



TECNOLÓGICO
NACIONAL DE MÉXICO

TECNOLÓGICO NACIONAL DE MÉXICO, TECNOLOGICO DE CULIACAN



**Carrera:
Ingeniería en Sistemas Computacionales**

**Inteligencia Artificial
11:00 – 12:00**

Imagenes para Clasificación de Emociones

**Nombres:
Osuna Russell Ana Isabel
Rodríguez Valerio Jesús Ricardo**

**Docente:
Zuriel Dathan Mora Félix**

22 de Mayo de 2025

1. Dataset:

<https://www.kaggle.com/datasets/shuvoalok/raf-db-dataset>

(Ejemplo de cada imagen)

'1': 'sorprendido',
'2': 'miedo',
'3': 'disgustado', '
'4': 'feliz',
'5': 'triste',
'6': 'enojado',
'7': 'serio'

2. Preprocesamiento de imágenes (Descripción):

Este código se diseñó para poder **preprocesar imágenes faciales** siendo función principal organizar **y preparar un conjunto de imágenes** clasificadas por emociones (como feliz, triste, enojado, etc.), asegurando que todas tengan un **tamaño uniforme** y estén listas para su uso en un modelo de aprendizaje automático.

Recurrimos a un dataset (link arriba), estructurado en carpetas, donde cada carpeta representa una emoción específica y contiene múltiples imágenes de rostros que expresan dicha emoción. Cada imagen es cargada, redimensionada a un tamaño estándar (250x250 píxeles), y luego guardada en una nueva ubicación con una estructura organizada y nombres de archivo que reflejan el procesamiento realizado.

Este paso de preprocesamiento asegura que los datos de entrada al modelo sean aptos para un futuro entrenamiento y la precisión del reconocimiento emocional.

Código:

```
import cv2
import os
import numpy as np

ruta_base_dataset = '/home/ricardebian/Documents/ia/fotos
rec/faces/DATASET/train'

carpetas_emociones = {
```

```

    '1': 'sorprendido',
    '2': 'miedo',
    '3': 'disgustado',
    '4': 'feliz',
    '5': 'triste',
    '6': 'enojado',
    '7': 'neutro'
}

tamaño_objetivo = (250, 250)

ruta_base_salida = './resultado'

def procesar_imagen(ruta_imagen, tamaño_obj):
    try:
        imagen = cv2.imread(ruta_imagen)
        if imagen is None:
            print(f"Error: No se pudo cargar la imagen {ruta_imagen}")
            return None

        imagen_redimensionada = cv2.resize(imagen, tamaño_obj,
interpolation=cv2.INTER_AREA)

        return imagen_redimensionada

    except Exception as e:
        print(f"Error procesando la imagen {ruta_imagen}: {e}")
        return None

if not os.path.exists(ruta_base_salida):
    os.makedirs(ruta_base_salida)

for num_carpeta, nombre_emocion in carpetas_emociones.items():
    ruta_emocion_entrada = os.path.join(ruta_base_dataset, num_carpeta)
    ruta_emocion_salida = os.path.join(ruta_base_salida, num_carpeta +
"_" + nombre_emocion)

    if not os.path.exists(ruta_emocion_salida):
        os.makedirs(ruta_emocion_salida)

    if not os.path.isdir(ruta_emocion_entrada):
        print(f"Advertencia: La carpeta {ruta_emocion_entrada} no
existe. Saltando.")

```

```

        continue

    print(f"Procesando carpeta: {nombre_emocion}
    ({ruta_emocion_entrada})")
    archivos_imagenes = [f for f in os.listdir(ruta_emocion_entrada) if
f.lower().endswith(('.png', '.jpg', '.jpeg'))]

    for nombre_imagen in archivos_imagenes:
        ruta_completa_imagen = os.path.join(ruta_emocion_entrada,
nombre_imagen)

        imagen_procesada = procesar_imagen(ruta_completa_imagen,
tamaño_objetivo)

        if imagen_procesada is not None:
            nombre_imagen_salida = "proc_" + nombre_imagen
            ruta_completa_imagen_salida =
os.path.join(ruta_emocion_salida, nombre_imagen_salida)
            try:
                cv2.imwrite(ruta_completa_imagen_salida,
imagen_procesada)
            except Exception as e:
                print(f"Error guardando la imagen
{ruta_completa_imagen_salida}: {e}")

    print("Procesamiento completado.")

```

3. Resultados:

[Link de imágenes ya procesadas](#) (En OneDrive)