

# Mengklasifikasikan Gambar Hewan dengan Metode MobileNet

NAMA : FIRDA AULIA  
KELAS : INSIGHT

---

## Latar Belakang

---

Convolutional Neural Network dengan arsitektur Mobilenet salah satu metode Deep Learning yang dapat digunakan untuk mengenali dan mengklasifikasi suatu objek. Membuat sebuah model untuk mengklasifikasikan gambar memiliki banyak potensi untuk membuat berbagai macam aplikasi. Klasifikasi gambar sudah bekerja di aplikasi foto, dan bisa menandai foto yang di klasifikasi sebagai wajah atau semacamnya. Klasifikasi Gambar dapat digunakan untuk mengenali sel-sel kanker, untuk mengenali kapal dalam citra satelit, atau untuk secara otomatis mengklasifikasikan gambar pada Pinterest. Klasifikasi bahkan dapat digunakan di luar ranah gambar, menganalisis peta panas dari aktivitas manusia untuk mendeteksi potensi pencurian, atau transformasi gelombang audio Fourier (Aurélien, 2017). Dalam penelitian ini peneliti menggunakan Pre-trained Model MobileNet yang memiliki dasar CNN yang merupakan salah satu metode Deep Learning yang sering digunakan dalam pengklasifikasian gambar

## Rumusan Masalah

---

Bagaimana cara untuk memprediksi gambar hewan dengan menggunakan MobileNet dengan menghasilkan accuracy tertinggi ?

# Variabel Data

Variabel data yang digunakan yaitu pada Kaggle: 4 animal classification

<https://www.kaggle.com/competitions/4-animal-classification>

```
[ ] 1 print('Jumlah training cat images :', len(os.listdir(train_cat_dir)))  
    2 print('Jumlah training deer images :', len(os.listdir(train_deer_dir)))  
    3 print('Jumlah training dog images :', len(os.listdir(train_dog_dir)))  
    4 print('Jumlah training horse images :', len(os.listdir(train_horse_dir)))
```

```
Jumlah training cat images : 700  
Jumlah training deer images : 600  
Jumlah training dog images : 900  
Jumlah training horse images : 600
```

```
1 train_dataset.class_indices
```

```
{'cat': 0, 'deer': 1, 'dog': 2, 'horse': 3}
```

```
1 validation_dataset.samples
```

```
560
```

# Preprocessing Data

Preprocessing Data yang digunakan adalah ImageDataGenerator

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

```
✓ [29] 1 train_dataset = image_datagen.flow_from_directory(  
0s      2     data+ '/train',  
      3     subset='training',  
      4     target_size=(224 , 224),  
      5     batch_size=32)  
      6  
      7 validation_dataset = image_datagen.flow_from_directory(  
      8     data+ '/train',  
      9     subset='validation',  
     10     target_size=(224 , 224),  
     11     batch_size=32)
```

Found 2240 images belonging to 4 classes.  
Found 560 images belonging to 4 classes.

# Model dan Parameter Model

Model yang digunakan adalah **MobileNet**

```
1 mobilenet = tf.keras.applications.mobilenet.MobileNet(input_shape=(224, 224, 3),
2                                                         include_top=False,
3                                                         weights='imagenet')
4
5 model = Sequential()
6 model.add(mobilenet)
7 model.add(GlobalAveragePooling2D())
8 model.add(Flatten())
9 model.add(Dense(512, activation="relu"))
10 model.add(Dense(512, activation="relu"))
11 model.add(Dense(4, activation="softmax", name="classification"))
```

```
1 model.compile(optimizer=tf.keras.optimizers.SGD(learning_rate=0.0005,momentum=0.9),
2               loss='categorical_crossentropy',
3               metrics=['accuracy'])
4
5 model.summary()
```

# Ukuran Keباikan Model yang Dihasilkan

```
1 history = model.fit(train_dataset,
2     steps_per_epoch=18,
3     epochs =15,
4     verbose=1,
5     validation_data = validation_dataset,
6     validation_steps=18)
```

Epoch 1/15  
18/18 [=====] - 142s 8s/step - loss: 0.1317 - accuracy: 0.9670 - val\_loss: 0.1716 - val\_accuracy: 0.9429  
Epoch 2/15  
18/18 [=====] - 141s 8s/step - loss: 0.1596 - accuracy: 0.9514 - val\_loss: 0.1671 - val\_accuracy: 0.9464  
Epoch 3/15  
18/18 [=====] - 140s 8s/step - loss: 0.1521 - accuracy: 0.9497 - val\_loss: 0.1686 - val\_accuracy: 0.9393  
Epoch 4/15  
18/18 [=====] - 141s 8s/step - loss: 0.1151 - accuracy: 0.9670 - val\_loss: 0.1588 - val\_accuracy: 0.9500  
Epoch 5/15  
18/18 [=====] - 140s 8s/step - loss: 0.1045 - accuracy: 0.9757 - val\_loss: 0.1511 - val\_accuracy: 0.9500  
Epoch 6/15  
18/18 [=====] - 143s 8s/step - loss: 0.1148 - accuracy: 0.9757 - val\_loss: 0.1640 - val\_accuracy: 0.9429  
Epoch 7/15  
18/18 [=====] - 140s 8s/step - loss: 0.1152 - accuracy: 0.9688 - val\_loss: 0.1526 - val\_accuracy: 0.9429  
Epoch 8/15  
18/18 [=====] - 140s 8s/step - loss: 0.1128 - accuracy: 0.9670 - val\_loss: 0.1686 - val\_accuracy: 0.9429  
Epoch 9/15  
18/18 [=====] - 140s 8s/step - loss: 0.1196 - accuracy: 0.9653 - val\_loss: 0.1527 - val\_accuracy: 0.9554  
Epoch 10/15  
18/18 [=====] - 140s 8s/step - loss: 0.1051 - accuracy: 0.9757 - val\_loss: 0.1409 - val\_accuracy: 0.9429  
Epoch 11/15  
18/18 [=====] - 140s 8s/step - loss: 0.0928 - accuracy: 0.9757 - val\_loss: 0.1354 - val\_accuracy: 0.9607  
Epoch 12/15  
18/18 [=====] - 141s 8s/step - loss: 0.1075 - accuracy: 0.9618 - val\_loss: 0.1485 - val\_accuracy: 0.9518  
Epoch 13/15  
18/18 [=====] - 140s 8s/step - loss: 0.1118 - accuracy: 0.9670 - val\_loss: 0.1366 - val\_accuracy: 0.9518  
Epoch 14/15  
18/18 [=====] - 140s 8s/step - loss: 0.0883 - accuracy: 0.9740 - val\_loss: 0.1377 - val\_accuracy: 0.9482  
Epoch 15/15  
18/18 [=====] - 140s 8s/step - loss: 0.0919 - accuracy: 0.9757 - val\_loss: 0.1408 - val\_accuracy: 0.9500

# Leaderboard & Ranking

82

Firda Aulia



0.96707

6

1s



Your Best Entry!

Your most recent submission scored 0.96707, which is an improvement of your previous score of 0.95610. Great job!

Tweet this



## KESIMPULAN

---

Berdasarkan hasil pengujian dan analisis yang sudah saya lakukan, didapatkan kesimpulan dalam membangun arsitektur model Convolutional neural network dengan menggunakan model MobileNet untuk pengklasifikasian Gambar hewan didapatkan nilai loss value terkecil dengan nilai 0.0883 pada epoch ke 14 dan mendapatkan tingkat nilai akurasi terbesar pada epoch ke 15 yaitu 0.9757.