

PRAKTIKUM DATA MINING II SD-A1

BAB : BAB 9 – IMAGE CLASSIFICATION

PRAKTIKUM KE : 6

NAMA : RATU NOOR HASANNAH

NIM : 162112133112

TGL PRAKTIKUM : 9 NOVEMBER 2023

Praktikum dilaksanakan melalui Google Colab yang dapat diakses pada link berikut:
[link colab Ratu Noor Hasannah](#)

Import Packages

```
import numpy as np
from ipywidgets import interact, widgets
import os
import matplotlib.pyplot as plt
from PIL import Image
from sklearn.model_selection import train_test_split

import tensorflow as tf
from keras.models import Sequential
from keras.layers import
Dense, Activation, Dropout, Conv2D, MaxPooling2D, BatchNormalization, Flatten
from keras import optimizers
from keras import Model
from tensorflow.python.keras.utils import np_utils
from keras.preprocessing.image import ImageDataGenerator

from keras.layers import Dense, Dropout
from keras.models import Model
from keras.optimizers import Adamax
from keras.applications import efficientnet
from keras.layers import BatchNormalization

import splitfolders
import zipfile
import random
import shutil
from shutil import copyfile
from scipy import misc
from shutil import copyfile

import keras as keras
from tensorflow.python.keras.optimizers import RMSprop
from tensorflow.python.keras.optimizers import Adam, Adamax
from tensorflow.python.keras import regularizers
from tensorflow.python.keras.models import Model, load_model, Sequential

from google.colab import drive
import os
```

PRAKTIKUM DATA MINING II SD-A1

BAB : BAB 9 – IMAGE CLASSIFICATION

PRAKTIKUM KE : 6

NAMA : RATU NOOR HASANNAH

NIM : 162112133112

TGL PRAKTIKUM : 9 NOVEMBER 2023

```
from PIL import Image
import numpy as np
```

Load Data

```
drive.mount('/content/drive')
os.chdir('/content/drive/My Drive/Felidae')
```

Data diakses dari Google Drive dengan path My Drive → Felidae

My Drive > Felidae ▾				
Type ▾		People ▾	Modified ▾	
Name ↑		Owner	Last mo... ▾	File size

Split Data

Melakukan split data dengan rasio training dan validation data 0,8 dan 0,2

```
# @title Split Data
splitfolders.ratio('/content/drive/My Drive/Felidae', output="felidae_split",
seed=42, ratio=(.8, .2))
```

Setelah menjalankan code diatas, muncul folder ‘felidae_split’ pada ‘Felidae’ berisi data yang sudah displit menjadi train data dan validation data

My Drive > Felidae ▾				
Type ▾		People ▾	Modified ▾	
Name ↑		Owner	Last mo... ▾	File size
Cheetah	me	Nov 9, 2023	—	⋮
felidae_split	me	8:12 PM	—	⋮
Leopard	me	Nov 9, 2023	—	⋮
Lion	me	Nov 9, 2023	—	⋮
Puma	me	Nov 9, 2023	—	⋮
Tiger	me	Nov 9, 2023	—	⋮

BAB : BAB 9 – IMAGE CLASSIFICATION

PRAKTIKUM KE : 6

NAMA : RATU NOOR HASANNAH

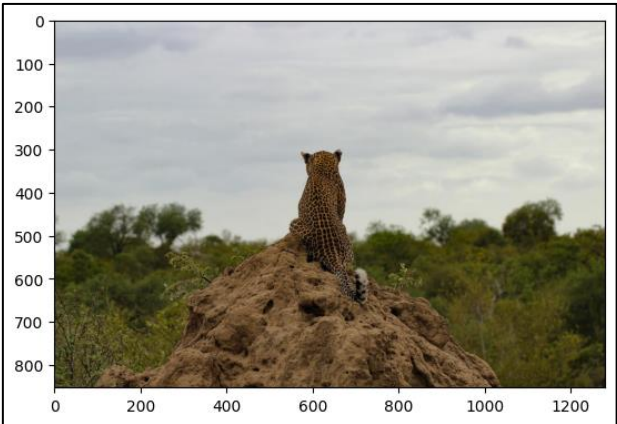
NIM : 162112133112

TGL PRAKTIKUM : 9 NOVEMBER 2023

My Drive > Felidae > felidae_split ▾					⌵ ⓘ
Type ▾		People ▾		Modified ▾	
Name	↑	Owner		Last mo... ▾	File size ⋮
train		me		8:12 PM	— ⋮
val		me		8:12 PM	— ⋮

Preview gambar

```
leopard = plt.imread('/content/drive/My Drive/Felidae/Leopard/Leopard_049.jpg')
imgplot = plt.imshow(leopard)
```



Preprocessing

```
# @title Preprocessing
train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=20,
    width_shift_range=0.15,
    height_shift_range=0.15,
    shear_range=0.15,
    zoom_range=0.2,
    #fill_mode='nearest'
)

train_generator = train_datagen.flow_from_directory(
```

PRAKTIKUM DATA MINING II SD-A1

BAB : BAB 9 – IMAGE CLASSIFICATION

PRAKTIKUM KE : 6

NAMA : RATU NOOR HASANNAH

NIM : 162112133112

TGL PRAKTIKUM : 9 NOVEMBER 2023

```
directory=r"felidae_split/train/",
target_size=(256, 256),
color_mode="rgb",
batch_size=12,
class_mode="categorical",
shuffle=True,
seed=42
)

validation_datagen = ImageDataGenerator()

validation_generator = validation_datagen.flow_from_directory(
    directory=r"felidae_split/val/",
    target_size=(256, 256),
    color_mode="rgb",
    batch_size=12,
    class_mode="categorical",
    shuffle=True,
    seed=42
)

number_of_classes = len(np.unique(train_generator.classes))
```

Train Data:

- Perintah rescale digunakan untuk normalisasi nilai piksel menjadi antara 0 dan 1
- Mengubah ukuran gambar menjadi 256x256 piksel
- Parameter augmentasi data (rotation_range, width_shift_range, height_shift_range, shear_range, zoom_range) untuk meningkatkan jumlah data pelatihan secara artifisial.
- zoom_range: Memperbesar atau memperkecil gambar hingga 20%
- color_mode: "rgb" untuk 3 saluran warna

Validation Data:

- flow_from_directory: Menghasilkan batch data pelatihan yang telah di-augmentasi dari gambar di direktori yang ditentukan.

Direktori yang digunakan untuk train dan val data berdasarkan direktori yang dihasilkan dari proses split data menggunakan ‘splitfolders’ yang sudah dilakukan sebelumnya.

Number_of_classes: untuk melihat jumlah kelas

Found 193 images belonging to 5 classes.
Found 50 images belonging to 5 classes.

BAB : BAB 9 – IMAGE CLASSIFICATION

PRAKTIKUM KE : 6

NAMA : RATU NOOR HASANNAH

NIM : 162112133112

TGL PRAKTIKUM : 9 NOVEMBER 2023

Diperoleh output diatas yang menunjukkan train data yang terdiri dari 193 gambar dan 5 kelas. Serta validation data yang terdiri dari 50 gambar dan 5 kelas.

Modelling

```
# @title Modelling
base_model = tf.keras.applications.efficientnet.EfficientNetB5(
include_top = False,
weights = 'imagenet',
input_shape = (256,256,3),
pooling = 'max')

x = base_model.output
x = keras.layers.BatchNormalization(axis = -1,momentum = 0.99,epsilon =
0.001)(x)

x = Dense(512,kernel_regularizer = regularizers.l2(l =
0.019),activity_regularizer = regularizers.l1(0.006),
      bias_regularizer = regularizers.l1(0.006),activation =
'relu')(x)

x = Dropout(rate = .42,seed = 42)(x)

output = Dense(number_of_classes,activation = 'softmax')(x)

model = Model(inputs = base_model.input,outputs = output)
model.compile(Adamax(learning_rate = 0.001),loss =
'categorical_crossentropy',metrics = ['accuracy'])
```

- Base Model:
base_model = tf.keras.applications.efficientnet.EfficientNetB5(...): Membuat model dasar menggunakan EfficientNetB5 dari TensorFlow. Model ini dilatih sebelumnya pada dataset ImageNet.
- Layer Tambahan:
 - x = base_model.output: Mengambil output dari model dasar.
 - x = keras.layers.BatchNormalization(...): Menambahkan lapisan normalisasi batch untuk mengurangi ketergantungan pada inisialisasi bobot.
- Fully Connected Layer (Dense Layer):
 - x = Dense(512, kernel_regularizer=..., activity_regularizer=..., bias_regularizer=..., activation='relu')(x): Menambahkan lapisan fully connected

BAB : BAB 9 – IMAGE CLASSIFICATION

PRAKTIKUM KE : 6

NAMA : RATU NOOR HASANNAH

NIM : 162112133112

TGL PRAKTIKUM : 9 NOVEMBER 2023

(dense) dengan 512 unit. Ini diikuti oleh fungsi aktivasi ReLU dan beberapa regularisasi seperti L2 kernel regularization, L1 activity regularization, dan L1 bias regularization.

- Dropout Layer:
 - `x = Dropout(rate=0.42, seed=42)(x)`: Menambahkan lapisan dropout untuk mencegah overfitting. Dropout secara acak "menonaktifkan" sejumlah unit selama pelatihan.
- Output Layer:
 - `output = Dense(number_of_classes, activation='softmax')(x)`: Menambahkan lapisan output dengan jumlah unit sesuai dengan jumlah kelas dalam dataset. Fungsi aktivasi softmax digunakan untuk menghasilkan probabilitas kelas.
- Model:
 - `model = Model(inputs=base_model.input, outputs=output)`: Membuat model keseluruhan dengan menghubungkan input dari model dasar ke output yang baru saja dibuat.
- Kompilasi Model:
 - `model.compile(Adamax(learning_rate=0.001), loss='categorical_crossentropy', metrics=['accuracy'])`: Mengompilasi model dengan pengoptimal Adamax, fungsi kerugian categorical crossentropy (cocok untuk masalah klasifikasi multikelas), dan metrik akurasi untuk evaluasi model.

Training and Validation Evaluation Metrics

```
from tensorflow.keras.callbacks import EarlyStopping

# Callback untuk menghentikan pelatihan jika tidak ada peningkatan akurasi validasi
early_stopping = EarlyStopping(monitor = 'val_loss',
                               patience = 5,
                               restore_best_weights = True)

best_val_accuracy = 0 # Inisialisasi variabel untuk menyimpan akurasi validasi terbaik
best_model = None # Inisialisasi variabel untuk menyimpan model terbaik

# Model training dengan menggunakan callback
history = model.fit(train_generator,
                    epochs = 15,
```

BAB : BAB 9 – IMAGE CLASSIFICATION

PRAKTIKUM KE : 6

NAMA : RATU NOOR HASANNAH

NIM : 162112133112

TGL PRAKTIKUM : 9 NOVEMBER 2023

```
verbose = 2,  
validation_data = validation_generator,  
callbacks=[early_stopping])
```

Untuk membantu mencegah overfitting dan mengoptimalkan performa model pada data validasi, digunakan `early_stopping`. Proses training model akan dihentikan jika tidak ada penurunan dalam validation loss selama 5 epoch berturut-turut. Model akan dikembalikan ke bobot terbaik yang dicapai selama pelatihan.

`early_stopping`:

- `monitor = 'val_loss'` → Memonitor nilai validation loss.
- `patience = 5` → Menetapkan jumlah epoch yang akan diizinkan tanpa penurunan dalam validation loss sebelum pelatihan dihentikan.
- `restore_best_weights = True` → Mengembalikan bobot model ke yang terbaik selama pelatihan jika penghentian diterapkan.

history :

- `model.fit(...)` → Melatih model dengan menggunakan generator data `train_generator` dan melakukan validasi dengan data `validation_generator`.
- `epochs = 15` → Menentukan jumlah epoch pelatihan.
- `verbose = 2` → Menampilkan detail pelatihan secara lebih lengkap.
- `callbacks = early_stopping` → Menyertakan callback `EarlyStopping` dalam proses pelatihan.

Output:

```
Epoch 1/15  
17/17 - 413s - loss: 18.3423 - accuracy: 0.4560 - val_loss: 22.1061 - val_accuracy: 0.8000 - 413s/epoch - 24s/step  
Epoch 2/15  
17/17 - 317s - loss: 16.8815 - accuracy: 0.7513 - val_loss: 18.9442 - val_accuracy: 0.9600 - 317s/epoch - 19s/step  
Epoch 3/15  
17/17 - 318s - loss: 15.9157 - accuracy: 0.8290 - val_loss: 17.9026 - val_accuracy: 0.9200 - 318s/epoch - 19s/step  
Epoch 4/15  
17/17 - 322s - loss: 15.1759 - accuracy: 0.8446 - val_loss: 16.1093 - val_accuracy: 0.9000 - 322s/epoch - 19s/step  
Epoch 5/15  
17/17 - 319s - loss: 14.3333 - accuracy: 0.9223 - val_loss: 14.9477 - val_accuracy: 0.9200 - 319s/epoch - 19s/step  
Epoch 6/15  
17/17 - 323s - loss: 13.7063 - accuracy: 0.9171 - val_loss: 14.0637 - val_accuracy: 0.9400 - 323s/epoch - 19s/step  
Epoch 7/15  
17/17 - 323s - loss: 13.0176 - accuracy: 0.9585 - val_loss: 13.2825 - val_accuracy: 0.9400 - 323s/epoch - 19s/step  
Epoch 8/15  
17/17 - 319s - loss: 12.5293 - accuracy: 0.9223 - val_loss: 12.5184 - val_accuracy: 0.9600 - 319s/epoch - 19s/step  
Epoch 9/15  
17/17 - 325s - loss: 11.8382 - accuracy: 0.9793 - val_loss: 11.9298 - val_accuracy: 0.9600 - 325s/epoch - 19s/step  
Epoch 10/15  
17/17 - 321s - loss: 11.3177 - accuracy: 0.9845 - val_loss: 11.4462 - val_accuracy: 0.9000 - 321s/epoch - 19s/step  
Epoch 11/15  
17/17 - 323s - loss: 10.8148 - accuracy: 0.9741 - val_loss: 10.9699 - val_accuracy: 0.9000 - 323s/epoch - 19s/step  
Epoch 12/15  
17/17 - 325s - loss: 10.3090 - accuracy: 0.9637 - val_loss: 10.5109 - val_accuracy: 0.9000 - 325s/epoch - 19s/step  
Epoch 13/15  
17/17 - 330s - loss: 9.8575 - accuracy: 0.9534 - val_loss: 10.0208 - val_accuracy: 0.8800 - 330s/epoch - 19s/step  
Epoch 14/15  
17/17 - 354s - loss: 9.4051 - accuracy: 0.9689 - val_loss: 9.5243 - val_accuracy: 0.9000 - 354s/epoch - 21s/step  
Epoch 15/15  
17/17 - 325s - loss: 8.9681 - accuracy: 0.9534 - val_loss: 9.1353 - val_accuracy: 0.9200 - 325s/epoch - 19s/step
```

```
acc = history.history['accuracy']
```

PRAKTIKUM DATA MINING II SD-A1

BAB : BAB 9 – IMAGE CLASSIFICATION

PRAKTIKUM KE : 6

NAMA : RATU NOOR HASANNAH

NIM : 162112133112

TGL PRAKTIKUM : 9 NOVEMBER 2023

```
val_acc = history.history['val_accuracy']
loss = history.history['loss']
val_loss = history.history['val_loss']

epochs = range(len(acc))

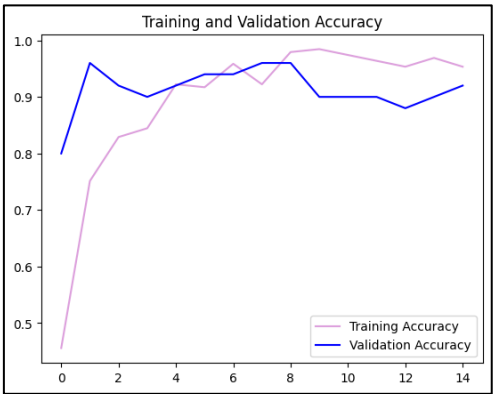
# Plot Training and Validation Accuracy
plt.plot(epochs, acc, 'plum', label='Training Accuracy')
plt.plot(epochs, val_acc, 'b', label='Validation Accuracy')
plt.title('Training and Validation Accuracy')
plt.legend() # Menambahkan legenda

plt.show()

# Plot Training and Validation Loss
plt.plot(epochs, loss, 'm', label='Training Loss')
plt.plot(epochs, val_loss, 'lightseagreen', label='Validation Loss')
plt.title('Training and Validation Loss')
plt.legend() # Menambahkan legenda

plt.show()
```

Output:



Berdasarkan plot diatas, terlihat besar akurasi di kedua train dan validation naik pada epoch perama ke epoch kedua, lalu naik-turun pada range akurasi 0,8 hingga 1,0. Selisih akurasi pada train dan validation tidak terlalu besar sehingga dapat dikatakan tidak ada overfitting yang terlalu parah.

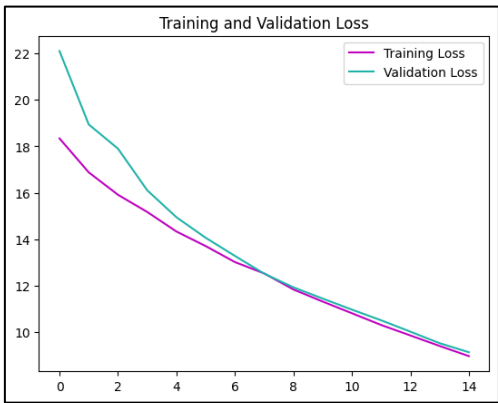
BAB : BAB 9 – IMAGE CLASSIFICATION

PRAKTIKUM KE : 6

NAMA : RATU NOOR HASANNAH

NIM : 162112133112

TGL PRAKTIKUM : 9 NOVEMBER 2023



Validation Loss adalah metrik yang mengukur seberapa baik model Anda melakukan prediksi pada data validasi yang tidak digunakan selama training model. Selama training model pada setiap epoch, sebagian kecil data dari set training digunakan untuk menghitung loss (kesalahan) model, dan hal ini membantu model untuk memperbarui bobotnya untuk meningkatkan kinerja pada data tersebut.

Berdasarkan plot diatas, terlihat validation dan training loss sama-sama menurun selama seluruh iterasi epoch. Berarti seiring pembelajaran model, kesalahan pada training dan validation terus menurun, hal ini menunjukkan bahwa model dapat memprediksi dengan lebih akurat seiring berjalannya epoch. Setelah beberapa epoch, training loss mulai menurun lebih lambat. Hal ini menunjukkan model telah mencapai titik optimal dan tidak dapat lagi meningkatkan akurasi prediksinya.

Confusion Matrix:

```
from sklearn.metrics import confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.metrics import classification_report
from tensorflow.python.keras.models import load_model

# Cari epoch dengan akurasi validasi terbaik
best_epoch = np.argmax(history.history['val_accuracy'])

# Muat model dari epoch dengan akurasi validasi terbaik
best_model = load_model(checkpoint_path.format(epoch=best_epoch))

# Gunakan model dengan akurasi terbaik untuk membuat confusion matrix
val_predictions = best_model.predict(validation_generator)
true_classes = validation_generator.classes
```

PRAKTIKUM DATA MINING II SD-A1

BAB : BAB 9 – IMAGE CLASSIFICATION

PRAKTIKUM KE : 6

NAMA : RATU NOOR HASANNAH

NIM : 162112133112

TGL PRAKTIKUM : 9 NOVEMBER 2023

```
predicted_classes = np.argmax(val_predictions, axis=1)

conf_matrix = confusion_matrix(true_classes, predicted_classes)

# Menampilkan confusion matrix menggunakan heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='PiYG', xticklabels=animal_names,
yticklabels=animal_names)
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()
```

BAB : BAB 9 – IMAGE CLASSIFICATION

PRAKTIKUM KE : 6

NAMA : RATU NOOR HASANNAH

NIM : 162112133112

TGL PRAKTIKUM : 9 NOVEMBER 2023

Prediction

Membuat fungsi untuk predict kelas:

```
def predict_animal(model, image, animal_names):  
    # Load and display the image  
    plt.imshow(image)  
  
    # Resize and preprocess the image  
    image = tf.image.resize(images=image, size=(256, 256))  
    image = tf.expand_dims(image, axis=0)  
  
    # Make prediction using the model  
    predictions = model.predict(image)  
    predicted_index = np.argmax(predictions[0])  
  
    # Display the prediction result  
    probability = round(predictions[0][predicted_index] * 100, 2)  
    predicted_animal = animal_names[predicted_index]  
  
    print(f"With a probability of {probability}%, this is a {predicted_animal}.")
```

Input Parameters:

- model: Model digunakan untuk melakukan prediksi.
- image: Gambar yang akan diprediksi.
- animal_names: Daftar nama kelas hewan yang mungkin.

Load and Display Image:

- plt.imshow(image): Menampilkan gambar.

Resize and Preprocess Image:

- image = tf.image.resize(images=image, size=(256, 256)): Mengubah ukuran gambar menjadi (256, 256).
- image = tf.expand_dims(image, axis=0): Menambah dimensi batch untuk memenuhi kebutuhan model.

Make Prediction Using the Model:

- predictions = model.predict(image): Menggunakan model untuk melakukan prediksi pada gambar yang telah diubah.

BAB : BAB 9 – IMAGE CLASSIFICATION

PRAKTIKUM KE : 6

NAMA : RATU NOOR HASANNAH

NIM : 162112133112

TGL PRAKTIKUM : 9 NOVEMBER 2023

- `predicted_index = np.argmax(predictions[0]):` Mengambil indeks kelas dengan probabilitas tertinggi.

Display the Prediction Result:

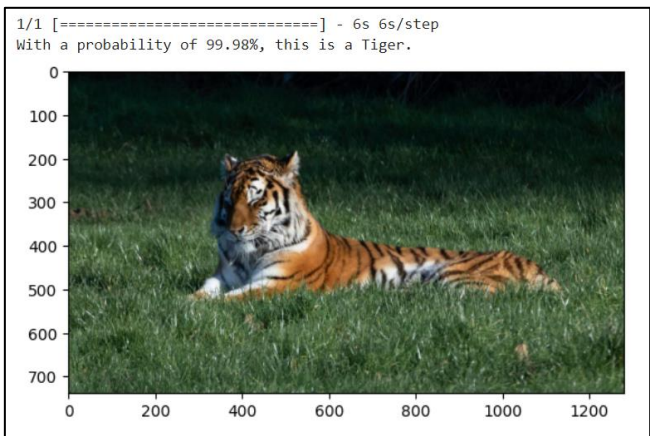
- `probability = round(predictions[0][predicted_index] * 100, 2):` Menghitung probabilitas prediksi dan membulatkannya.
- `predicted_animal = animal_names[predicted_index]:` Mendapatkan nama kelas hewan yang diprediksi.

Fungsi diatas akan menampilkan preview gambar yang akan dipredict beserta kelas atau label prediksi yang dihasilkan dengan probabilitasnya

```
animal_names = []
for key in train_generator.class_indices.keys():
    animal_names.append(key)
animal_names
```

Preview gambar dan label untuk data validation:

```
tiger = plt.imread('/content/drive/My Drive/Felidae/felidae_split/val/Tiger/Tiger_018.jpg')
predict_animal(model, tiger, animal_names)
```



Artinya dengan probabilitas sebesar 99,98% model berhasil memprediksi label gambar sebagai ‘Tiger’

Predict data test:

```
hewan_apa = plt.imread('/content/drive/My Drive/Test/test_1.jpeg')
predict_animal(model, hewan_apa, animal_names)
```

PRAKTIKUM DATA MINING II SD-A1

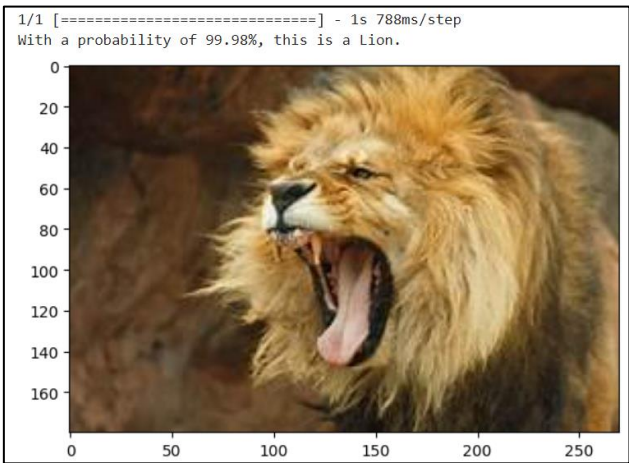
BAB : BAB 9 – IMAGE CLASSIFICATION

PRAKTIKUM KE : 6

NAMA : RATU NOOR HASANNAH

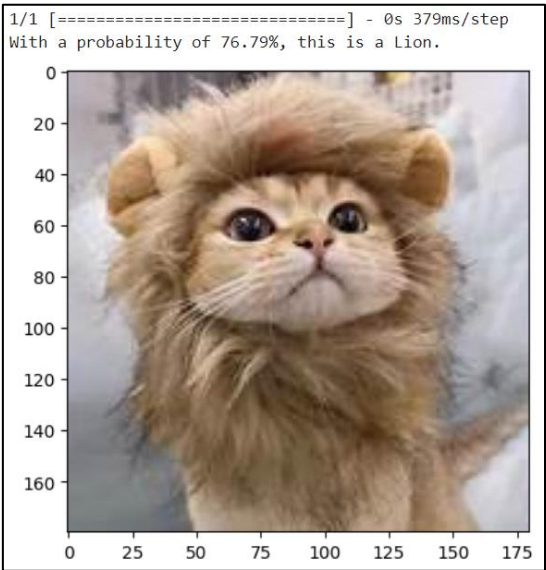
NIM : 162112133112

TGL PRAKTIKUM : 9 NOVEMBER 2023



Artinya dengan probabilitas sebesar 99,98% model berhasil meprediksi label gambar sebagai ‘Lion’

```
hewan_apa2 = plt.imread('/content/drive/My Drive/Test/test_2.jpeg')
predict_animal(model, hewan_apa2, animal_names)
```



Artinya dengan probabilitas sebesar 76,79% model berhasil memprediksi label gambar sebagai ‘Lion’