pp. 1–9.
- [23] A. Panja, J. J. Christy, and Q. M. Abdul, “An Approach to Skin Cancer Detection using Keras and Tensorflow,” *J. Phys. Conf. Ser.*, vol. 1911, no. 1, p. 012032, 2021, doi: 10.1088/1742-6596/1911/1/012032.
- [24] A. Koul, S. Ganju, and M. Kasam, *Practical Deep Learning for Cloud, Mobile, and Edge*, vol. 53, no. 9. O’Reilly Media, 2018.
- [25] Y. N. U. R. Fuadah, I. D. Ubaidullah, N. U. R. Ibrahim, F. F. Taliningsing, N. K. Sy, and M. Adnan, “Optimasi Convolutional Neural Network dan K-Fold Cross Validation pada Sistem Klasifikasi Glaukoma,” vol. 10, no. 3, pp. 728–741, 2022.
- [26] R. Agustina, R. Magdalena, and N. O. R. K. Caecar, “Klasifikasi Kanker Kulit menggunakan Metode Convolutional Neural Network dengan Arsitektur VGG-16,” *Elkomika*, vol. 10, no. 2, pp. 446–457, 2022, [Online]. Available: <https://ejurnal.itenas.ac.id/index.php/elkomika/article/view/5674/2879>.
- [27] A. Alamsyah, B. Prasetyo, M. F. Al Hakim, and F. D. Pradana, “Prediction of COVID-19 Using Recurrent Neural Network Model,” *Sci. J. Informatics*, vol. 8, no. 1, pp. 98–103, 2021, doi: 10.15294/sji.v8i1.30070.
- [28] M. A. Abu, N. H. Indra, A. H. A. Rahman, N. A. Sapiee, and I. Ahmad, “A study on image classification based on deep learning and tensorflow,” *Int. J. Eng. Res. Technol.*, vol. 12, no. 4, pp. 563–569, 2019.

juti dimas

ORIGINALITY REPORT

16%

SIMILARITY INDEX

16%

INTERNET SOURCES

8%

PUBLICATIONS

11%

STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Sogang University Student Paper	3%
2	dspace.uii.ac.id Internet Source	2%
3	publikasi.mercubuana.ac.id Internet Source	1%
4	www.it145.com Internet Source	1%
5	123dok.com Internet Source	1%
6	repository.unib.ac.id Internet Source	1%
7	www.coursehero.com Internet Source	1%
8	Submitted to Konsorsium Turnitin Relawan Jurnal Indonesia Student Paper	1%
9	Submitted to Universitas Dian Nuswantoro Student Paper	<1%

10	Submitted to Sriwijaya University Student Paper	<1 %
11	Submitted to University of Sydney Student Paper	<1 %
12	jurnal.unimor.ac.id Internet Source	<1 %
13	www.scribd.com Internet Source	<1 %
14	eprints.akakom.ac.id Internet Source	<1 %
15	docplayer.info Internet Source	<1 %
16	journal.unpar.ac.id Internet Source	<1 %
17	repository.upi.edu Internet Source	<1 %
18	jurnalmahasiswa.stiesia.ac.id Internet Source	<1 %
19	openlibrarypublications.telkomuniversity.ac.id Internet Source	<1 %
20	pels.umsida.ac.id Internet Source	<1 %
21	Yohannes Yohannes, Siska Devella, Kelvin Arianto. "Deteksi Penyakit Malaria	<1 %

Menggunakan Convolutional Neural Network Berbasis Saliency", JUITA: Jurnal Informatika, 2020

Publication

22	docobook.com Internet Source	<1 %
23	fr.scribd.com Internet Source	<1 %
24	mvdw.org Internet Source	<1 %
25	repository.pelitabangsa.ac.id:8080 Internet Source	<1 %
26	sti.vista.gov.vn Internet Source	<1 %
27	text-id.123dok.com Internet Source	<1 %
28	belajarofficeonline.blogspot.com Internet Source	<1 %
29	citec.amikom.ac.id Internet Source	<1 %
30	id.scribd.com Internet Source	<1 %
31	media.neliti.com Internet Source	<1 %

32

repository.upnvj.ac.id

Internet Source

<1 %

33

statecon.rea.ru

Internet Source

<1 %

34

medium.com

Internet Source

<1 %

35

repositori.usu.ac.id

Internet Source

<1 %

Exclude quotes On

Exclude matches Off

Exclude bibliography On