



TECNOLÓGICO  
NACIONAL DE MÉXICO



**Instituto Tecnológico de Culiacán  
Campus Culiacán**

**Inteligencia Artificial**

**Preprocesamiento de Dataset**

**Docente:**

**Zuriel Dathan Mora Felix**

**Equipo:**

**-Yosef Emiliano Soto Iribe**

**-Cristian Andrea Ramirez Medina**

La intención de este proyecto es desarrollar un programa el cual por medio de un modelo clasifique emociones, esto lo queremos lograr con una aplicación hecha en python que utiliza la webcam y de esa manera detectar la emoción que muestre la persona en ese momento en vivo.

El primer paso para desarrollar este proyecto es conseguir imágenes con las cuales se va a entrenar el modelo, para ello conseguimos un dataset en la página de Kaggle llamado RAF-DB DATASET, el cual se compone de distintas carpetas separadas por emociones y que dentro de cada carpeta contiene miles de fotos representativas a dicha emoción.

Link de la fuente donde se consiguió el dataset:  
<https://www.kaggle.com/datasets/shuvoalok/raf-db-dataset>

Existe un paso que sigue después de haber conseguido un dataset con imágenes el cual es el preprocesamiento de dichas imágenes, la importancia del preprocesamiento radica en fortalecer y hacer más preciso el modelo que vamos a entrenar, un ejemplo de preprocesamiento de las imágenes es realizar cambios de intensidad de luz, rotaciones, cambios de escala etc.

Para ello se implementó un programa en python que pudiera hacer eso automáticamente.

Primeramente lo que se hizo fue acceder al directorio donde se encuentra el dataset para así poder obtener todas las rutas en donde se encuentran las imágenes e ir las guardando en nuestra lista "todasImágenes" y así poderlas procesar más adelante.

```
# Directorio base de entrada
carpeta_base = "DATASET/train"
carpeta_salida = "DATASET/train_aumentado"
os.makedirs(carpeta_salida, exist_ok=True)

# Cargar todas las rutas de imágenes y sus etiquetas
todasImágenes = []
for emocion in os.listdir(carpeta_base):
    carpeta_emocion = os.path.join(carpeta_base, emocion)
    if os.path.isdir(carpeta_emocion):
        for nombre_img in os.listdir(carpeta_emocion):
            ruta_img = os.path.join(carpeta_emocion, nombre_img)
            todasImágenes.append((ruta_img, emocion))
```

Para el preprocesamiento de las imágenes se optó por implementar dos funciones que nos ayudaran al proceso, las cuales son `ajusteBrillo` y `rotarImagen`, estas nos ayudaran a el ajuste del brillo de las imágenes multiplicando los valores de los píxeles por un valor y a rotar las imágenes dado un ángulo en específico.

```
def ajusteBrillo(img, factor):
    return np.clip(img * factor, 0, 255).astype(np.uint8)

def rotarImagen(img, angulo):
    (h, w) = img.shape[:2]
    centro = (w // 2, h // 2)
    M = cv2.getRotationMatrix2D(centro, angulo, 1.0)
    giro = cv2.warpAffine(img, M, (w, h), flags=cv2.INTER_LINEAR,
borderMode=cv2.BORDER_REPLICATE)
    return giro
```

Ahora leemos las imágenes originales del dataset para así poder procesarlas y aplicar el ajuste de brillo y las rotaciones para así guardar las imágenes procesadas en un nuevo directorio, también se crea un archivo csv en el que por cada imagen establece que tipo de emoción es y qué tipo de procesamiento se le aplicó.

```
with open("metadata.csv", "w", newline='') as f:
    writer = csv.writer(f)
    writer.writerow(["nombre_archivo", "etiqueta_emocion",
"tipo_transformacion"])

    for rutaImg, emocion in random.sample(todasImagenes, min(400,
len(todasImagenes))):
        img = cv2.imread(rutaImg, cv2.IMREAD_GRAYSCALE)
        if img is None:
            continue

        img_name = os.path.basename(rutaImg)

        transformaciones = [
            (img, f"original_{img_name}", "original"),
            (ajusteBrillo(img, 0.6), f"dark_{img_name}",
"brillo_reducido"),
```

```

        (ajusteBrillo(img, 1.4), f"bright_{img_name}",
"brillo_aumentado"),
        (rotarImagen(img, 15), f"rot15_{img_name}", "rotacion_15"),
        (rotarImagen(img, -30), f"rotneg30_{img_name}", "rotacion_-30")
    ]

    for imgTransformado, nombre, tipo in transformaciones:
        carpeta_emocion_salida = os.path.join(carpeta_salida, emocion)
        os.makedirs(carpeta_emocion_salida, exist_ok=True)
        rutaSalida = os.path.join(carpeta_emocion_salida, nombre)
        cv2.imwrite(rutaSalida, imgTransformado)
        writer.writerow([nombre, emocion, tipo])

```

Link en drive del dataset:  
[https://drive.google.com/file/d/1N\\_VX9S28EGkYNLDS\\_5lCrWzi4BXvbp7s/view?usp=sharing](https://drive.google.com/file/d/1N_VX9S28EGkYNLDS_5lCrWzi4BXvbp7s/view?usp=sharing)