



**TECNOLÓGICO
NACIONAL DE MÉXICO**

TECNM

Tecnológico Nacional de México

Campus Culiacán

Ingeniería en Sistemas computacionales

Inteligencia artificial

09:00 – 10:00

Tarea 1 Unidad 4 – Pre-procesado de imágenes

Integrantes:

Caro García Jorge Ariel

Galván González Sebastián

Docente:

ZURIEL DATHAN MORA FELIX

24/05/2025

Desprecio

1/1  0s 111ms/step

anger: 16.84%

contempt: 25.39%

disgust: 10.64%

fear: 14.53%

happy: 8.40%

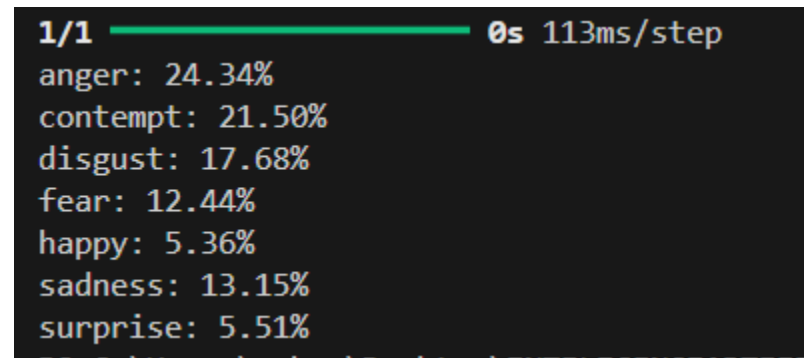
sadness: 14.04%

surprise: 10.15%


□



Enojo



Disgusto

1/1  **0s** 115ms/step
anger: 17.38%
contempt: 24.05%
disgust: 11.77%
fear: 14.01%
happy: 7.82%
sadness: 14.97%
surprise: 10.00%



Miedo

1/1 — 0s 110ms/step

anger: 8.88%

contempt: 16.07%

disgust: 7.80%

fear: 40.73%

happy: 8.61%

sadness: 8.74%

surprise: 9.16%



Felicidad

1/1  0s 115ms/step

anger: 16.20%
contempt: 23.57%
disgust: 11.89%
fear: 16.26%
happy: 9.22%
sadness: 13.33%
surprise: 9.53%



Tristeza

1/1  0s 110ms/step

anger: 17.28%

contempt: 23.34%

disgust: 11.01%

fear: 15.56%

happy: 8.12%

sadness: 14.01%

surprise: 10.69%

□



Sorpresa

1/1 0s 111ms/step
anger: 2.90%
contempt: 6.22%
disgust: 6.94%
fear: 10.36%
happy: 2.29%
sadness: 5.24%
surprise: 66.05%



Librerías

```
import tensorflow as tf
from keras import layers, models
import os
import numpy as np
import cv2
import random
```

- **TensorFlow/Keras:** Para construir y entrenar la red neuronal.
- **os:** Para manejar archivos y carpetas.
- **numpy:** Para manipulación de arrays.
- **cv2:** OpenCV, para leer y procesar imágenes.

Definición de parámetros y rutas

```
width = 48
height = 48

ruta_train = 'archive/ck/CK+48/'
ruta_predict = 'archive/CK+48/surprise/S010_002_00000012.png'
```

- width, height: Tamaño al que se redimensionarán las imágenes.
- ruta_train: Carpeta donde están las imágenes de entrenamiento, organizadas por carpetas de emociones.
- ruta_predict: Ruta de una imagen para predecir su emoción.

Preparando datos de entrenamiento

```
train_x = []
train_y = []

labels = os.listdir(ruta_train)
for i in os.listdir(ruta_train):
    for j in os.listdir(ruta_train + i):
        img = cv2.imread(ruta_train+i+'/'+j)
        resized_image = cv2.resize(img, (width, height))
        train_x.append(resized_image)
        for x,y in enumerate(labels):
            if y == i:
                array = np.zeros(len(labels))
                array[x] = 1
                train_y.append(array)
```

- **labels:** Lista de nombres de las carpetas (cada una representa una emoción).
- **Bucle:** Recorre cada carpeta y cada imagen dentro de ella.
 - Lee la imagen y la redimensiona.
 - La agrega a `train_x`.
 - Crea un vector one-hot para la etiqueta correspondiente y lo agrega a `train_y`.

Conversión a arrays de numpy

```
x_data = np.array(train_x)
y_data = np.array(train_y)
```

- Convierte las listas de imágenes y etiquetas a arrays de numpy, que son el formato que espera Keras.

Definición del modelo CNN

```
model = tf.keras.Sequential([
    layers.Conv2D(64, (3, 3), activation='relu', input_shape=(width, height, 3)),
    layers.MaxPooling2D(pool_size=(2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D(pool_size=(2, 2)),
    layers.Conv2D(128, (3, 3), activation='relu'),
    layers.MaxPooling2D(pool_size=(2, 2)),
    layers.Flatten(),
    layers.Dense(128, activation='relu'),
    layers.Dropout(0.5),
    layers.Dense(7, activation='softmax')
])
```

- Modelo secuencial: Tres capas convolucionales con max pooling, seguidas de una capa densa y una de salida softmax (para clasificación multiclase).
- Dropout: Ayuda a evitar el sobreajuste.
- Dense(7): Asume que hay 7 emociones/clases.

Compilación y entrenamiento

```
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
epochs = 200
model.fit(x_data, y_data, epochs=epochs)
```

- Compilación: Usa categorical_crossentropy para clasificación multiclase y el optimizador Adam.
- Entrenamiento: Ajusta el modelo a los datos durante 200 épocas.

Guardar y cargar modelo

```
model.save('emociones2.keras')
model = models.load_model('emociones2.keras')
```

- Guarda el modelo entrenado y luego lo vuelve a cargar (para usar el modelo sin volver a entrenar).

Predicción sobre una imagen nueva

```
my_image = cv2.imread(ruta_predict)
my_image = cv2.resize(my_image, (width, height))
my_image = my_image / 255.0

result = model.predict(np.array([my_image]))[0]
```

- Lee y redimensiona la imagen a predecir.
- Normaliza los valores de los píxeles a [0,1].
- Predice la emoción usando el modelo.

Resultados

```
for i, label in enumerate(labels):
    porcentaje = result[i] * 100
    print(f"{label}: {porcentaje:.2f}%")

cv2.imshow('Hola', cv2.imread(ruta_predict))
cv2.waitKey(0)
```

- Muestra el porcentaje de probabilidad para cada emoción.
- Muestra la imagen predicha en una ventana.

Dataset utilizado:

<https://www.kaggle.com/datasets/shawon10/ckplus/data>