Commands + Code + Text ▷ Run all

Files

- ..
- sample_data
- R.jpg
- converted_keras.zip
- g.jpg
- keras_model[2].h5
- labels[1].txt

Disk 66.80 GB available

```
np.set_printoptions(suppress=True)

model = load_model("keras_model[2].h5", compile=False)

class_names = ["Healthy Food", "Unhealthy Food"]

data = np.ndarray(shape=(1, 224, 224, 3), dtype=np.float32)

image = Image.open("R.jpg").convert("RGB")

size = (224, 224)
image = ImageOps.fit(image, size, Image.Resampling.LANCZOS)

image_array = np.asarray(image)

normalized_image_array = (image_array.astype(np.float32) / 127.5) - 1

data[0] = normalized_image_array

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

print("🍽️ Predicted class:", class_name)
print("🔍: ", f"{confidence_score * 100:.2f}%")
```

```
1/1 [==============================] - 3s 3s/step
🍽️ Predicted class: Healthy Food
🔍:  99.87%
```

Commands + Code + Text ▷ Run all

Files

- ..
- sample_data
- R.jpg
- converted_keras.zip
- g.jpg
- keras_model[2].h5
- labels[1].txt

Disk 66.80 GB available

```
np.set_printoptions(suppress=True)

model = load_model("keras_model[2].h5", compile=False)

class_names = ["Healthy Food", "Unhealthy Food"]

data = np.ndarray(shape=(1, 224, 224, 3), dtype=np.float32)

image = Image.open("g.jpg").convert("RGB")

size = (224, 224)
image = ImageOps.fit(image, size, Image.Resampling.LANCZOS)

image_array = np.asarray(image)

normalized_image_array = (image_array.astype(np.float32) / 127.5) - 1

data[0] = normalized_image_array

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

print("📸 Predicted class:", class_name)
print("🔍: ", f"{confidence_score * 100:.2f}%")
```

```
1/1 [==============================] - 4s 4s/step
📸 Predicted class: Unhealthy Food
🔍:  97.20%
```