input image (test1.jpg)

Output

```
1/1 [==============================] - 1s 1s/step
Class: Robot Arm
Confidence Score: 0.9955468
```

Welcome To Colab - Colab | Untitled0.ipynb - Colab | colab.google | google colab - بحث | ربوت ارم - بحث

https://colab.research.google.com/drive/1qs57p7BX3zGRTTTGcjutgwFcxu1EegfO?authuser=0&hl=en#scrollTo=giz_rXETqskD

Untitled0.ipynb

File Edit View Insert Runtime Tools Help

Share Gemini

Commands + Code + Text Run all

RAM Disk

Files

..
sample_data
keras_model.h5
labels.txt
test.jpg

Disk 67.51 GB available

```python
from keras.models import load_model  # TensorFlow is required for Keras to work
from PIL import Image, ImageOps  # Install pillow instead of PIL
import numpy as np

# Disable scientific notation for clarity
np.set_printoptions(suppress=True)

# Load the model
model = load_model("keras_model.h5", compile=False)

# Load the labels
class_names = open("labels.txt", "r").readlines()

# Create the array of the right shape to feed into the keras model
# The 'length' or number of images you can put into the array is
# determined by the first position in the shape tuple, in this case 1
data = np.ndarray(shape=(1, 224, 224, 3), dtype=np.float32)

# Replace this with the path to your image
image = Image.open("test.jpg").convert("RGB")

# resizing the image to be at least 224x224 and then cropping from the center
size = (224, 224)
image = ImageOps.fit(image, size, Image.Resampling.LANCZOS)

# turn the image into a numpy array
image_array = np.asarray(image)

# Normalize the image
normalized_image_array = (image_array.astype(np.float32) / 127.5) - 1

# Load the image into the array
data[0] = normalized_image_array

# Predicts the model
prediction = model.predict(data)
index = np.argmax(prediction)
class_name = class_names[index]
confidence_score = prediction[0][index]

# Print prediction and confidence score
print("Class:", class_name[2:], end="")
print("Confidence Score:", confidence_score)
```

```
1/1 [==============================] - 1s 1s/step
Class: Robot Arm
Confidence Score: 0.9955468
```

Variables Terminal

7:58 PM Python 3

input image (test1.jpg)

Output

```
1/1 [==============================] -
Class: Humanoid Robot
Confidence Score: 0.9883116
```

Welcome To Colab - Colab | Untitled0.ipynb - Colab | colab.google | google colab - بحث | ربوت ارم - بحث

https://colab.research.google.com/drive/1qs57p7BX3zGRTTTGcjutgwFcxu1EegfO?authuser=0&hl=en#scrollTo=giz_rXETqskD

Untitled0.ipynb

File Edit View Insert Runtime Tools Help

Share | Gemini

Commands | + Code | + Text | Run all

RAM Disk

Files

- ..
- sample_data
- keras_model.h5
- labels.txt
- test1.jpg

Disk 67.51 GB available

```python
from keras.models import load_model  # TensorFlow is required for Keras to work
from PIL import Image, ImageOps  # Install pillow instead of PIL
import numpy as np

# Disable scientific notation for clarity
np.set_printoptions(suppress=True)

# Load the model
model = load_model("keras_model.h5", compile=False)

# Load the labels
class_names = open("labels.txt", "r").readlines()

# Create the array of the right shape to feed into the keras model
# The 'length' or number of images you can put into the array is
# determined by the first position in the shape tuple, in this case 1
data = np.ndarray(shape=(1, 224, 224, 3), dtype=np.float32)

# Replace this with the path to your image
image = Image.open("test1.jpg").convert("RGB")

# resizing the image to be at least 224x224 and then cropping from the center
size = (224, 224)
image = ImageOps.fit(image, size, Image.Resampling.LANCZOS)

# turn the image into a numpy array
image_array = np.asarray(image)

# Normalize the image
normalized_image_array = (image_array.astype(np.float32) / 127.5) - 1

# Load the image into the array
data[0] = normalized_image_array

# Predicts the model
prediction = model.predict(data)
index = np.argmax(prediction)
class_name = class_names[index]
confidence_score = prediction[0][index]

# Print prediction and confidence score
print("Class:", class_name[2:], end="")
print("Confidence Score:", confidence_score)
```

```
1/1 [==============================] - 1s 1s/step
Class: Humanoid Robot
Confidence Score: 0.9883116
```

Variables | Terminal

7:47 PM | Python 3