

Klasifikasi 4 Kelas Hewan menggunakan MobileNet

Ditulis oleh Giselle Halim (Kelas Tensor)

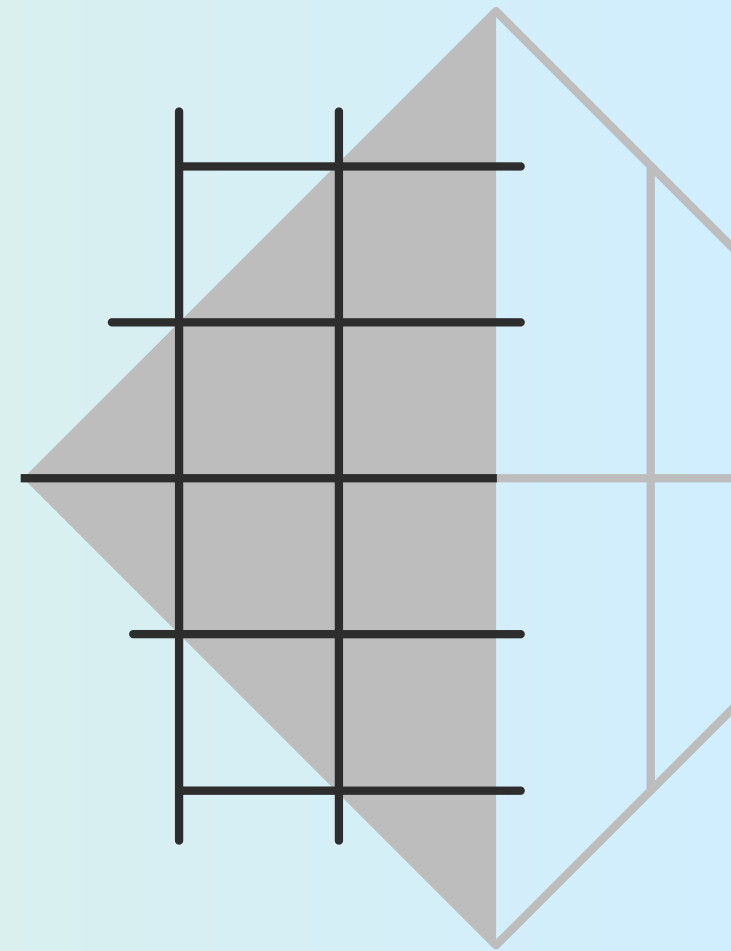
AI Mastery | Kampus Merdeka 4

Ujian Praktik | Domain Computer Vision

Latar Belakang

Computer-aided recognition adalah sebuah alat atau metode yang berguna untuk klasifikasi gambar tertentu dengan benar dan sesuai dengan di dunia nyata dengan bantuan AI (artificial intelligence). Banyak metode yang dapat digunakan untuk melakukan computer-aided recognition, salah satunya akan dicontohkan pada tugas ini.

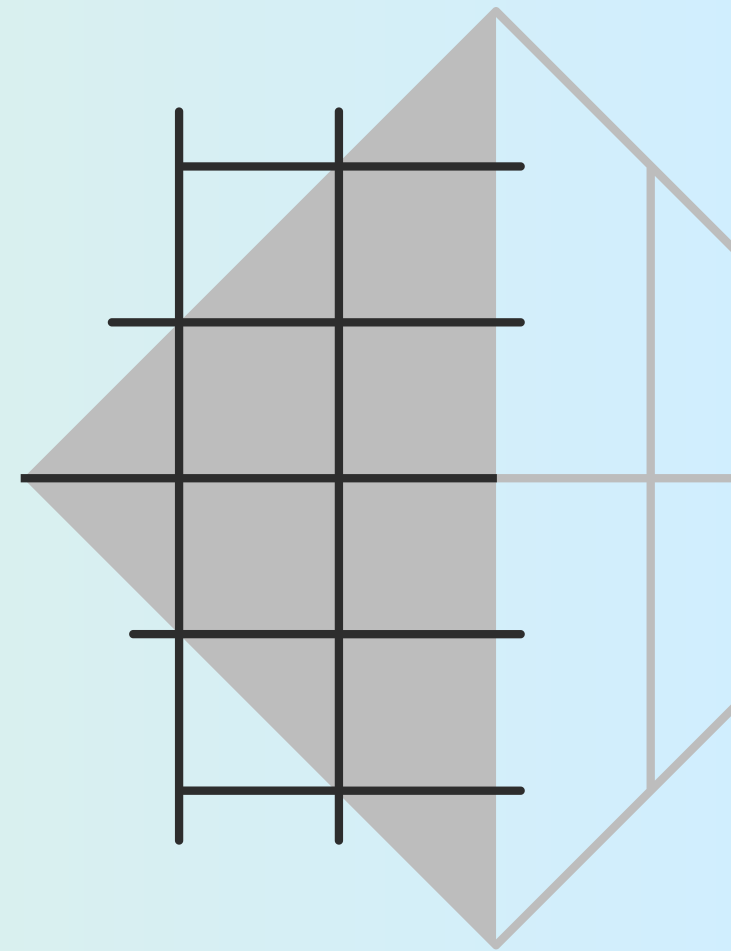
Pada tugas kali ini, akan dilakukan klasifikasi pada sejumlah foto hewan yang dibagi menjadi beberapa kelas dengan menggunakan metode CNN dengan MobileNet.



Rumusan Masalah

Rumusan masalah dari tugas ini adalah:

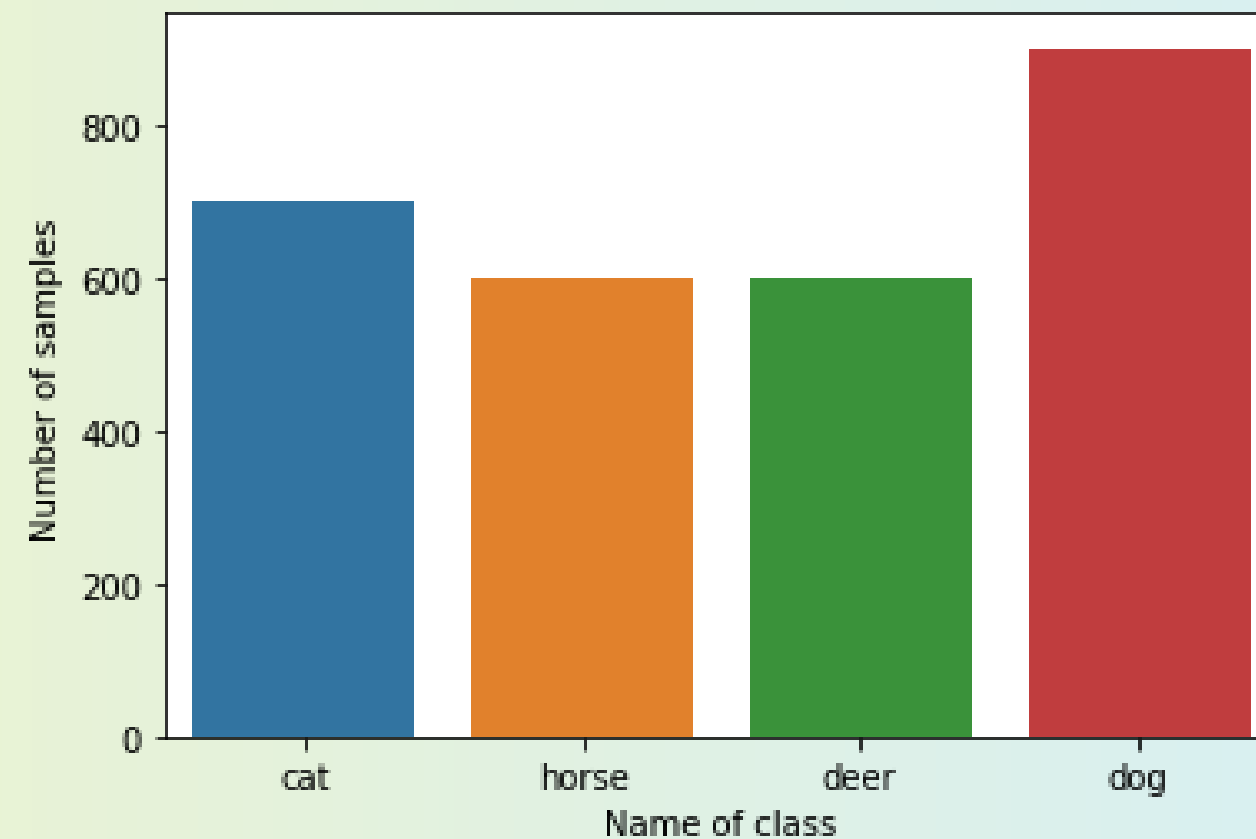
- Bagaimana tahapan melakukan data preprocessing pada dataset ini?
- Arsitektur apa yang akan digunakan untuk dataset ini dan apa saja parameter yang digunakan?



Data yang digunakan

Dataset yang digunakan adalah dataset kumpulan hewan yang terdiri dari 2800 train images dan 729 test images yang terbagi menjadi 4 class/jenis hewan, yaitu:

- 700 gambar kucing (cat)
- 600 gambar kuda (horse)
- 600 gambar rusa (deer),
- 900 gambar anjing (dog)



Data Preprocessing

Image Augmentation dan Splitting Dataset

Dataset gambar diatur dengan:

- rescale = 1./255 (normalisasi input)
- rotation_range=20 (pemutaran gambar maksimal 20 derajat)
- width dan height_shift_range=0.2 (penggeseran gambar maksimal 0.2)
- horizontal_flip=True (ini berarti images dapat di flip secara horizontal untuk menambah dataset dan meningkatkan akurasi).
- Untuk proses training, dilakukan splitting data training dan data validasi dengan rasio 80:20.

```
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)
```

Modelling

Digunakan arsitektur MobileNet untuk membangun model. Model dibangun secara sequential dan diberi tambahan GlobalAveragePooling2D, Flatten, 2 layer dense, dan 1 output layer.

Layer dense terdiri dari:

- Layer 1: 1024, dengan activation ReLu
- Layer 2: 512, dengan activation ReLu
- Output Layer: 4 (sesuai jumlah kelas), dengan activation Softmax

```
# Using InceptionV3, 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"))
```

Experiment Design

Optimizer

Optimizer yang digunakan adalah SGD dengan learning rate 0.0005 dan momentum 0.9

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

Epochs dan Batch Size

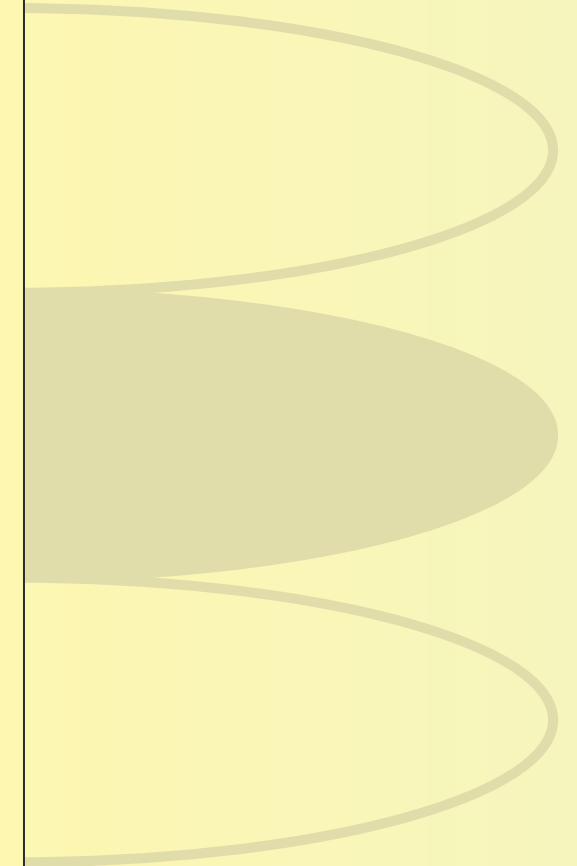
Dilakukan pelatihan model dengan epochs (pengulangan) sebanyak 25 kali dan batch size sebanyak 32

Metric Evaluation

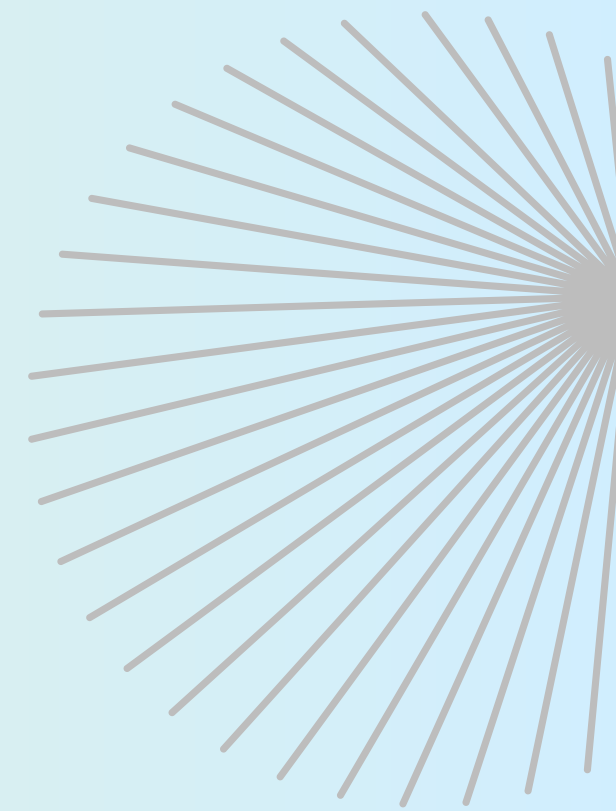
Untuk mengevaluasi model, digunakan metric validation accuracy dan validation loss yang juga digambarkan ke dalam grafik. Loss yang digunakan adalah categorical crossentropy



Hasil Experiment



Dari experimen model yang telah dilakukan, hasil yang didapat menunjukkan bahwa akurasi validasi akhir adalah sebesar 0.9607 atau 96.07% dengan loss sebesar 12.54%. Selain itu, pada proses pelatihan, nilai loss training menurun dan akurasi training cenderung meningkat sedangkan nilai loss dan akurasi validasi naik turun.

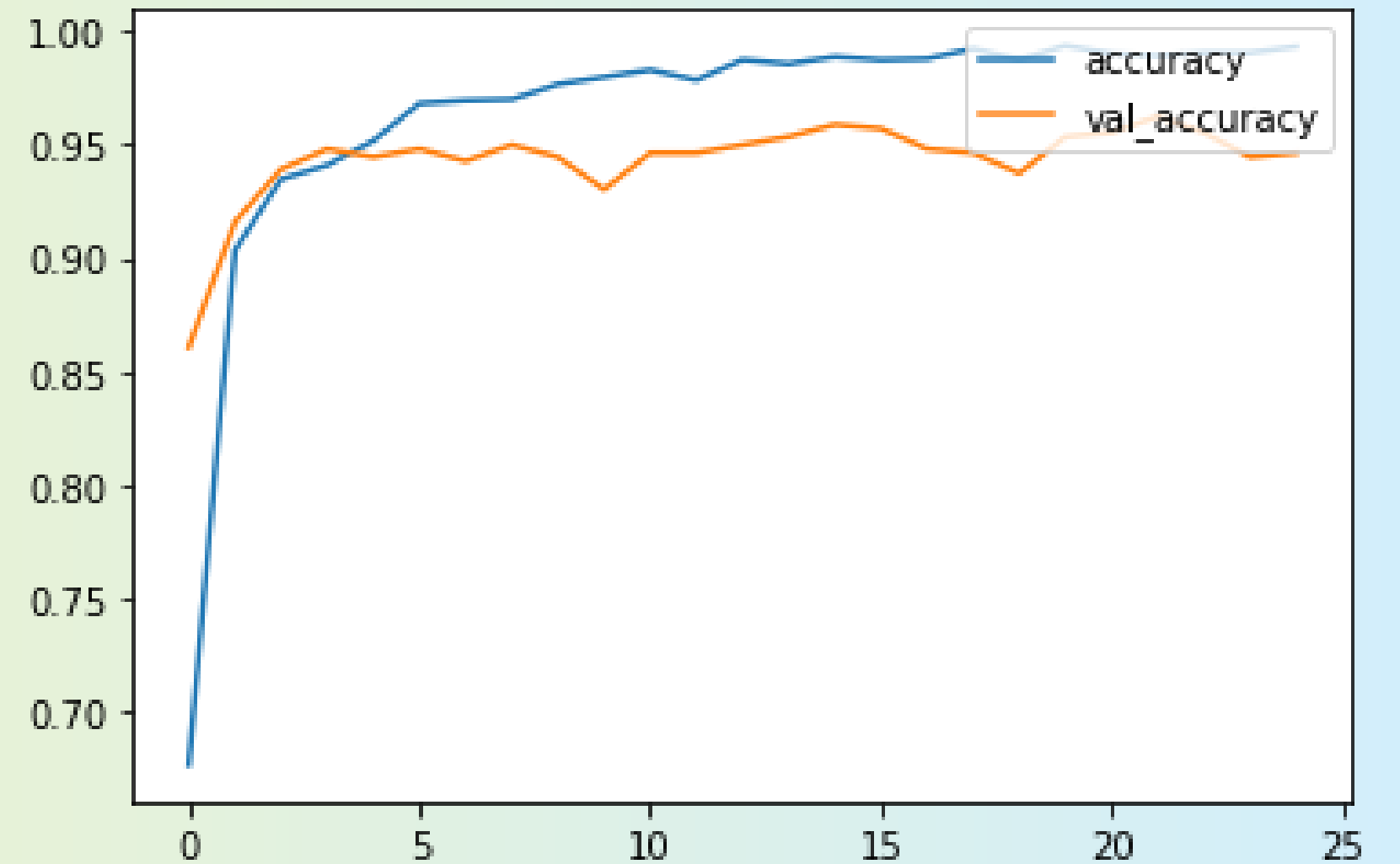
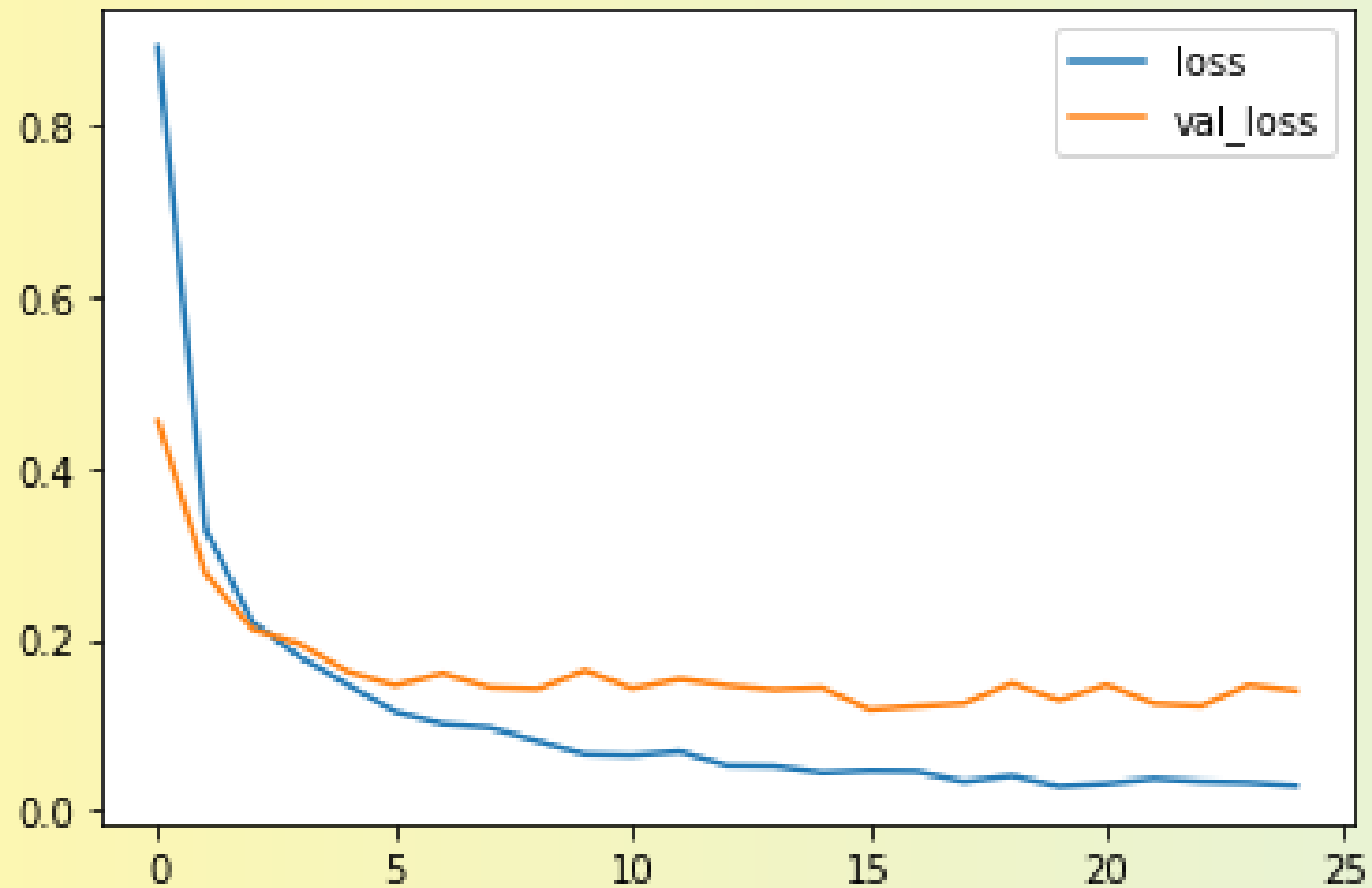



```
model.evaluate(val_ds)

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 [=====] - 10s 545ms/step - loss: 0.1254 - accuracy: 0.9607
```



Screenshot Leaderboard Kaggle

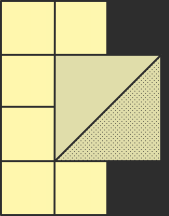
per 27 Maret 2023

38	Irma Amanda Putri		0.97668	3	2d
39	Afni yah		0.97668	12	2d
40	Abdallah Wagih Ibrahim		0.97530	2	3h
41	Linsay Berlian		0.97530	3	14h
42	Nur Fitriana Widuri #2		0.97530	2	4d
43	『η. αυλία αβρααμηςη』		0.97530	9	7h
44	Akhiril Anwar Harahap		0.97530	8	19h
45	Titania Ferrodova		0.97530	6	3h
46	Giselle Halim		0.97530	11	2m



Kesimpulan

- Tahap data preprocessing yang dilakukan adalah dengan normalisasi dataset dan menentukan range pemutaran gambar untuk menjadikan model lebih robust.
- Model yang dibangun dengan arsitektur MobileNet memiliki akurasi validasi sebesar 96.07% dengan loss sebesar 12.54%.
- Kelemahan model yang dibangun adalah nilai akurasi dan loss pada data validasi yang naik turun pada setiap epochs/pengulangan dan nilai akurasi validasi yang berbeda setiap kali menjalankan program.
- Deep learning mampu mengenali gambar hewan dengan baik karena komputer melihat pola dari setiap gambar dan mengkategorikannya menjadi beberapa kelas, selain itu, pada algoritma/model bawaan, biasanya sudah ada pelatihan khusus untuk mengenali objek-objek yang ada di dunia sehingga mempermudah proses klasifikasi.



Thank you for listening

GOOGLE COLAB LINK

AI Mastery | Kampus Merdeka 4

Ujian Praktik | Domain Computer Vision

