

Clasificación de imágenes con TensorFlow utilizando InceptionResNetV2

Imagen para clasificar



Código para clasificar imágenes

[crayon-6a9d57c70fdc1298261069/]

← → ↻ https://colab.research.google.com/drive/15S8uDOE54TAoo_oOieZzmQayji4gHULK?hl=es-419 ☆ 🗄️ | 📄

Untitled3.ipynb ☆

Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda [Se guardaron todos los cambios](#) Comentar

Archivos

🔍 🗄️ ↻ 🗑️ 🔒

{x}

🔗

📁

Disco 83.00 GB disponibles

..
sample_data
README.md
anscombe.json
california_housing_test.csv
california_housing_train.csv
mnist_test.csv
mnist_train_small.csv
mujer.jpeg
panda.jpeg

+ Código + Texto

14 s

```
def classify_image(image_path):  
    # Carga del modelo InceptionResNetV2 pre-entrenado  
    model = tf.keras.applications.InceptionResNetV2(include_top=True, weights='imagenet')  
  
    # Preprocesamiento de la imagen  
    img = preprocess_image(image_path)  
    img = np.expand_dims(img, axis=0)  
  
    # Clasificación de la imagen  
    predictions = model.predict(img)  
    predicted_classes = tf.keras.applications.inception_resnet_v2.decode_predictions(predictions,  
  
    return predicted_classes  
  
def main():  
    # Ruta de la imagen a clasificar  
    image_path = '/content/sample_data/panda.jpeg'  
  
    # Clasificar la imagen  
    predicted_classes = classify_image(image_path)  
    print("Predictions for", image_path)  
    for i, (imagenet_id, label, score) in enumerate(predicted_classes):  
        print("{}: {:.2f}%".format(i+1, label, score*100))  
  
if __name__ == "__main__":  
    main()
```

1/1 [=====] - 4s 4s/step
Predictions for /content/sample_data/panda.jpeg
1. giant_panda: 82.51%
2. indri: 1.54%
3. lesser_panda: 0.25%
4. earthstar: 0.22%
5. Lakeland_terrier: 0.10%

✓ 14 s se ejecutó 6:32 p.m.