



**TECNOLÓGICO  
NACIONAL DE MÉXICO**



**Integrantes:**

Edgar Ochoa

Aviles Alonso

Castro Vega

**Carrera:**

Ingeniería En Sistemas Computacionales

**Materia:**

Inteligencia Artificial

**Tarea:**

Tarea 1 Modulo 4 Manual

**Profesor:**

ZURIEL DATHAN MORA FELIX

**Grupo:**

11:00 a 12:00 a.m

## 1. Requisitos Previos

- **Entorno de Ejecución:**
  - Opción 1: Google Colab (recomendado para GPU gratuita).
  - Opción 2: Entorno local con Python 3.8+ y 12 GB de RAM mínimo.
- **Dependencias:**
- bash
- Copy
- Download
- `!pip install numpy pandas opencv-python tensorflow albumentations matplotlib scikit-learn`
- **Dataset:**
  - Archivo ZIP estructurado con:
    - markdown
    - Copy
    - Download

```
dataset/
├── subcarpeta1/
│   ├── imagen1.jpg
│   └── metadata.csv (con columnas: "filename", "emotion")
└── subcarpeta2/
    └── imagen2.png
```

○ | └── metadata.csv

---

## 2. Ejecución Paso a Paso

### 2.1. Carga Inicial en Google Colab

1. **Habilitar GPU:**
  - Ir a Entorno de ejecución > Cambiar tipo de entorno de ejecución > GPU.
2. **Ejecutar Configuración Inicial:**
3. python
4. Copy
5. Download
6. # Ejecutar todas las celdas hasta la función main()  
El programa detectará automáticamente si está en Colab.

### 2.2. Subir Dataset

- **En Google Colab:**
  - Al ejecutar main(), se mostrará un botón para subir el ZIP.
  - Seleccionar el archivo ZIP que contiene las imágenes y CSVs.



- **En Local:**

- Proporcionar la ruta absoluta del ZIP cuando se solicite:
- python
- Copy
- Download
- Ruta del archivo ZIP: /ruta/al/dataset.zip

### 2.3. Procesamiento Automático

El programa realizará:

1. **Extracción del ZIP** en ./dataset.
2. **Validación de Datos:**
  - Chequeo de correspondencia imagen-CSV.
  - Filtrado de imágenes corruptas (se muestran en consola).
3. **División del Dataset:**
  - 80% entrenamiento, 10% validación, 10% prueba (estratificado).
4. **Data Augmentation:**
  - Aumenta el dataset original x3 usando transformaciones aleatorias.

---

### 3. Parámetros Personalizables

Modificar antes de ejecutar main():

python

Copy

Download

# Configuración Global (buscar en el código)

---

### 4. Salidas y Resultados

#### 4.1. Visualizaciones

- **Muestras del Dataset:**
  - Se mostrarán 12 imágenes aleatorias con sus etiquetas.  

- **Verificación Post-Aumentación:**
  - Gráficos de distribución de clases y muestras aumentadas.

#### 4.2. Datos para Entrenamiento

Acceder a los datos procesados:

python

Copy

Download

```
# Retorna diccionario con:  
# Datos de entrenamiento (aumentados)  
# Validación  
# Prueba  
# Codificador de etiquetas
```

---

## 5. Gestión de Memoria (Google Colab)

- **Monitorear RAM:**
- python
- Copy
- Download
- !free -h # Ver uso de memoria
- **Si falla por memoria:**
  1. Reducir AUGMENT\_FACTOR (ej. de 3 a 2).
  2. Limpiar variables innecesarias:
    3. python
    4. Copy
    5. Download

del X\_train, y\_train

```
6. import gc; gc.collect()
```

---

## 6. Customización Avanzada

### 6.1. Añadir Nuevas Emociones

1. Modificar la lista EMOTIONS:
2. python
3. Copy
4. Download
5. EMOTIONS = ['surprise', 'disgust', ...] # Máximo 8 clases para 12 GB RAM
6. Asegurar que los CSVs usen estas mismas etiquetas.

### 6.2. Modificar Aumentación

Editar create\_augmentation\_pipeline():

python

Copy

Download

```
def create_augmentation_pipeline():  
    return A.Compose([  
        A.RandomRain(p=0.3),    # Añadir lluvia  
        A.ColorJitter(p=0.5),   # Modificar saturación  
        # ... otras transformaciones  
    ])
```

---

## 7. Solución de Errores Comunes

Error	Solución
FileNotFoundError	Verificar rutas en CSVs y nombres de archivos.
RAM agotada	Reducir IMG_SIZE o AUGMENT_FACTOR.
Discrepancia etiquetas-imágenes	Revisar que la columna emotion en CSVs coincida con EMOTIONS.
Formato de imagen no soportado	Convertir imágenes a JPG/PNG con PIL.Image.open().convert('RGB').

---

## 8. Ejemplo de Uso para Entrenamiento

python

Copy

Download

```
from tensorflow.keras.models import Sequential  
from tensorflow.keras.layers import Dense, Conv2D, Flatten  
  
# Crear modelo simple
```



```
print(" - result['label']  
else:  
    print("❌ El procesamiento")
```



🔍 Verificando integridad de Prue

Total de imágenes: 276

Total de etiquetas: 276

Distribución:

angry: 61 imágenes

fear: 21 imágenes

happy: 66 imágenes

neutral: 72 imágenes

sad: 56 imágenes

angry  
(código: 0)



Paso 8: Aplicando data augmentat

Método de augmentación: albumenta



Aplicando augmentación (factor

Generadas 0/4408 imágenes aume

Generadas 100/4408 imágenes au

Variables



Terminal