Instituto tecnológico de culiacán

Ingeniería en sistemas computacionales

Módulo 4

Dataset para el modelo de visión artificial

Nombre:
- De La Rocha Linarez Diego Alejandro
- Roriguez Cazarez Joaquín

No. De control:
- 21170302
- 21170458

Materia:
Inteligencia artificial
08-09 hrs

Maestro:
Dr. Zuriel Dathan Mora Felix

Semestre: 8

24 de mayo del 2025

Dataset para el reconocimiento de emociones.

Este dataset contiene las emociones solicitadas para la tarea con el preprocesamiento solicitado. El dataset fue revisado para asegurarnos de que no hubiera fotografías raras
link: https://universe.roboflow.com/uemc-y7rsy/emociones/dataset/2

Para el preprocesamiento de las imágenes pudimos haber usado la libreria opencv para la rotación, cambio de intensidad y escala

```
# Cargar imagen
img = cv2.imread("ruta_de_tu_imagen.jpg")

# Ajustar brillo y contraste (alpha: contraste, beta: brillo)
alpha = 1.5  # >1 aumenta contraste, <1 lo reduce
beta = 30    # >0 aumenta brillo, <0 lo reduce
adjusted_img = cv2.convertScaleAbs(img, alpha=alpha, beta=beta)

# Guardar resultado
cv2.imwrite("imagen_ajustada.jpg", adjusted_img)
```

con la función convertScaleAbs se pueden cambiar los parámetros de contraste y brillo y guardarse la imagen modificada

```
height, width = img.shape[:2]
center = (width // 2, height // 2)
angle = 45
scale = 1.0  # Mantener escala original

# Generar matriz de rotación
rotation_matrix = cv2.getRotationMatrix2D(center, angle, scale)

# Aplicar rotación (con borde blanco para áreas vacías)
rotated_img = cv2.warpAffine(img, rotation_matrix, (width, height), borderValue=(255, 255, 255))

cv2.imwrite("imagen_rotada.jpg", rotated_img)
```

Con la combinación de getRotatioMatrix2D para rotar y warpAffine para rellenar los espacios vacíos se puede lograr una correcta rotación de las imágenes.

```
new_width = int(width * 0.5)
new_height = int(height * 0.5)
resized_img = cv2.resize(img, (new_width, new_height), interpolation=cv2.INTER_LINEAR)

cv2.imwrite("imagen_escalada.jpg", resized_img)
```

Y con el metodo resize se puede cambiar la escala de las fotografías.

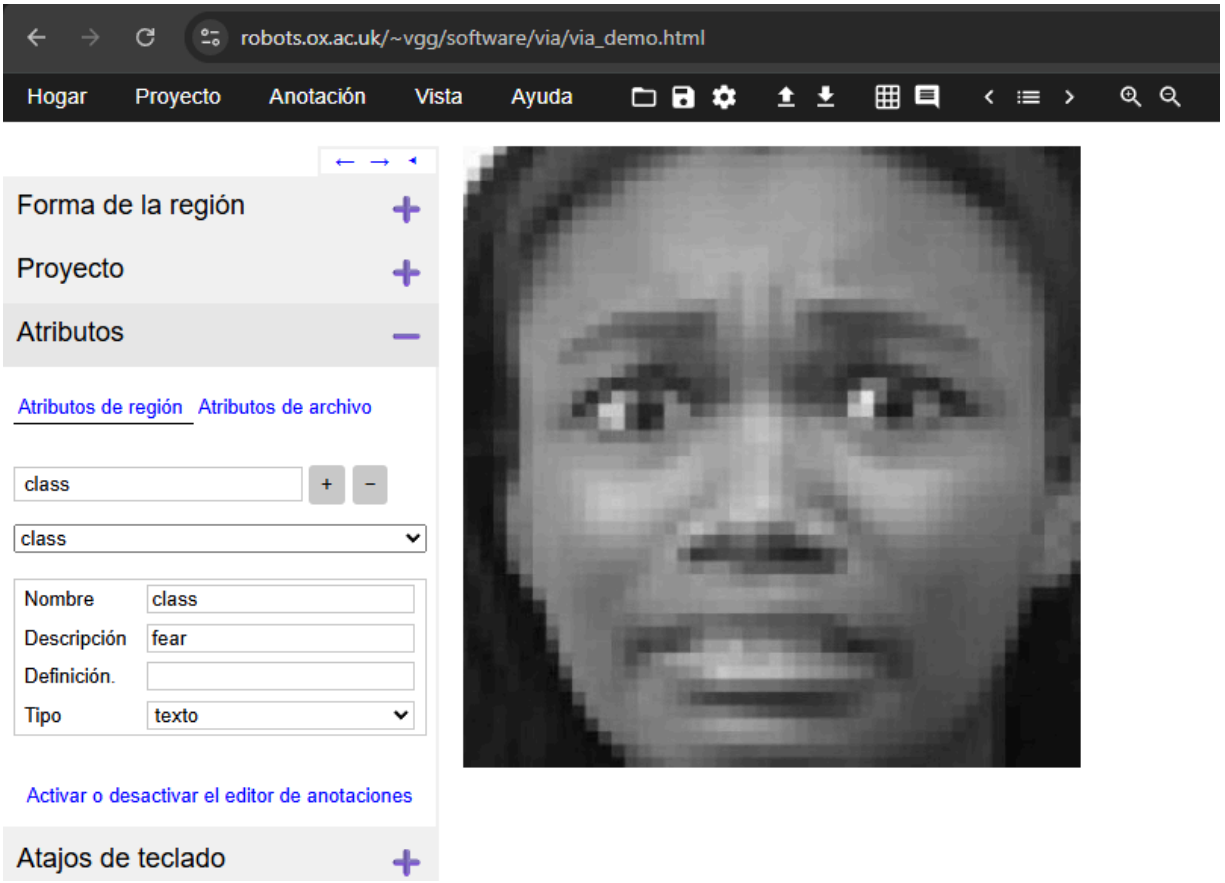
Para el etiquetado de las imágenes se utiliza el formato COCO JSON.

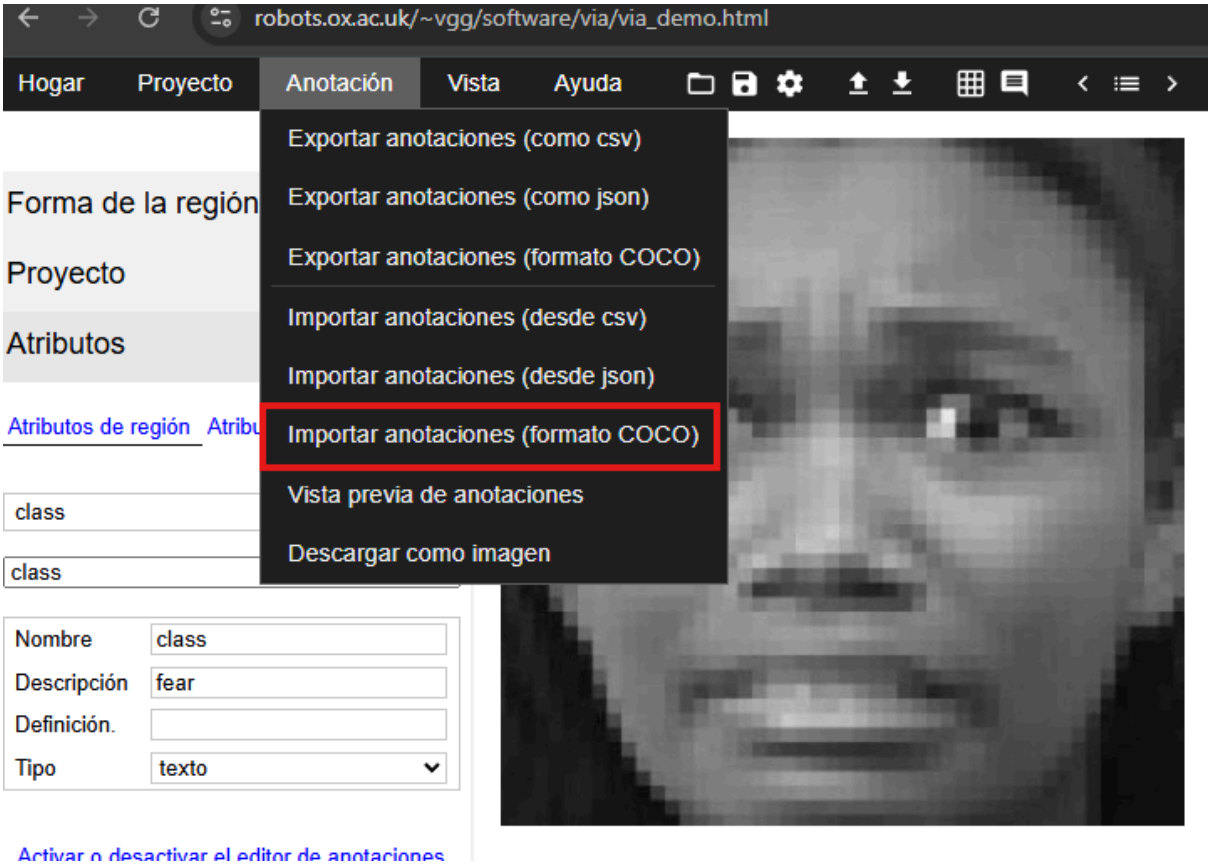
El formato COCO es ampliamente utilizado en tareas de visión por computadora, como detección de objetos y segmentación. Su estructura se basa en un archivo JSON que contiene información sobre las imágenes, las anotaciones y las categorías.

El archivo JSON en formato COCO incluye las siguientes secciones principales:

- **info**: Información general sobre el dataset (nombre, versión, descripción, etc.).
- **licenses**: Detalles sobre las licencias de las imágenes.
- **images**: Lista de imágenes con atributos como `id`, `file_name`, `width` y `height`.
- **annotations**: Anotaciones para cada imagen, incluyendo `id`, `image_id`, `category_id`, `bbox`, `area`, `iscrowd`, entre otros.
- **categories**: Definición de las categorías presentes en el dataset, cada una con un `id` y un `name`.

Este tipo de archivos se pueden generar con herramientas como el VGG Image Annotator (VIA) de la universidad de oxford




robots.ox.ac.uk/~vgg/software/via/via_demo.html
Hogar
Proyecto
Anotación
Vista
Ayuda
Exportar anotaciones (como csv)
Exportar anotaciones (como json)
Exportar anotaciones (formato COCO)
Importar anotaciones (desde csv)
Importar anotaciones (desde json)
Importar anotaciones (formato COCO)
Vista previa de anotaciones
Descargar como imagen
Forma de la región
Proyecto
Atributos
Atributos de región
class
class
Nombre
class
Descripción
fear
Definición.
Tipo
texto