



# INSTITUTO TECNOLÓGICO DE CULIACÁN

*Ingeniería en Sistemas Computacionales*

**Inteligencia artificial**

**Hora: 09:00 – 10:00 AM**

**Tarea:**

Dataset de imágenes y preprocesamiento

**Integrantes:**

Peña López Miguel Ángel

Robles Rios Jacquelin

**Docente:**

Zuriel Dathan Mora Felix

**Culiacán, Sinaloa**

19/05/2025

En la visión por computadora, un dataset es un conjunto de imágenes etiquetadas que sirve como base para entrenar, validar y probar modelos de reconocimiento.

Un buen dataset debe tener:

- Variedad de clases: múltiples categorías que el modelo debe distinguir.
- Diversidad dentro de cada clase: diferentes ángulos, iluminaciones, escalas, etc.
- Etiquetas precisas: cada imagen debe estar correctamente identificada para que el modelo aprenda de forma efectiva.

En este caso elegimos un dataset desde la plataforma roboflow el cual contiene 7 clases (emociones) distintas.

Dataset: <https://universe.roboflow.com/my-workspace-2-9rl26/facial-emotion-detection-rijqy/dataset/2>

Además dentro de la misma plataforma creamos otra “versión” del dataset con un preprocesamiento previo que se configura en la plataforma:

The screenshot displays the 'Create New Version' interface in Roboflow. It shows a progress bar with five steps: 1. Source Images (completed), 2. Train/Test Split (active), 3. Preprocessing, 4. Augmentation, and 5. Create. The 'Train/Test Split' step is detailed with a table showing the distribution of images into three sets: Train Set (852 Images, 78%), Valid Set (243 Images, 28%), and Test Set (122 Images, 14%). A 'Continue' button is visible below the table, and a 'Rebalance' button is located at the bottom right of the configuration area.

| Set       | Images     | Percentage |
|-----------|------------|------------|
| TRAIN SET | 852 Images | 78%        |
| VALID SET | 243 Images | 28%        |
| TEST SET  | 122 Images | 14%        |

REGUELLOPEZ




Image Classi...  
Classification

DATA

↑ Upload Data

📄 Annotate

📁 Dataset 1217

**🔍 Versions** Train

📊 Analytics

🏷️ Classes & Tags

MODELS

📈 Models

📊 Visualize

DEPLOY

👉 Deployments

📁 Versions

No versions created yet.

Create New Version

Prepare your images and data for training by compiling them into a version.  
Experiment with different configurations to achieve better training results.

✓ Source Images

Images: 1217  
Classes: 21

✓ Train/Test Split

Training Set: 852 images  
Validation Set: 243 images  
Testing Set: 122 images

3 Preprocessing

📖 What can preprocessing do?  
Decrease training time and increase performance by applying image transformations to all images in this dataset.

Auto-Orient

Edit

×

Resize

Stretch to 640x640

Edit

×

Auto-Adjust Contrast

Using Histogram Equalization

Edit

×

+

Add Preprocessing Step

Continue

4 Augmentation

Create new training examples for your model to learn from by generating augmented versions of each image in your training set.

Flip

Horizontal

Edit

×

Rotation

Between -15° and +15°

Edit

×

Brightness

Between -15% and +15%

Edit

×

Blur

Up to 0.8px

Edit

×

+

Add Augmentation Step

Continue

Clear All

Posteriormente usando Python descargamos el dataset para su debida división en conjunto de entrenamiento o de validación, para así pasar al procesamiento de las imágenes de entrenamiento.

El preprocesamiento de imágenes es una etapa fundamental en proyectos de visión por computadora. Consiste en aplicar técnicas que mejoran o modifican las imágenes antes de ser utilizadas por un modelo de aprendizaje automático o red neuronal. Esto ayuda a reducir ruido y normalizar condiciones de captura.

Para Nuestro dataset realizamos las siguientes transformaciones utilizando Python

- Brillo (`brightness`): Ajusta la intensidad de luz para simular diferentes condiciones de iluminación.
- Rotación (`rotation`): Simula distintas orientaciones de los rostros en la imagen.
- Escalado (`scale`): Cambia el tamaño para mejorar la capacidad del modelo de reconocer objetos en diferentes proporciones.

Además tanto las imágenes originales como las transformadas fueron guardadas para así poder verificar aquello que haya cambiado, sin embargo esto no es lo mejor debido al espacio y rapidez que necesita. En caso de querer entrenar algún modelo lo ideal sería usar ImageDataGenerator ya que aplica que las transformaciones sean en tiempo real.