



**TECNOLÓGICO  
NACIONAL DE MÉXICO**



## INTELIGENCIA ARTIFICIAL

### UNIDAD 4 – TAREA 1

#### **NOMBRE:**

SARABIA GUZMÁN JESÚS ALDHAIR  
ZAMUDIO LIZÁRRAGA BRYAN MARTÍN

#### **NUMERO DE CONTROL:**

22170824

22170855

#### **FECHA:**

24/05/2025

#### **MAESTRO:**

ZURIEL DATHAN MORA FÉLIX

## Adquisición de Imágenes

Para realizar este trabajo consideramos varias opciones, principalmente **FER-2013**. Sin embargo, la resolución de esta es bastante baja (48x48) y se pueden perder detalles más finos en las expresiones. Como una posible solución a esto, utilizamos una herramienta llamada Upscayl para aumentar la resolución de imágenes en masa de manera local; a continuación, podemos ver una comparación entre una imagen original y otra upscaled a 96x96:



Como podemos observar, esto aumenta el nivel de detalle considerablemente, pero aún así quisiéramos mantener la opción de utilizar el dataset original con el propósito de comparar los resultados.

De este dataset tomaremos 2000 imágenes de cada expresión, para no tener más imágenes de una expresión que de otra y así no presente un problema después del entrenamiento.

Como alternativa, también consideramos usar **UIBAIFED**, este es un conjunto de imágenes con las expresiones requeridas, a una resolución bastante más alta (512x512). La tomamos en cuenta ya que, al ser de mucha más resolución, podemos observar los detalles más finos de cada cara. Sin embargo, al ser de una resolución mucho más alta que FER-2013, podría tomar considerablemente más tiempo pre-procesar y entrenar al modelo usando estas imágenes, además de que hay menos imágenes de cada expresión (~400, la más baja siendo sorpresa con 146).

## Pre-procesamiento

Para realizar el pre procesamiento, hicimos un algoritmo en Python usando la función de ImageDataGenerator de TensorFlow. Mediante esta, realizamos varios cambios a cada imagen: reescalamos el valor de cada píxel a un valor entre 0 y 1 para normalizar, rotamos en un rango de -20° a 20°, aplicamos un zoom de -20% a 20%, cambiamos el brillo en un rango de 60% a 140% y volteamos horizontalmente la imagen. Los cambios anteriores que se realizarán a cada imagen serán al azar usando los parámetros mencionados, y se harán en tiempo real.

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator

ruta = r"IA\Expresiones\Face Recognition Images\FER2013\train"

# Definimos el generador
datagen = ImageDataGenerator(
    rescale=1./255,          # Reescalado de los valores de cada
    rotation_range=20,       # Rotación de -20° a 20°
    zoom_range=0.2,          # Zoom de -20% a 20%
    brightness_range=[0.6, 1.4], # Cambio de brillo de 60% a 140%
    horizontal_flip=True,    # Volteo horizontal
    fill_mode='nearest'      # Relleno de píxeles tomando el más
cercano
)

# Creamos el generador
train_generator = datagen.flow_from_directory(
    ruta,
    target_size=(96, 96),
    batch_size=32,
    class_mode='categorical' # Etiquetamos cada imagen según la
carpeta en la que esté
)
```

Ejemplo de las imágenes tras el pre-procesado:



En conclusión, al hacer el modelo lo entrenaremos con FER-2013 original, si la precisión del modelo es baja, utilizaremos las imágenes upscaleadas, y si sigue siendo baja usaremos UIBAIFED.

## **Herramientas utilizadas**

**Upscayl:** <https://github.com/upscayl/upscayl>

**FER-2013:** <https://www.kaggle.com/datasets/msambare/fer2013/data>

**UIBAIFED:** <https://zenodo.org/records/15181380>