

# Penerapan *Convolutional Neural Network* pada Klasifikasi Jenis Ras Kucing Menggunakan *ResNet50V2*

Cut Agusniar<sup>1</sup>, Della Adelia<sup>2</sup>

Program Studi Teknik Informatika, Fakultas Teknik, Universitas Malikussaleh

<sup>1</sup>[cutagusniar@unimal.ac.id](mailto:cutagusniar@unimal.ac.id), <sup>2</sup>[della.210170051@mhs.unimal.ac.id](mailto:della.210170051@mhs.unimal.ac.id)

**Abstrak**— Kucing adalah salah satu hewan peliharaan yang paling populer di seluruh dunia. Banyak orang memilih untuk memelihara kucing karena berbagai keunggulannya, seperti meredakan stres dan kecemasan. Banyak jenis ras kucing yang diakui secara internasional yaitu 142 ras kucing dan para ahli biologi telah mengklasifikasikan ras-ras ini ke dalam beberapa kategori. Namun dari banyaknya ras yang ada muncul suatu tantangan bagi manusia dalam membedakan setiap ras karena banyak kucing yang memiliki ciri-ciri fisik yang serupa. Hal ini menyebabkan para pemilik kucing tidak mengetahui jenis ras dari kucing yang mereka pelihara sehingga menjadi masalah karena setiap ras kucing membutuhkan perawatan khusus jika terserang penyakit. Oleh karena itu dalam penelitian ini penulis membuat suatu sistem yang dapat mengklasifikasikan jenis ras kucing berdasarkan 12 kelas yaitu *Abyssinia*, *Bengal*, *Bombay*, *Birman*, *British Shorthair*, *Egyptian*, *Maine Coon*, *Persian*, *Ragdoll*, *Russian Blue*, *Siamese*, dan *Sphynx*. Penelitian ini menggunakan dataset sebanyak 2.402 gambar kucing dan menggunakan metode *Convolutional Neural Network* (CNN) dengan arsitektur *ResNet50V2* serta memanfaatkan pengoptimalan *stochastic gradient descent* (SGD). Hasil dari penelitian ini mendapatkan akurasi *training* sebesar 99,17%, ditambah dengan akurasi *validation* sebesar 88,13%. Selain itu, *loss training* dan *loss validation* yang diperoleh masing-masing adalah 0,3760 dan 0,7857. Melalui penggunaan *confusion matrix* dan evaluasi seperti *precision*, *recall*, *f1-score*, dan *support*, menghasilkan akurasi rata-rata sebesar 88%.

**Kata Kunci**— *Convolutional Neural Network*, *Deep Learning*, *Klasifikasi Ras Kucing*, *ResNet50V2*, *Transfer Learning*

## I. PENDAHULUAN

Kucing adalah salah satu hewan peliharaan yang paling populer di seluruh dunia. Banyak orang memilih untuk memelihara kucing karena berbagai keunggulannya, seperti sifatnya yang tenang, mandiri dan tidak terlalu bergantung, kebutuhan makan yang minimal, dan perawatan relatif mudah. Selain itu, kepemilikan kucing dapat memberikan efek yang baik bagi pemiliknya, termasuk meredakan stres dan kecemasan [1].

Kucing (*Felis Catus*) adalah mamalia karnivora karena termasuk ke dalam keluarga *Felidae*. Kucing telah berinteraksi dengan manusia dan telah menyebar secara global sejak 6000 SM [1]. Ada banyak ras kucing, masing-masing memiliki karakteristik fisik yang berbeda. Banyak jenis ras kucing yang diakui secara internasional yaitu 142 ras kucing dan para ahli biologi telah mengklasifikasikan ras-ras ini ke dalam beberapa

kategori, termasuk *Birman*, *Ragdoll*, *Russian*, *British*, dan lainnya [2][3]. Namun, membedakan ras-ras ini dapat menjadikan tantangan, bahkan bagi mata manusia, karena banyak kucing yang memiliki ciri-ciri fisik yang serupa.

Beberapa pemilik kucing mungkin akan tidak mengetahui jenis ras dari kucing yang mereka pelihara, yang dapat menjadi masalah karena setiap ras membutuhkan perawatan khusus, terutama ketika terserang penyakit. Sangat penting untuk mengidentifikasi ras kucing untuk memastikan pemeliharaan dan perawatan yang tepat. Oleh karena itu, diperlukan sistem otomatis yang mampu mengklasifikasikan setiap ras kucing secara akurat untuk memfasilitasi perawatan yang tepat.

Untuk mengidentifikasi ras kucing, dibutuhkan metode klasifikasi gambar yaitu *Convolutional Neural Network* (CNN). Klasifikasi gambar adalah teknik yang digunakan untuk mengategorikan gambar ke dalam kelompok tertentu. *Convolutional Neural Network* adalah jenis algoritma *Deep Learning* yang memproses *input* gambar, mengidentifikasi fitur atau objek yang relevan dalam gambar untuk pembelajaran mesin, dan membedakan berbagai gambar [4][5][6].

Penelitian yang dilakukan oleh Joseph Alberto dan Dedy Hermanto dengan judul *Klasifikasi Jenis Burung Menggunakan Metode CNN Dan Arsitektur ResNet-50*. Klasifikasi spesies burung diuji dengan menggunakan 400 *Bird Species – Classification* data Klasifikasi, yang disaring untuk memasukkan hanya burung-burung yang ditemukan di Indonesia. Hal ini menghasilkan *dataset* yang terdiri dari 63 spesies burung dengan total 9.445 gambar, termasuk 8.185 gambar pelatihan dan 1.260 gambar uji. Klasifikasi dilakukan dengan menggunakan metode *Convolutional Neural Network* (CNN), dengan model yang dibangun menggunakan arsitektur *ResNet50*. Proses pelatihan menggunakan pengoptimal ADAM dan SGD untuk menentukan kinerja optimal, mencapai akurasi 98% dengan mengoptimal SGD setelah 10 epoch [7].

Kemudian penelitian yang dilakukan oleh Umi Kulsum dan Anis Cherid dengan judul *Penerapan Convolutional Neural Network pada Klasifikasi Tanaman Menggunakan Resnet50*. Penelitian ini mengategorikan tanaman apel ke dalam dua klasifikasi, yaitu sehat dan busuk, dengan menggunakan citra digital yang diproses melalui *Convolutional Neural Network* (CNN) dengan model *ResNet50*. *Dataset* pelatihan terdiri dari 1.545 gambar, sedangkan *dataset* pengujian terdiri dari 661 gambar. Proses klasifikasi membedakan antara gambar daun apel yang sehat dan gambar daun apel yang busuk. Evaluasi

model menggunakan *Confusion Matrix* menunjukkan akurasi sebesar 91% setelah 50 kali pelatihan [8].

Untuk mengatasi permasalahan yang ada, penelitian ini bertujuan untuk mengembangkan sistem klasifikasi ras kucing otomatis yang secara akurat mengidentifikasi setiap ras untuk memastikan perawatan yang tepat. Model *ResNet50* khususnya versi 2 akan digunakan untuk *transfer learning*, dengan memanfaatkan pelatihan sebelumnya pada set data yang luas dan beragam. *Transfer learning* melibatkan penerapan pengetahuan yang diperoleh dari model yang telah dilatih sebelumnya ke model baru dengan kasus-kasus yang relevan. Pendekatan ini memungkinkan model untuk dilatih dengan set data yang lebih kecil, sehingga menghemat waktu dan sumber daya biasanya diperlukan untuk pelatihan model awal [9][10].

## II. METODOLOGI PENELITIAN

### A. Pengumpulan Data

*Dataset* untuk penelitian ini di unduh dari *website Kaggle* yaitu "*The Oxford-IIIT Pet Dataset*". Pengumpulan data dilakukan dengan mengunduh data gambar kemudian diproses di dalam folder tersendiri untuk menyertakan gambar kucing saja dengan total 2.402 gambar dari 12 ras kucing. Ras-ras ini termasuk *Abyssinia*, *Bengal*, *Bombay*, *Birman*, *British Shorthair*, *Egyptian*, *Maine Coon*, *Persia*, *Ragdoll*, *Russian Blue*, *Siamese*, dan *Sphynx*. *Dataset* yang telah diolah kemudian di *upload* ke *Google Drive* dan di *load* ke dalam *Goole Colaboratory*.



Gbr. 1 Contoh *dataset* gambar kucing

### B. Pre-processing Data

Dalam penelitian ini, dilakukan tahap *pre-processing* data pada gambar kucing. Tujuan dari *pre-processing* ini adalah untuk meningkatkan kualitas gambar, sehingga memudahkan dan mempercepat kinerja model dalam mengenali ras kucing. pada tahap *pre-processing*, dilakukan beberapa tahapan, antara lain pelabelan, *splitting* data, dan *resize* data [11][12].

```
import glob
import pandas as pd

dir_name = '/content/images/images/'
cats_files = glob.glob(dir_name)
cats_df = pd.DataFrame(cats_files, columns=['full_path'])

cats_df['file_name'] = cats_df['full_path'].str.split('images/images/').str[1]
cats_df['file_name'] = cats_df['file_name'].astype(str)
cats_df['label'] = cats_df['file_name'].str.split('_', n=1).str[0]
classNames = cats_df['label'].unique()

cats_df.head()
```

Gbr. 2 Pelabelan *dataset*

Pada Gbr. 2 dilakukan prosedur pelabelan melibatkan pemberian label pada setiap kelas. Jumlah total label yang dihasilkan sesuai dengan jumlah kelas yang dilakukan dalam penelitian. Untuk penelitian ini, pelabelan dilakukan untuk 12 kelas (*Abyssinia*, *Bengal*, *Bombay*, *Birman*, *British Shorthair*, *Egyptian*, *Maine Coon*, *Persia*, *Ragdoll*, *Russian Blue*, *Siamese*, dan *Sphynx*).

```
datagen = image.ImageDataGenerator(rescale=1/255., validation_split=0.2)

train_generator = datagen.flow_from_dataframe(dataframe=cats_df,
                                             directory= '/content/images/images',
                                             x_col='file_name',
                                             y_col='label',
                                             target_size=(224,224),
                                             class_mode='categorical',
                                             batch_size=64,
                                             subset='training')

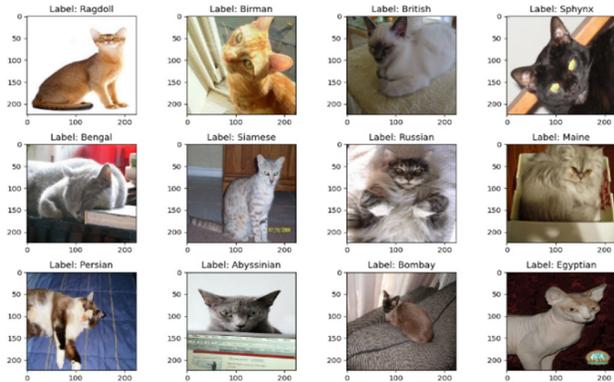
validation_generator = datagen.flow_from_dataframe(dataframe=cats_df,
                                                  directory= '/content/images/images',
                                                  x_col='file_name',
                                                  y_col='label',
                                                  target_size=(224,224),
                                                  class_mode='categorical',
                                                  batch_size=10,
                                                  subset='validation',
                                                  seed=7)
```

Gbr. 3 Proses *rescale*, *split* data, dan *resize*

Pada Gbr. 3 dilakukan *rescaling* atau normalisasi data dengan menyesuaikan nilai yang digunakan untuk perbandingan data sebelum data tambahan dimasukkan. Misalnya, gambar berwarna atau RGB (*Red*, *Green*, *Blue*) memiliki rentang nilai 0-255, yang berdampak pada data saat melakukan pelatihan. Untuk mengatasi masalah tersebut, nilai dapat diskalakan dengan faktor 1/255, sehingga menghasilkan rentang nilai 0-1.

Selanjutnya, *dataset* akan dibagi secara acak menjadi dua bagian yaitu 80% untuk data *training* dan 20% untuk data *validation*. Data *training*, yang terdiri dari 1.920 gambar kucing di 12 kelas, akan digunakan untuk proses pelatihan jenis kucing. Data *validation*, terdiri dari 480 gambar, akan digunakan selama proses pelatihan untuk menilai akurasi model[13].

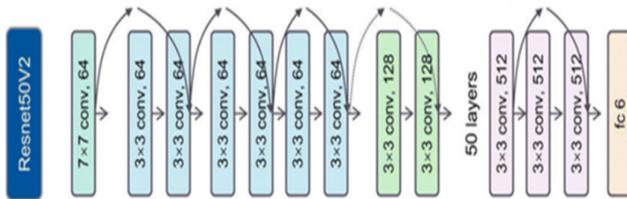
Kemudian, dilakukan *resize* pada *dataset* agar memiliki ukuran yang seragam. Proses ini menyesuaikan data *input* ke sistem, mencegah kesalahan karena ketidakseimbangan ukuran data [13]. Dalam penelitian ini, gambar akan di ubah ukurannya menjadi 224x224 *pixel*.



Gbr. 4 Gambar kucing setelah dilakukan *rescale*, *split* data, dan *resize*

### C. ResNet50

Dalam penelitian ini, model arsitektur yang digunakan adalah *ResNet* khususnya menggunakan arsitektur *ResNet50* versi 2. *ResNet50V2* adalah model arsitektur *Convolutional Neural Network* (CNN) yang merupakan versi perbaikan dari model *ResNet50* [14]. Model *ResNet50* dikembangkan oleh *Microsoft Research* pada tahun 2015. Model ini merupakan bagian dari model *ResNet* yang lebih besar dan menempati posisi tengah, yang terdiri dari 50 lapisan (*depth*). Dibandingkan dengan arsitektur *ResNet* yang lebih umum, *ResNet50* menawarkan keuntungan karena relatif lebih ringan dalam hal biaya komputasi jika disandingkan dengan *ResNet101* atau *ResNet152*, namun tetap memiliki kedalaman yang lebih besar daripada model *ResNet18* dan *ResNet34* yang lebih rendah. Keseimbangan antara kedalaman dan biaya komputasi ini menjadikan *ResNet50V2* sebagai pilihan yang tepat dalam penelitian ini.



Gbr. 5 Arsitektur *ResNet50V2*

Arsitektur *ResNet50V2* terdiri dari lapisan konvolusi awal, yang kemudian diikuti oleh serangkaian blok residu. Masing-masing blok residu ini terdiri dari beberapa lapisan konvolusi dengan *shortcut connection* yang memungkinkan penyebaran informasi di seluruh lapisan. Setelah blok residu terakhir terdapat lapisan *pooling global*, yang berfungsi untuk mengurangi dimensi spasial dari fitur yang dihasilkan. Fitur-fitur ini kemudian disebar ke lapisan *fully connected*, yang pada akhirnya menghasilkan *output* klasifikasi, yang mewakili puncak dari proses arsitektur [15].

Model: "model"

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 224, 224, 3)]	0
keras_layer (KerasLayer)	(None, 1001)	25615849
dense (Dense)	(None, 256)	256512
dropout (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 128)	32896
dropout_1 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 12)	1548

=====  
Total params: 25906805 (98.83 MB)  
Trainable params: 25861365 (98.65 MB)  
Non-trainable params: 45440 (177.50 KB)

Gbr. 6 Arsitektur model dalam penelitian

Dalam penelitian ini, arsitektur *ResNet50V2* ditambahkan dengan beberapa lapisan tambahan. Pertama, lapisan *fully-connected* (*Dense*) dengan 256 unit disertai dengan fungsi aktivasi ReLU (*Rectified Linear Unit*). Selanjutnya, lapisan *Dropout* dengan tingkat 0,3 (30%) diterapkan untuk mengurangi risiko *overfitting*. Setelah itu, lapisan *Dense* lainnya diimplementasikan, yang terdiri dari 128 unit dan fungsi aktivasi ReLU. Untuk mengurangi potensi *overfitting* ditambahkan lapisan *Dropout* 0,3. Kemudian pada lapisan *output* dari model yang merupakan lapisan *Dense* dengan 12 unit, menggunakan fungsi aktivasi *softmax*, yang mengindikasikan bahwa model ditujukan untuk klasifikasi 12 kelas yang berbeda.

### III. HASIL DAN PEMBAHASAN

Pada penelitian ini memanfaatkan metode *Convolutional Neural Network* (CNN), dengan menggunakan model arsitektur *ResNet50V2*. Model yang dihasilkan menunjukkan kemampuan untuk mengklasifikasikan gambar di 12 kelas ras kucing yang berbeda, yaitu *Abyssinia*, *Bengal*, *Bombay*, *Birman*, *British Shorthair*, *Egyptian*, *Maine Coon*, *Persia*, *Ragdoll*, *Russian Blue*, *Siamese*, dan *Sphynx*.

```
lr_schedule = tf.keras.optimizers.schedules.ExponentialDecay(initial_learning_rate=1e-2,
                                                             decay_steps=10000,
                                                             decay_rate=0.9)

optimizer = tf.keras.optimizers.SGD(learning_rate=lr_schedule)
model.compile(loss = 'categorical_crossentropy',
              optimizer = optimizer,
              metrics = ['accuracy'])
```

Gbr. 7 Model *compile*

```

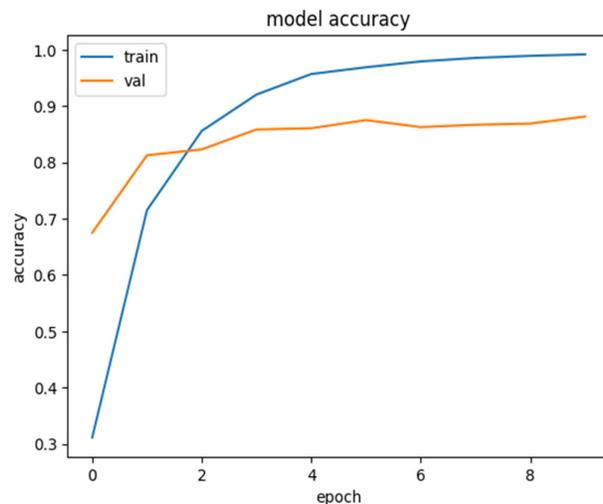
model.summary()

history = model.fit(
    train_generator,
    validation_data= validation_generator,
    epochs=10,
    callbacks=[tensorboard_callback]
)
    
```

Gbr. 8 Model train

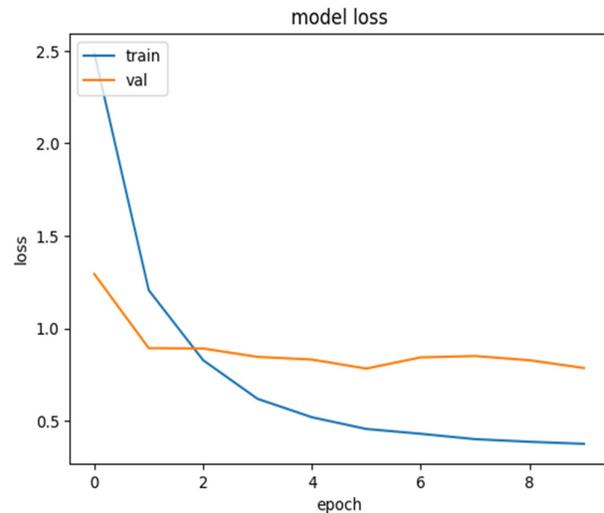
Pada Gbr. 7 dan Gbr. 8 merupakan proses pelatihan model untuk mengenali dan mempelajari pola yang ada di dalam gambar. Proses pelatihan ini dilakukan selama sejumlah *epoch*, dalam penelitian ini menggunakan 10 *epoch*. Evaluasi kinerja model dilakukan pada data *training* dan data *validation*. Proses ini dilakukan dengan *dataset* yang telah dipisah menjadi 80% untuk data *training* dan 20% untuk data *validation*. Model yang telah dilatih kemudian ditugaskan untuk memprediksi label kelas untuk gambar-gambar yang berasal dari *dataset validation*. Label yang diprediksi ini disandingkan dengan label *ground truth*, sehingga memudahkan penghitungan metrik evaluasi seperti *precision*, *recall*, *f1-score*, dan *support*, yang berfungsi sebagai pengukur kinerja model secara keseluruhan.

A. Akurasi



Gbr. 9 Grafik training accuracy ResNet50V2

Pada Gbr. 9 mengilustrasikan kinerja model selama proses pelatihan pada *dataset* yang dilatih. Pada awal *epoch* pertama, model menunjukkan akurasi sebesar 31.15% dan 67.50% pada data *training* dan data *validation*. Setelah menyelesaikan *epoch* ke-10, model mencapai akurasi 99.17% pada data *training*, sementara kinerja pada data *validation* mencapai 88.13%.



Gbr. 10 Grafik training loss ResNet50V2

Pada Gbr. 10 mengilustrasikan kinerja model selama proses pelatihan pada *dataset* yang dilatih. Pada awal *epoch* pertama, model menunjukkan *loss* sebesar 2,4871 dan 1,2936 pada data *training* dan data *validation*. Setelah menyelesaikan *epoch* ke-10, model mencapai akurasi 0.3760 pada data *training*, sementara kinerja pada data *validation* mencapai 0.7857.

B. Precision Recall

*Precision Recall* menunjukkan hasil evaluasi kinerja model klasifikasi menggunakan arsitektur *ResNet50V2* untuk mengklasifikasikan berbagai jenis ras kucing. evaluasi didasarkan pada beberapa metrik, yaitu *precision*, *recall*, *f1-score*, dan *support*.

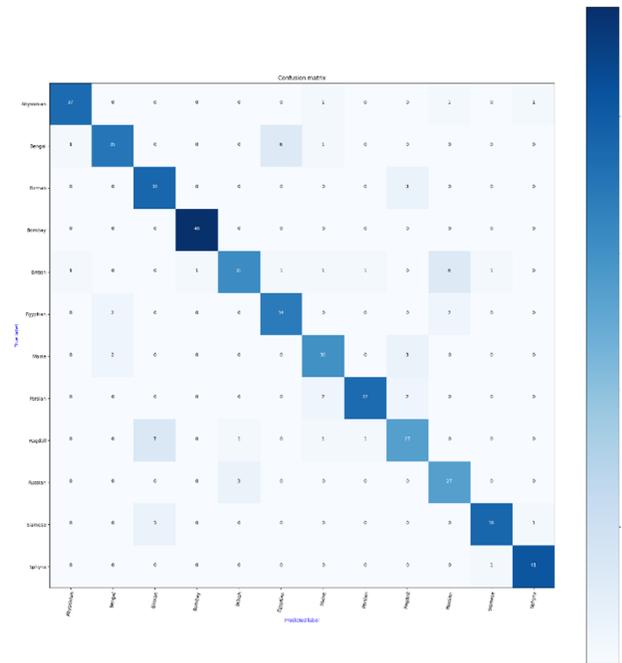
	precision	recall	f1-score	support
Abyssinian	0.95	0.93	0.94	40
Bengal	0.90	0.81	0.85	43
Birman	0.79	0.93	0.85	41
Bombay	0.98	1.00	0.99	48
British	0.89	0.72	0.79	43
Egyptian	0.83	0.89	0.86	38
Maine	0.83	0.86	0.85	35
Persian	0.95	0.90	0.92	41
Ragdoll	0.77	0.73	0.75	37
Russian	0.75	0.90	0.82	30
Siamese	0.95	0.90	0.93	42
Sphynx	0.95	0.98	0.96	42
accuracy			0.88	480
macro avg	0.88	0.88	0.88	480
weighted avg	0.88	0.88	0.88	480

Gbr. 11 Precision recall

Pada Gbr. 11 di atas, tingkat *precision* yang mengindikasikan akurasi model untuk setiap kelas sebagai berikut, *Abyssinian* 95%, *Bengal* 90%, *Birman* 79%, *Bombay* 98%, *British Shorthair* 89%, *Egyptian* 83%, *Maine Coon* 83%, *Persian* 95%,

Ragdoll 77%, Russian Blue 75%, Siam 95%, dan Sphynx 95%. Mengenai *recall*, yang memprediksi tingkat keberhasilan model dalam mengidentifikasi contoh positif, hasilnya adalah Abyssinian 93%, Bengal 81%, Birman 93%, Bombay 100%, British Shorthair 72%, Egyptian 89%, Maine Coon 86%, Persian 90%, Ragdoll 73%, Russian Blue 90%, Siamese 90%, dan Sphynx 98%. Setiap kelas menunjukkan nilai *precision*, *recall*, *f1-score*, dan *support* yang relatif tinggi, yang menghasilkan akurasi keseluruhan sekitar 88%.

C. Evaluasi dan Prediksi dari Seluruh Kelas



Gbr. 12 Confusion matrix

Pada Gbr. 12 memperlihatkan hasil prediksi yang diperoleh dengan menggunakan model *ResNet50V2* dari total 480 data *validation* gambar kucing yang mencakup 12 kelas. Model tersebut menunjukkan hasil yang baik untuk klasifikasi yang akurat di seluruh *dataset* yang beragam. Secara khusus, hasil prediksi menunjukkan hasil sebagai berikut:

- 1) Untuk kelas "*Abyssinian*", yang terdiri dari total 40 gambar, model berhasil mengklasifikasikan 37 contoh dengan benar.
- 2) Untuk kelas "*Bengal*", yang terdiri dari 43 gambar, model berhasil mengklasifikasikan 35 contoh dengan benar.
- 3) Untuk kelas "*Birman*", yang mencakup 41 gambar, model berhasil mengklasifikasikan 38 contoh dengan benar.
- 4) Untuk kelas "*Bombay*", dari total 48 gambar, model berhasil mengklasifikasikan 48 contoh dengan benar.
- 5) Pada kelas "*British Shorthair*", yang terdiri dari 43 gambar, model berhasil mengklasifikasikan 31 contoh dengan benar.
- 6) Untuk kelas "*Egyptian*", dari total 38 gambar, model berhasil mengklasifikasikan 34 contoh dengan benar.

- 7) Untuk kelas "*Maine Coon*", yang terdiri dari 35 gambar, model berhasil mengklasifikasikan 30 contoh dengan benar.
- 8) Untuk kelas "*Persian*", yang terdiri dari 41 gambar, model berhasil mengklasifikasikan 37 contoh dengan benar.
- 9) Untuk kelas "*Ragdoll*", yang terdiri dari 37 gambar, model berhasil mengklasifikasikan 27 contoh dengan benar.
- 10) Untuk kelas "*Russian Blue*", dari total 30 gambar, model berhasil mengklasifikasikan 27 contoh dengan benar.
- 11) Pada kelas "*Siamese*", yang terdiri dari 42 gambar, model berhasil mengklasifikasikan 38 contoh dengan benar.
- 12) Untuk kelas "*Sphynx*", yang terdiri dari 42 gambar, model berhasil mengklasifikasikan 41 contoh dengan benar.

D. Hasil Klasifikasi

Prediksi dilakukan menggunakan gambar yang tidak terdapat di dalam *dataset* yang digunakan. Pada hasil Klasifikasi pada Gbr. 13, kucing pertama teridentifikasi sebagai ras *Persian*. Sedangkan kucing kedua teridentifikasi sebagai ras *Sphynx*. Prediksi ini menunjukkan bahwa model *ResNet50V2* yang digunakan mampu untuk mengklasifikasikan jenis ras kucing dengan akurat.



Gbr. 13 Hasil Klasifikasi Citra Kucing

IV. KESIMPULAN

Penelitian yang dilakukan untuk mengklasifikasikan ras kucing dengan menggunakan *Convolutional Neural Network* (CNN) telah menunjukkan keefektifan model arsitektur *ResNet50* dan memanfaatkan pengoptimalan *stochastic gradient descent* (SGD), berhasil diterapkan untuk mengklasifikasikan 12 ras kucing yang berbeda. Model ini mencapai akurasi *training* sebesar 99,17%, ditambah dengan akurasi *validation* sebesar 88,13%. Selain itu, *loss training* dan *loss validation* yang diperoleh masing-masing adalah 0,3760 dan 0,7857. Melalui penggunaan *confusion matrix* dan evaluasi seperti *precision*, *recall*, *f1-score*, dan *support*, menghasilkan akurasi rata-rata sebesar 88%.

## REFERENSI

- [1] N. A. Choirunisa, T. Karlita, and R. Asmara, "Deteksi Ras Kucing Menggunakan Compound Model Scaling Convolutional Neural Network," *Technomedia Journal*, vol. 6, no. 2 Februari, pp. 236–251, 2022.
- [2] A. T. Ramadhan and A. Setiawan, "Catbreedsnet: An Android Application for Cat Breed Classification Using Convolutional Neural Networks," *Jurnal Online Informatika*, vol. 8, no. 1, pp. 52–60, 2023.
- [3] K. D. Linda, K. Kusriani, and A. D. Hartanto, "Studi Literatur Mengenai Klasifikasi Citra Kucing Dengan Menggunakan Deep Learning: Convolutional Neural Network (CNN)," *Journal of Electrical Engineering and Computer (JEECOM)*, vol. 6, no. 1, pp. 129–137, 2024.
- [4] O. Kembuan, G. C. Rorimpandey, and S. M. T. Tengker, "Convolutional neural network (CNN) for image classification of indonesia sign language using tensorflow," in *2020 2nd International Conference on Cybernetics and Intelligent System (ICORIS)*, IEEE, 2020, pp. 1–5.
- [5] D. Iswanto and D. H. UN, "Klasifikasi Penyakit Tanaman Jagung Menggunakan Metode Convolutional Neural Network (CNN)," *Jurnal Ilmiah Universitas Batanghari Jambi*, vol. 22, no. 2, pp. 900–905, 2022.
- [6] A. M. Tama and R. C. N. Santi, "Klasifikasi Jenis Tanaman Hias Menggunakan Metode Convolutional Neural Network (CNN)," *INTECOMS: Journal of Information Technology and Computer Science*, vol. 6, no. 2, pp. 764–770, 2023.
- [7] J. Alberto and D. Hermanto, "Klasifikasi Jenis Burung Menggunakan Metode CNN Dan Arsitektur ResNet-50," *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*, vol. 10, no. 3, pp. 34–46, 2023.
- [8] U. Kulsum and A. Cherid, "Penerapan Convolutional Neural Network Pada Klasifikasi Tanaman Menggunakan ResNet50," *Jurnal Sistem Informasi dan Sistem Komputer*, vol. 8, no. 2, pp. 221–228, 2023.
- [9] D. S. Dewantara, R. Hidayat, H. Susanto, and A. M. Arymurthy, "CNN with multi stage image data augmentation methods for Indonesia rare and protected orchids classification," in *2020 International Conference on Computer Science and Its Application in Agriculture (ICOSICA)*, IEEE, 2020, pp. 1–5.
- [10] M. A. Gómez-Guzmán *et al.*, "Classifying brain tumors on magnetic resonance imaging by using convolutional neural networks," *Electronics (Basel)*, vol. 12, no. 4, p. 955, 2023.
- [11] N. D. Miranda, L. Novamizanti, and S. Rizal, "Convolutional Neural Network pada klasifikasi sidik jari menggunakan RESNET-50," *Jurnal Teknik Informatika (Jutif)*, vol. 1, no. 2, pp. 61–68, 2020.
- [12] C. E. Nainggolan, M. Nasir, and D. Udariansyah, "Perbandingan Klasifikasi Jenis Sampah Menggunakan Convolutional Neural Network Dengan Arsitektur ResNet18 dan ResNet50," *Computer Science Research and Its Development Journal*, vol. 16, no. 1, pp. 76–90, 2024.
- [13] L. Trihardianingsih, A. Sunyoto, and T. Hidayat, "Classification of Tea Leaf Diseases Based on ResNet-50 and Inception V3," *Sinkron: jurnal dan penelitian teknik informatika*, vol. 8, no. 3, pp. 1564–1573, 2023.
- [14] J. S. Iskandar and R. P. Kristianto, "Pengenalan dan Klasifikasi Ragam Kue Indonesia menggunakan Arsitektur ResNet50V2 pada Convolutional Neural Network (CNN)," in *Prosiding Seminar Nasional Amikom Surakarta*, 2023, pp. 81–92.
- [15] R. P. Astuti *et al.*, "Vegetation classification algorithm using convolutional neural network ResNet50 for vegetation mapping in Bandung district area," *JURNAL INFOTEL*, vol. 14, no. 2, pp. 146–153, 2022.