



## Implementasi Arsitektur CNN *Inception V3* untuk Identifikasi Spesies Burung Endemik di Indonesia

Kahfi Zairan Maulana<sup>1</sup>, Agus Susanto<sup>2</sup>

<sup>1,2</sup> Program Studi Informatika, Fakultas Teknik, Universitas Bengkulu

<sup>1</sup> kahfizairan12@gmail.com <sup>2</sup>agus.susanto@unib.ac.id\*

### Abstract

*The conservation of endemic bird species in Indonesia is a priority, as the country is home to numerous bird species that cannot be found elsewhere. This study implements the Inception V3 Convolutional Neural Network (CNN) architecture to identify endemic bird species in Indonesia. The data used was sourced from Kaggle, comprising 4,390 images of 10 bird species. The study employs transfer learning techniques to train the model on a limited specific dataset. After fine-tuning, the model was tested on eight different test data sets. The results demonstrate that the model achieved a classification accuracy of 96%, as evidenced by the confusion matrix and classification report. This indicates that the Inception V3 model is effective in identifying endemic bird species and can contribute significantly to conservation efforts.*

**Keywords:** *Convolutional Neural Network, Inception V3, Bird Species Identification, Transfer Learning, Conservation.*

### Abstrak

Konservasi burung endemik di Indonesia menjadi prioritas mengingat Indonesia adalah habitat bagi banyak spesies burung yang tidak ditemukan di tempat lain. Penelitian ini menerapkan arsitektur Convolutional Neural Network (CNN) Inception V3 untuk mengidentifikasi spesies burung endemik Indonesia. Data yang digunakan berasal dari situs Kaggle, mencakup 4.390 gambar dari 10 spesies burung. Penelitian ini menggunakan teknik transfer learning untuk melatih model dengan dataset spesifik yang terbatas. Setelah melalui proses *fine-tuning*, model diuji menggunakan delapan set data uji yang berbeda. Hasilnya menunjukkan bahwa model mampu mencapai akurasi klasifikasi sebesar 96%, yang ditunjukkan melalui confusion matrix dan laporan klasifikasi. Ini menunjukkan bahwa model *Inception V3* efektif dalam identifikasi spesies burung endemik dan dapat berkontribusi pada upaya konservasi.

**Kata kunci:** *Convolutional Neural Network, Inception V3, Identifikasi Spesies Burung, Transfer Learning, Konservasi.*

© 2024 Jurnal Pustaka Robot Sister

### 1. Pendahuluan

Konservasi burung endemik di Indonesia merupakan hal penting mengingat Indonesia merupakan rumah bagi sejumlah spesies burung yang tidak ditemukan di tempat lain di dunia. Dengan lebih dari 1.771 spesies burung, di antaranya 513 spesies merupakan jenis endemik. Indonesia memiliki tanggung jawab besar untuk melindungi keanekaragaman hayati ini, terutama mengingat bahwa banyak dari spesies

tersebut terancam punah akibat berbagai faktor, termasuk kerusakan habitat, perburuan, dan perubahan iklim [1].

Peran teknologi dalam identifikasi spesies burung semakin penting seiring dengan meningkatnya kebutuhan untuk melestarikan keanekaragaman hayati dan memahami ekosistem. Teknologi pemrosesan citra, seperti algoritma *Convolutional Neural Network* (CNN), telah digunakan untuk

mengklasifikasikan spesies burung berdasarkan gambar. Penelitian yang dilakukan [2] menunjukkan bahwa metode ini efektif dalam mengidentifikasi berbagai spesies burung dari famili Accipitridae, dengan menggunakan dataset gambar yang cukup besar.

Dalam penelitian [3] diketahui bahwa CNN dapat digunakan untuk mengklasifikasikan spesies burung yang terancam punah. Mereka menemukan bahwa model CNN yang diterapkan dalam aplikasi ini mampu memberikan hasil yang baik dalam pengenalan dan klasifikasi spesies burung berdasarkan gambar. Hal ini menunjukkan potensi besar CNN dalam membantu upaya konservasi dengan memberikan alat yang efisien untuk identifikasi spesies.

Selain itu, CNN dapat digunakan untuk pengenalan citra dalam berbagai kondisi, termasuk pengenalan spesies hewan, meskipun penelitian ini lebih fokus pada pengenalan emosi dalam sinyal suara dan tidak secara langsung berkaitan dengan pengenalan spesies burung [4]. Dengan menggunakan teknik *transfer learning*, CNN dapat dilatih dengan dataset yang lebih kecil, yang sangat berguna dalam konteks pengenalan spesies burung yang mungkin memiliki jumlah data terbatas.

Sementara itu, dalam penelitian lain, disebutkan bahwa CNN merupakan salah satu metode paling populer dalam pengenalan citra, termasuk pengenalan spesies hewan [5]. Mereka mencatat bahwa CNN telah digunakan secara luas dalam berbagai aplikasi, termasuk pengenalan spesies ikan, yang menunjukkan fleksibilitas dan kemampuan adaptasi metode ini dalam konteks pengenalan spesies.

*Inception V3* adalah salah satu arsitektur *Convolutional Neural Network* (CNN) yang terkenal karena efisiensinya dalam pengenalan citra, termasuk dalam aplikasi pengenalan spesies burung. Dalam konteks pengenalan spesies burung, *Inception V3* telah digunakan dalam berbagai studi untuk meningkatkan akurasi klasifikasi. Misalnya, penelitian oleh [6] menunjukkan bahwa *Inception V3*, ketika diterapkan dalam pengenalan citra burung, menunjukkan kinerja yang baik dibandingkan dengan arsitektur lain seperti *ResNet152V2* dan *DenseNet201*.

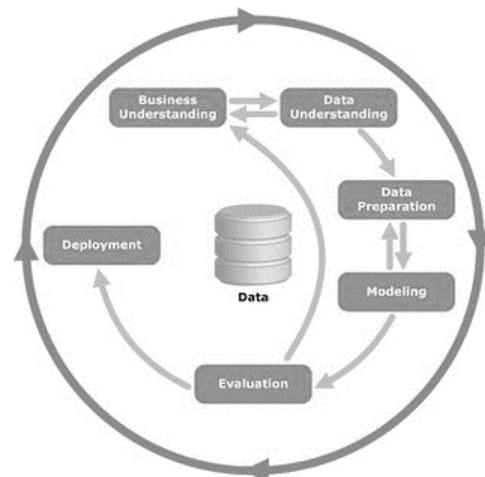
Selain itu, *Inception V3* juga telah diterapkan dalam pengenalan penyakit pada tanaman, yang menunjukkan fleksibilitas arsitektur ini dalam berbagai domain. Dalam penelitian [7], model berbasis *Inception V3* digunakan untuk mendeteksi penyakit pada daun tanaman, yang menunjukkan bahwa arsitektur ini dapat diadaptasi untuk tugas-tugas lain di luar pengenalan spesies burung. Hal ini menunjukkan bahwa teknik *transfer learning* yang digunakan dalam *Inception V3* dapat diterapkan untuk meningkatkan kinerja model dalam konteks yang berbeda.

Keunggulan lain dari *Inception V3* adalah kemampuannya untuk melakukan *transfer learning*, yang memungkinkan model untuk memanfaatkan pengetahuan dari dataset besar yang telah dilatih sebelumnya. Ini sangat berguna dalam konteks pengenalan spesies burung, di mana dataset mungkin tidak selalu besar. Penelitian oleh [8] menunjukkan bahwa penggunaan *transfer learning* dengan *Inception V3* dapat meningkatkan akurasi pengenalan burung secara signifikan.

Secara keseluruhan, penerapan arsitektur CNN *Inception V3* untuk identifikasi spesies burung endemik di Indonesia tidak hanya berpotensi meningkatkan akurasi klasifikasi, tetapi juga dapat berkontribusi pada upaya konservasi dan pemahaman lebih dalam tentang biodiversitas di wilayah tersebut.

## 2. Metode Penelitian

Metode penelitian yang digunakan dalam penelitian ini mengadopsi konsep *Cross-Industry Standard Process for Data Mining* (CRISP-DM). Metodologi ini terdiri dari enam tahapan yaitu *Business Understanding*, *Data Understanding*, *Data Preparation*, *Modelling*, *Evaluation*, dan *Deployment*. Namun, dalam penelitian ini tidak menyertakan tahap *deployment*. Ilustrasi dari metode penelitian ini disajikan pada Gambar 1.



Gambar 1. Metode CRISP-DM

### 2.1. Business Understanding

Pada tahap ini dilakukan pemahaman kebutuhan serta tujuan dari sudut pandang bisnis dan mengartikan pengetahuan ke dalam bentuk pendefinisian masalah pada data mining dan kemudian menentukan rencana serta strategi untuk mencapai tujuan.

### 2.2. Data Understanding

Tahap *Data Understanding* diawali dengan mengumpulkan data, mendeskripsikan data, serta mengevaluasi kualitas data. Data yang digunakan dalam penelitian ini berasal dari situs Kaggle. Dataset yang digunakan adalah berupa gambar burung dengan 10 kelas, yaitu Burung Sempur Hujan Darat,

Sempidan Kalimantan, Maleo, Kuau Kerdil Kalimantan, Junai Emas, Jalak Bali, Cica Daun Kalimantan, Pencuk Ular Asia, Tiong Batu Kalimantan, dan Cenderawasih Botak. Data yang digunakan yaitu sebanyak 4390.

### 2.3. Data Preparation

Pada tahap ini, dilakukan persiapan data untuk digunakan dalam proses pelatihan model. Inti dari tahap ini adalah *preprocessing data*, yang mencakup dua teknik utama yaitu

- Split Data* – dataset dibagi menjadi dua bagian, yaitu data latih dan data validasi.
- Augmentasi Data* – teknik ini digunakan untuk memperkaya data latih dengan menerapkan transformasi seperti rotasi gambar, pergeseran horizontal atau vertikal, *zoom*, serta penyesuaian kecerahan.

### 2.4. Modeling

Pada tahapan ini, dilakukan pemodelan CNN menggunakan *transfer learning Inception*. Proses modeling dalam penelitian ini dilakukan dengan pendekatan berikut:

- Pemilihan Model* – dilakukan pemilihan metode pemodelan yang tepat, yaitu CNN dengan *transfer learning* menggunakan model Inception V3.
- Fine-Tuning* – Model *Inception V3* yang telah dilatih sebelumnya pada dataset ImageNet di-*fine-tuning* ulang menggunakan dataset spesifik penelitian ini.
- Pengaturan Parameter* – Beberapa parameter penting seperti *learning rate*, *epoch*, *batch size*, *iterations*, dan *loss function* diatur selama proses pelatihan model.
- Pelatihan Model* – Data latih digunakan untuk melatih model, sementara akurasi dan *loss* dihitung selama pelatihan untuk memantau perkembangan model.

### 2.5. Evaluation

Pada tahap ini, dilakukan evaluasi terhadap model yang dihasilkan dengan menggunakan *confusion matrix* yang didasarkan pada persebaran kesalahan prediksi gambar, *precision*, *recall*, dan *f1-score*. *Confusion matrix* ini dapat menjadi matriks yang sesuai untuk menilai model yang telah dihasilkan. Model yang dihasilkan kemudian digunakan untuk klasifikasi spesies burung endemik Indonesia, dengan hasil akhirnya dievaluasi berdasarkan *probability score* dari prediksi jenis burung

## 3. Hasil dan Pembahasan

### 3.1. Business Understanding

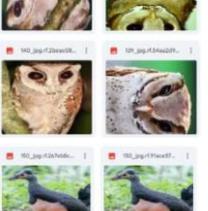
Konservasi burung di Indonesia masih sering mengalami kesulitan dalam proses mengenali spesies burung yang endemik pada wilayah tertentu. Untuk

mengatasi masalah ini, teknologi seperti *Convolutional Neural Network (CNN)* digunakan dalam penelitian untuk klasifikasi spesies burung, dengan memanfaatkan teknik *transfer learning*. Penelitian ini bertujuan untuk mempermudah identifikasi burung endemik di Indonesia tanpa mengganggu habitatnya, dengan memanfaatkan model yang telah dilatih untuk mengklasifikasikan gambar menggunakan data minimal.

### 3.2. Data Understanding

Dari hasil pemahaman data yang dilakukan dalam penelitian ini, maka diperoleh data citra 10 kelas dengan contoh sampel ditunjukkan pada Tabel 1.

Tabel 1. Contoh Sampel dari 10 Kelas Data

No	Kelas	Contoh Sampel
1	Burung Tiong Batu Kalimantan	
2	Burung Sempur Hujan Darat	
3	Burung Sempidan Kalimantan	
4	Burung Pencuk Ular Asia	
5	Burung Maleo	
6	Burung Kuau Kerdil Kalimantan	

No	Kelas	Contoh Sampel
7	Burung Junai Emas	
8	Burung Jalak Bali	
9	Burung Cica Daun Hijau	
10	Burung Cendrawasih Botak	

### 3.3. Data Preparation

Awal dari tahap *data preparation* ini adalah menentukan distribusi data sehingga memungkinkan evaluasi keseimbangan data antar kelas. Selanjutnya dilakukan *preprocessing data* yang mencakup dua teknik utama, yaitu *split data*, dan augmentasi data.

*Split data* digunakan untuk membagi *dataset* menjadi dua bagian, yaitu data latih dan data validasi. Pembagian dilakukan dengan rasio 80-20, di mana 80% data digunakan untuk pelatihan dan 20% untuk validasi. Data latih diacak (*shuffling*) untuk memastikan kerandoman dan reproduktibilitas melalui penggunaan parameter *random\_state*. Berdasarkan jumlah data keseluruhan yang berjumlah 4.390, maka distribusi data latih dan data validasi ditunjukkan pada Tabel 2.

Tabel 2. Pembagian Data Latih dan Data Validasi

No	Data	Jumlah
1	Data Latih (80%)	3.512
2	Data Validasi (20%)	868
	Jumlah	4.390

Implementasi dari *split data* ini dalam pemrograman dilakukan menggunakan kode program sebagai berikut.

```
# Import necessary functions from sklearn
from sklearn.utils import shuffle
from sklearn.model_selection import train_test_split

# Shuffle the dataframe to ensure random distribution of data
image_df = shuffle(image_df, random_state=1)

# Separate the data into training and validation sets
# train_test_split splits the data into two sets: 80% for training and 20% for validation
train_df, validasi_df = train_test_split(image_df, train_size=0.8, shuffle=True, random_state=1)
```

*Augmentasi Data* digunakan untuk memperkaya data latih dengan menerapkan transformasi seperti rotasi gambar, pergeseran horizontal atau vertikal, *zoom*, serta penyesuaian kecerahan. Augmentasi ini membantu model untuk lebih baik dalam melakukan generalisasi dengan menciptakan variasi pada data input tanpa harus mengumpulkan gambar baru. Untuk data validasi, hanya dilakukan *rescaling* tanpa augmentasi agar data tetap asli dan memberikan evaluasi performa yang akurat. Implementasi augmentasi data dalam pemrograman digunakan kode program sebagai berikut.

```
# Import the necessary module from Keras for image data generation
from keras.preprocessing.image import ImageDataGenerator

# Create an ImageDataGenerator for the training data with various augmentations
train_datagen = ImageDataGenerator(
    rescale=1./255, # Rescale the pixel values from [0, 255] to [0, 1]
    rotation_range=20, # Randomly rotate images in the range (degrees, 0 to 20)
    width_shift_range=0.2, # Randomly shift images horizontally (fraction of total width)
    height_shift_range=0.2, # Randomly shift images vertically (fraction of total height)
    shear_range=0.2, # Apply shear transformations
    zoom_range=0.2, # Randomly zoom in and out on images
    fill_mode='nearest', # Fill in newly created pixels with the nearest pixel value
    brightness_range=[0.8, 1.2], # Randomly change the brightness of the images
    horizontal_flip=True # Randomly flip images horizontally
)

# Create an ImageDataGenerator for the validation data with only rescaling
val_datagen = ImageDataGenerator(rescale=1./255) # Rescale the pixel values from [0, 255] to [0, 1]
```

### 3.3. Modeling

#### Pre-trained Model Inception V3

Model *Inception V3* diimpor dari *TensorFlow Keras* dan bobot yang telah dilatih sebelumnya dimuat ke dalam model. Arsitektur *Inception V3* ini memiliki bentuk input gambar berukuran (416, 416, 3), dengan tiga saluran warna (RGB). Parameter *include\_top=False* digunakan untuk menghilangkan lapisan atas, sehingga memungkinkan penambahan lapisan baru sesuai kebutuhan aplikasi klasifikasi spesifik. Implementasi pre-trained Model *Inception V3* dalam bentuk kode program adalah sebagai berikut.

```
# Import the necessary modules from TensorFlow Keras
from tensorflow.keras.applications.inception_v3 import InceptionV3
from tensorflow.keras import layers

# Set the path to the weights file that you downloaded
local_weights_file = '/tmp/inception_v3_weights_tf_dim_ordering_tf_kernels_notop.h5'

# Initialize the base model from the InceptionV3 architecture.
# Set the input shape and exclude the fully connected layers (include_top=False).
pre_trained_model = InceptionV3(input_shape=(416, 416, 3), # Input shape of the images
include_top=False, # Exclude the fully connected layers
input_tensor='model_input', # Set the input tensor
weights='imagenet') # Weights will be loaded from the file, so set to None

# Load the pre-trained weights into the base model
pre_trained_model.load_weights(local_weights_file)

# Freeze the weights of the layers in the base model to prevent them from being updated during training
for layer in pre_trained_model.layers:
    layer.trainable = False # Set trainable to False to freeze the layers
```

#### Fine-tuning Inception V3

Setelah bobot *pre-trained* dimuat, model di-*freeze* (lapisan-lapisan dasar model tidak diperbarui selama pelatihan) dan lapisan tambahan ditambahkan. Langkah ini memungkinkan penyesuaian lebih lanjut dengan data spesifik yang sedang dianalisis. *Fine-tuning* dilakukan dengan menambahkan beberapa *fully connected layers* di atas model dasar. Lapisan tambahan ini menghubungkan output model *pre-trained* dengan tugas klasifikasi spesifik burung endemik Indonesia. Proses ini bertujuan untuk menyesuaikan model yang dilatih sebelumnya dengan dataset baru. Implementasi *Fine-tuning*

Inception V3 dalam kode program adalah sebagai berikut.

```
# Import the necessary libraries
from tensorflow.keras.optimizers import RMSprop
from tensorflow.keras.layers import Flatten, Dense, Dropout
from tensorflow.keras.models import Model

# Flatten the output layer to 1 dimension
x = Flatten()(last_output)
# Add a fully connected layer with 1,024 hidden units and ReLU activation
x = Dense(1024, activation='relu')(x)
# Add a dropout rate of 0.2
x = Dropout(0.2)(x)
# Add a final dense layer for classification with softmax activation (assuming 10 classes)
x = Dense(10, activation='softmax')(x)

# Create the final model
inception = Model(pre_trained_model.input, x)

# Compile the model with RMSprop optimizer and categorical crossentropy loss
inception.compile(optimizer=RMSprop(lr=0.0001),
                  loss='categorical_crossentropy',
                  metrics=['accuracy'])

# Print the model summary
inception.summary()
```

**Hyperparameter**

Hyperparameter yang digunakan dalam penelitian ini meliputi: *Optimizer* yang dipilih adalah Adam dengan learning rate sebesar 0.0001 untuk mempercepat konvergensi model. *Loss function* yang digunakan adalah *categorical\_crossentropy*, yang cocok untuk tugas klasifikasi multi-kelas. Selama pelatihan, performa model dipantau menggunakan metrik *accuracy*, dan diterapkan *callback* untuk menghentikan pelatihan secara otomatis jika akurasi mencapai lebih dari 95%, guna mencegah *overfitting*. Implementasi hyperparameter dalam kode program adalah sebagai berikut.

```
import tensorflow as tf
from tensorflow.keras.optimizers import Adam

# Definisikan kelas callback
class myCallback(tf.keras.callbacks.Callback):
    def on_epoch_end(self, epoch, logs={}):
        if logs.get('accuracy') > 0.95: # Menghentikan pelatihan jika akurasi mencapai lebih dari 95%
            print("\nTraining dihentikan karena akurasi mencapai 95%!")
            self.model.stop_training = True

# Membuat instance dari kelas callback
callbacks = myCallback()

# Kompilasi model Inception
inception.compile(optimizer=Adam(learning_rate=0.0001), # Menggunakan optimizer Adam dengan learning rate 0.0001
                  loss='categorical_crossentropy', # Fungsi loss yang digunakan untuk klasifikasi multikelas
                  metrics=['accuracy']) # Metrik evaluasi yang akan dimonitor selama pelatihan
```

**Pelatihan Model**

Pada tahap ini, model dilatih selama 10 *epoch* dengan *batch size* yang sudah ditetapkan. *Epoch* merupakan jumlah iterasi lengkap melalui seluruh dataset pelatihan, dan *batch size* adalah jumlah sampel data yang diproses sekaligus dalam satu iterasi. Tujuan dari pelatihan ini adalah untuk mengoptimalkan performa model dengan menggunakan parameter yang telah ditentukan sebelumnya. Implementasi pelatihan model dalam kode program adalah sebagai berikut.

```
# Melatih model Inception dengan data pelatihan dan validasi
history_inception = inception.fit(
    train_data, # Data pelatihan
    validation_data=validasi_data, # Data validasi
    steps_per_epoch=train_data.n // 32, # Jumlah langkah per epoch (dissesuaikan dengan ukuran batch)
    validation_steps=validasi_data.n // 32, # Jumlah langkah validasi per epoch (dissesuaikan dengan ukuran batch)
    epochs=10, # Jumlah epoch pelatihan
    callbacks=[callbacks] # Callback yang digunakan selama pelatihan
)
```

**3.4. Evaluasi**

**Pengujian Model**

Pengujian model dilakukan terhadap delapan set data yang berbeda, seperti yang ditunjukkan pada Gambar 2. Hasil evaluasi menunjukkan bahwa model dapat

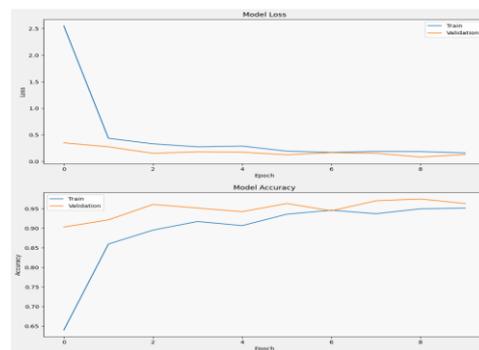
memprediksi dengan benar semua data uji yang diberikan *True* label dan hasil prediksi, serta memiliki besar akurasi untuk masing-masing dari delapan data uji. Ini menunjukkan bahwa model dilatih dengan baik dan mampu memberikan prediksi yang akurat pada data yang belum pernah dilihat sebelumnya.



Gambar 2. Hasil Pengujian Model

**Grafik Training**

Plot akurasi model yang ditunjukkan ditunjukkan pada Gambar 3. Berdasarkan Gambar 3, akurasi model terhadap data pelatihan (akurasi pelatihan) dan akurasi model terhadap data validasi (akurasi validasi) dinilai pada setiap *epoch*. Dalam contoh ini, jelas bahwa akurasi pada data pelatihan dan validasi meningkat secara bertahap seiring dengan berjalannya *epoch*, yang menunjukkan bahwa kinerja model semakin baik. Kedua kurva akurasi tersebut cenderung bergerak bersamaan, yang menunjukkan bahwa model tidak mengalami *overfitting* atau *underfitting*. Kedekatan kurva akurasi dan kehilangan pada kedua grafik menunjukkan bahwa model tersebut berhasil melakukan prediksi pada data pelatihan dan validasi.



Gambar 3. Grafik Training

3.7. Classification Report

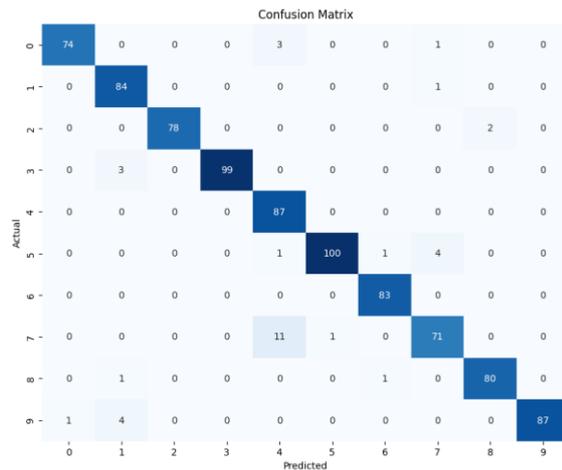
	precision	recall	f1-score	support
BURUNG CENDERAWASIH BOTAK	0.99	0.95	0.97	78
BURUNG CICA DAUN KALIMANTAN	0.91	0.99	0.95	85
BURUNG JALAK BALI	1.00	0.97	0.99	80
BURUNG JUNAI EMAS	1.00	0.97	0.99	102
BURUNG KUAU Kerdil Kalimantan	0.85	1.00	0.92	87
BURUNG MALEO	0.99	0.94	0.97	106
BURUNG PENCUK ULAR ASIA	0.98	1.00	0.99	83
BURUNG SEMPIDAN KALIMANTAN	0.92	0.86	0.89	83
BURUNG SEMPUR HUJAN DARAT	0.98	0.98	0.98	82
BURUNG TIONG BATU KALIMANTAN	1.00	0.95	0.97	92
accuracy			0.96	878
macro avg	0.96	0.96	0.96	878
weighted avg	0.96	0.96	0.96	878

Gambar 4. Classification Report

Berdasarkan gambar di atas, f1-score rata-rata model yang diuji menunjukkan 96% data yang benar dari 878 prediksi dari dataset yang tidak digunakan. Ini menunjukkan bahwa model yang dilatih telah dilakukan dengan baik.

Confusion Matrix

Evaluasi confusion matrix menunjukkan bahwa nilai untuk setiap kelas hampir sempurna, dengan hanya beberapa prediksi data yang salah. Ini menunjukkan bahwa model yang dapat memprediksi dengan akurat.



Gambar 5. Confusion Matrix

4. Kesimpulan

Penelitian ini menghasilkan model pembelajaran mesin yang dapat dengan akurat memprediksi sepuluh jenis burung endemik Indonesia. Selain itu, grafik akurasi dan kehilangan model menunjukkan peningkatan yang signifikan seiring dengan berjalannya waktu. Ini dibuktikan oleh laporan

klasifikasi dan confusion matrix yang diperoleh, yang menunjukkan bahwa model ini memiliki akurasi hingga 96% pada data uji di luar dataset. Diharapkan dengan adanya model ini, pengetahuan tentang klasifikasi jenis burung endemik Indonesia akan ditingkatkan, dan Inception V3 akan digunakan sebagai sumber pembelajaran untuk klasifikasi burung.

Daftar Rujukan

- [1] Nurhaida and W. As, "Hubungan Perilaku Harian Ayam Terhadap Lingkungannya," 2023, doi: 10.31219/osf.io/cyevz.
- [2] N. Wolas, H. Hamzah, and M. E. Hiswati, "Aplikasi Deep Learning Menggunakan Algoritma Convolutional Neural Network (CNN) Untuk Klasifikasi Burung Famili Accipitridae," Seminar Nasional Teknik Elektro Informatika Dan Sistem Informasi, vol. 1, no. 1, 2022, doi: 10.35842/sintaks.v1i1.4.
- [3] D. Choe, E. Choi, and D. K. Kim, "The Real-Time Mobile Application for Classifying of Endangered Parrot Species Using the CNN Models Based on Transfer Learning," Mobile Information Systems, vol. 2020, pp. 1–13, 2020, doi: 10.1155/2020/1475164.
- [4] A. Tursunov, Mustaqeem, and S. Kwon, "Deep-Net: A Lightweight CNN-Based Speech Emotion Recognition System Using Deep Frequency Features," Sensors, vol. 20, no. 18, p. 5212, 2020, doi: 10.3390/s20185212.
- [5] R. S. Monteiro, M. C. M. Ribeiro, C. Viana, M. W. L. Moreira, G. Araújo, and J. J. P. C. Rodrigues, "Fish Recognition Model for Fraud Prevention Using Convolutional Neural Networks," 2021, doi: 10.21203/rs.3.rs-849174/v1.
- [6] A. Manna, N. Upasani, S. Jadhav, R. Mane, R. Chaudhari, and V. Chatre, "Bird Image Classification Using Convolutional Neural Network Transfer Learning Architectures," International Journal of Advanced Computer Science and Applications, vol. 14, no. 3, 2023, doi: 10.14569/ijacsa.2023.0140397.
- [7] B. Gardie, S. Asemie, K. Azezew, and Z. Solomon, "Potato Plant Leaf Diseases Identification Using Transfer Learning," Indian J Sci Technol, vol. 15, no. 4, pp. 158–165, 2022, doi: 10.17485/ijst/v15i4.1235.
- [8] Md. M. Rahman, A. A. Biswas, A. Rajbongshi, and A. Majumder, "Recognition of Local Birds of Bangladesh Using MobileNet and Inception-V3," International Journal of Advanced Computer Science and Applications, vol. 11, no. 8, 2020, doi: 10.14569/ijacsa.2020.0110840.