Akurasi Model Klasifikasi Hewan dengan Menggunakan MobileNet

Cenditya Ayu Aurelia – Kelas Fibonacci

Agenda

01 Latar Belakang Masalah dan Rumusan Masalah

02 Data dan Variabel Data

03 Preprocessing Data

04 Model dan Parameter Model

05 Ukuran Kebaikan Model

06 Kesimpulan

01

Latar Belakang Masalah dan Rumusan Masalah

Latar Belakang

Dalam melakukan identifikasi pada hewan memiliki persamaan maupun perbedaan sehingga perlu dilakukannya klasifikasi hewan untuk menentukan persamaan maupun perbedaan yang dimiliki. Klasifikasi hewan dapat dilakukan dengan menerapkan ilmu Computer Vision agar dapat memudahkan dalam melakukan klasifikasi hewan berupa gambar dari data gambar hewan yang dimiliki. Oleh karena itu, agar dapat menemukan evaluasi tingkat akurasi pada saat melakukan klasifikasi hewan dapat menerapkan model MobilNet pada ilmu Computer Vision. Dengan menerapkan model MobilNet diharapkan dapat menghasilkan akurasi yang lebih bagus dan akurat untuk melakukan klasifikasi pada hewan.

Rumusan Masalah

Berdasarkan latar belakang masalah tersebut, diperoleh rumusan masalah sebagai berikut :

1. Bagaimana hasil model MobilNet pada saat melakukan evaluasi akurasi klasifikasi hewan?
2. Apakah akurasi yang dihasilkan oleh model MobilNet dapat melakukan klasifikasi hewan dengan baik?

02

Data dan Variabel Data

Data dan Variabel Data

Data :
Dataset yang digunakan adalah data 4-animal-classification yang terdapat pada https://www.kaggle.com/competitions/4-animal-classification/data

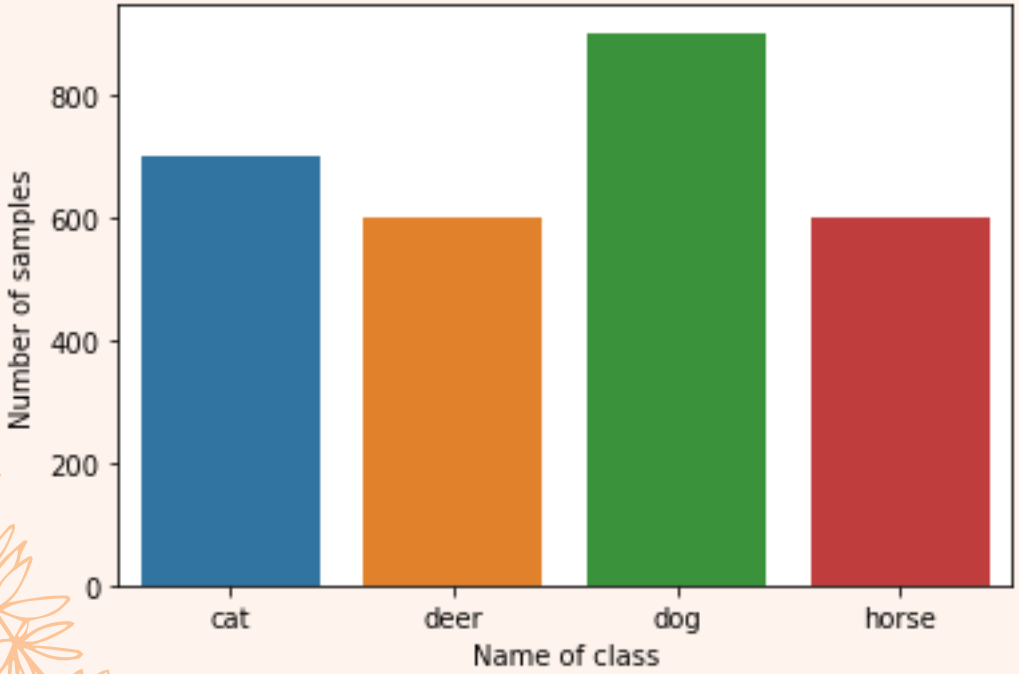
Variabel Data :
Berdasarkan dataset tersebut memiliki 4 kelas dengan pelabelan yang dimiliki, yaitu :

- Label 0 sebagai Cat
- Label 1 sebagai Deer
- Label 2 sebagai Dog
- Label 3 sebagai Horse

	Name of class	Number of samples
0	cat	700
1	deer	600
2	dog	900
3	horse	600

Data dan Variabel Data

Visualisasi data dengan menggunakan Bar Chart



Pada saat melakukan visualisasi dataset pada data train. Grafik tersebut menunjukkan bahwa data tertinggi dari keempat kelas dimiliki oleh kelas Dog dengan sebanyak 900 gambar, lalu kelas Cat sebanyak 700 gambar, kelas Deer dan Horse sebanyak 600 gambar.

03

Preprocessing Data

Preprocessing Data

```
image_datagen = tf.keras.preprocessing.image.ImageDataGenerator(rescale = 1./255 , rotation_range=20,
                                                                 width_shift_range=0.2,
                                                                 height_shift_range=0.2,
                                                                 horizontal_flip=True, validation_split=0.2)
```

Pada model diatas menggunakan ImageDateGenerator class untuk melakukan normalize data karena data yang akan diproses untuk improve performance secara keseluruhan. Parameter ‘rescale’ yang digunakan adalah 1./255 dengan width shift range dan height shift range yaitu 0.2. Lalu validation split yang digunakan pada metode ImageDataGenerator adalah 0.2.

```
# Flow training images in batches of 32 using train_ds
train_ds = image_datagen.flow_from_directory(
        path+'/train', # Direktori sumber untuk melakukan training images
        subset='training',
        target_size=(224 , 224), # Semua gambar akan diubah ukurannya menjadi 224x224
        batch_size=32)
```

```
# Flow validation images in batches of 32 using train_ds
val_ds = image_datagen.flow_from_directory(
        path+'/train', # Direktori sumber untuk melakukan training images
        subset='validation',
        target_size=(224 , 224), # Semua gambar akan diubah ukurannya menjadi 224x224
        batch_size=32 )
```

Dari output diatas, menunjukkan bahwa terdapat 2240 gambar dari 4 kelas pada train ds. Lalu terdapat 560 gambar dari 4 kelas pada validation ds.

04

Model dan Parameter Model

Model dan Parameter Model

```
model = tf.keras.models.Sequential([tf.keras.layers.Conv2D(64, (5, 5), strides=(2, 2),
                                                           activation='relu', padding = 'same',
                                                           input_shape = (200, 200, 3),
                                                           kernel_initializer='he_normal',
                                                           bias_initializer='zeros'),
                                    tf.keras.layers.MaxPool2D(pool_size = (2,2) ),
                                    tf.keras.layers.Conv2D(128, (3, 3), strides=(2, 2),
                                                           activation='relu', padding = 'same',
                                                           kernel_initializer='he_normal',
                                                           bias_initializer='zeros'),
                                    tf.keras.layers.MaxPool2D(pool_size = (2,2)),
                                    tf.keras.layers.Conv2D(128, (3, 3), strides=(2, 2),
                                                           activation='relu', padding = 'same',
                                                           kernel_initializer='he_normal',
                                                           bias_initializer='zeros'),
                                    tf.keras.layers.Flatten(),
                                    tf.keras.layers.Dense(256, activation=tf.nn.relu),
                                    tf.keras.layers.Dense(1, activation=tf.nn.sigmoid)])
```

Model yang digunakan adalah Conv2D dan MobileNet untuk menentukan model Sequential dengan menambahkan layering.

```
# Menambahkan accuracy pada metrics yang terdapat pa
model.compile(optimizer = tf.optimizers.Adam(),
              loss = 'binary_crossentropy',
              metrics=['accuracy'])
```

Berdasarkan kode diatas dilakukan untuk menambahkan accuracy pada metrics yang terdapat pada model sequential dengan melakukan train model dengan binary_crossentropy` loss dan Adam optimizer yang merupakan sensible optimization algorithm karena dapat melakukan automasi learning-rate tuning. Model tersebut akan memonitor accuracy saat melakukan training.

Model dan Parameter Model

```
# i'll be using inception_v3 for this model , along with 2 extra dense layers and the output layer

mobilenet = tf.keras.applications.mobilenet.MobileNet(input_shape=(224 , 224, 3),
                                                       include_top=False,
                                                       weights='imagenet')

model = Sequential()
model.add(mobilenet)
model.add(GlobalAveragePooling2D())
model.add(Flatten())
model.add(Dense(1024, activation="relu"))
model.add(Dense(512, activation="relu"))
model.add(Dense(4, activation="softmax" , name="classification"))
```

```
model.compile(optimizer=tf.keras.optimizers.SGD(learning_rate=0.0005,momentum=0.9),
              loss='categorical_crossentropy',
              metrics = ['accuracy'])

model.summary()
```

Berdasarkan kode diatas dilakukan untuk menambahkan accuracy pada metrics yang terdapat pada model sequential dengan menggunakan model MobileNet Dimana train model “categorical_crossentropy”` loss dan “SGD` optimizer” memiliki learning_rate = 0.0005 dan momentum = 0.9.

Training and Evaluating

```
# Melakukan training dengan 40 epochs
history = model.fit(train_ds , validation_data = val_ds , epochs = 40)
```

```
0.9607
Epoch 35/40
70/70 [==============================] - 579s 8s/step - loss: 0.0155 - accuracy: 0.9960 - val_loss: 0.1301 - val_accuracy: 0.
9607
Epoch 36/40
70/70 [==============================] - 464s 7s/step - loss: 0.0087 - accuracy: 0.9991 - val_loss: 0.1187 - val_accuracy: 0.
9643
Epoch 37/40
70/70 [==============================] - 469s 7s/step - loss: 0.0141 - accuracy: 0.9964 - val_loss: 0.1477 - val_accuracy: 0.
9554
Epoch 38/40
70/70 [==============================] - 464s 7s/step - loss: 0.0117 - accuracy: 0.9982 - val_loss: 0.1600 - val_accuracy: 0.
9643
Epoch 39/40
70/70 [==============================] - 452s 6s/step - loss: 0.0104 - accuracy: 0.9982 - val_loss: 0.1485 - val_accuracy: 0.
9589
Epoch 40/40
70/70 [==============================] - 465s 7s/step - loss: 0.0130 - accuracy: 0.9964 - val_loss: 0.1055 - val_accuracy: 0.
9732
```

Melakukan training dan evaluating untuk melihat nilai loss, val loss, accuracy, dan val accuracy pada model dengan menggunakan epochs sebanyak 40.

05

Ukuran Kebaikan Model

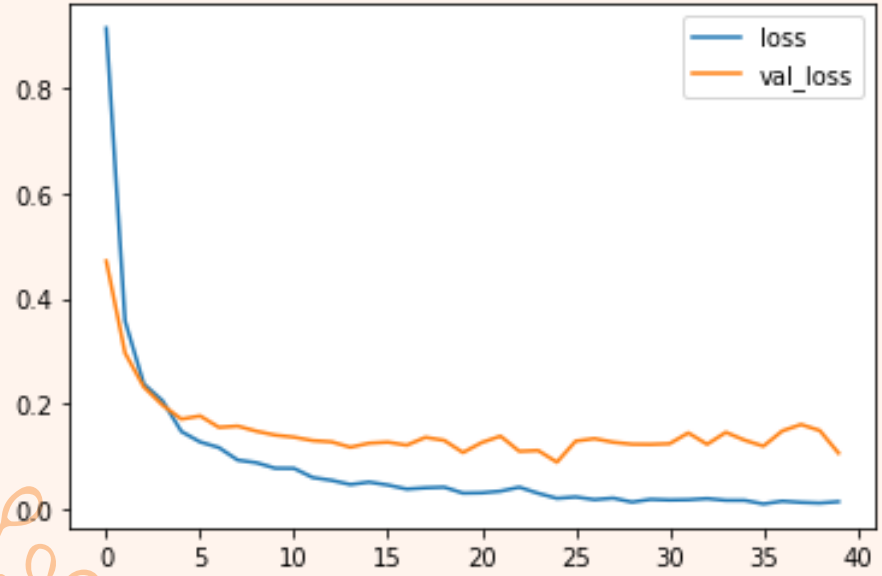
Ukuran Kebaikan Model

```
# Melakukan evaluasi accuracy model
model.evaluate(val_ds)

plt.figure()
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.legend(['loss','val_loss'],loc='upper right')
plt.show()


plt.figure()
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.legend(['accuracy','val_accuracy'],loc='upper right')
plt.show()
```

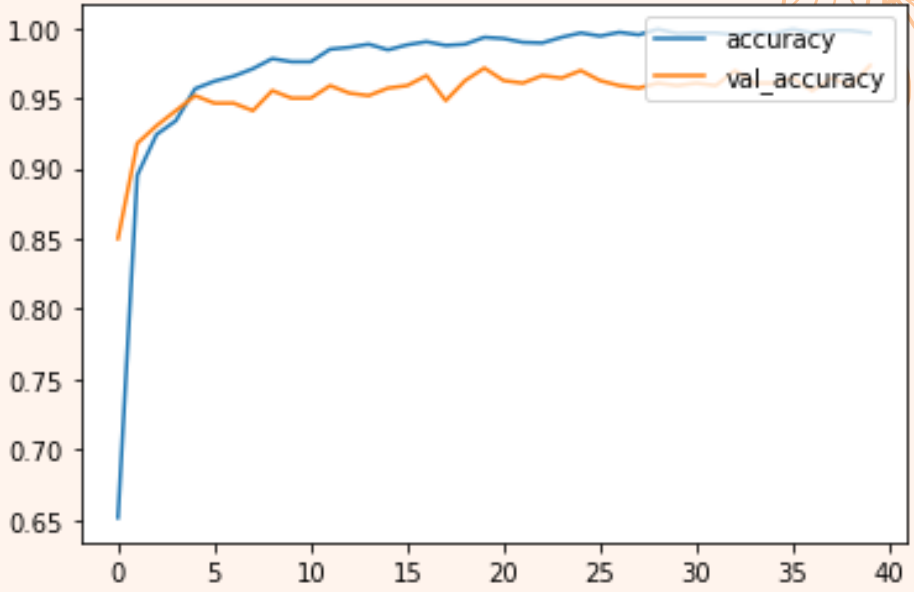
```
18/18 [==============================] - 29s 2s/step - los
s: 0.1327 - accuracy: 0.9607
```

Setelah melakukan train model pada model Sequential dengan “categorical_ crossentropy” loss dan “SGD optimizer “ learning rate = 0.0005, momentum = 0.9, dan epochs sebanyak 40 maka diperoleh akurasi sebesar 0.9607 dan nilai loss sebesar 0.1327.

Ukuran Kebaikan Model



Grafik diatas merupakan perbandingan tingkat loss dan val_loss yang menunjukkan penurunan.



Grafik diatas merupakan perbandingan tingkat accuracy dan val_accuracy yang menunjukkan kenaikkan.

06

Leaderboard & Ranking

Leaderboard

Raw Data | Refresh

YOUR RECENT SUBMISSION

file_submission.csv
Submitted by Cenditya Ayu Aurelia · Submitted just now

Score: 0.97393

Jump to your leaderboard position

42	Cenditya Ayu Aurelia	0.97393	1	1s

Your First Entry!
Welcome to the leaderboard!

07

Kesimpulan

Kesimpulan

Berdasarkan model yang digunakan untuk mengklasifikasikan hewan pada model MobileNet dengan memasukkan epochs sebesar 40 diperoleh tingkat akurasi sebesar 0,9607 atau 96% dan nilai loss sebesar 0.1327 atau 13%. Akurasi yang dihasilkan dari model MobilNet sudah bagus sehingga dapat mengklasifikasikan 4 hewan dengan baik.

Terima Kasih

@cendityaaurelia

Cenditya Ayu Aurelia

Universitas Pembangunan Nasional “Veteran”
Jawa Timur