Maestro: Mauricio Alejandro Cabrera Arellano

Alumno: Nene Miranda José Said 22110319

Materia: Visión Artificial

Tarea: Proyecto

Fecha: 14-06-2025

Proyecto final:

Objetivo general:

Desarrollar un sistema de clasificación de emociones faciales a partir de imágenes usando una red neuronal convolucional (CNN) entrenada con el dataset FER2013.

Objetivos específicos:

- Cargar y procesar el dataset FER2013.

- Construir y entrenar una red neuronal convolucional.

- Evaluar el modelo y mostrar resultados de precisión.

- Documentar el desarrollo del proyecto y su funcionamiento.

Explicación del Proyecto:

Se implementa un sistema que detecta emociones en rostros humanos a partir de imágenes de 48x48 píxeles en escala de grises, usando redes neuronales convolucionales (CNN). Se utilizó el dataset FER2013, disponible en Kaggle, el cual contiene miles de rostros etiquetados con emociones como felicidad, tristeza, enojo, miedo, etc.

El flujo de trabajo incluyó:

- Limpieza de datos.

- Preprocesamiento (normalización y formato).

- Construcción de una red CNN en Keras.

- Entrenamiento con 10 épocas.

- Evaluación y análisis de precisión.

Pruebas y Errores:

Durante el desarrollo se identificaron errores como:

- Filas corruptas en el CSV que causaban errores al hacer reshape.

- Variación brusca en la precisión de validación (resuelto con normalización y dropout).

- Disparidad entre precisión de entrenamiento y validación (se controló con aumento de datos y evaluación por épocas).

Los resultados mostraron una tendencia de aprendizaje progresiva, con precisión en validación de alrededor del 35% tras 10 épocas.

Código:

```
# CÓDIGO COMPLETO DE ENTRENAMIENTO CNN CON FER2013
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout
from sklearn.model_selection import train_test_split
data = pd.read_csv('fer2013.csv')
faces = []
labels = []
for i, row in data.iterrows():
    pixel_sequence = row['pixels']
    pixel_values = [int(pixel) for pixel in pixel_sequence.split()]
    if len(pixel_values) == 48*48:
        face = np.asarray(pixel_values).reshape(48, 48)
        faces.append(face.astype('float32'))
        labels.append(row['emotion'])
faces = np.array(faces)
faces = np.expand_dims(faces, -1)
faces = faces / 255.0
labels = pd.get_dummies(labels).values
X_train, X_test, y_train, y_test = train_test_split(faces, labels, test_size=0.2,
random_state=42)
model = Sequential([
```

```
  Conv2D(32, (3,3), activation='relu', input_shape=(48,48,1)),
  MaxPooling2D((2,2)),
  Conv2D(64, (3,3), activation='relu'),
  MaxPooling2D((2,2)),
  Flatten(),
  Dense(128, activation='relu'),
  Dropout(0.5),
  Dense(7, activation='softmax')
])
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
history = model.fit(X_train, y_train, epochs=10, batch_size=64, validation_data=(X_test, y_test))
test_loss, test_acc = model.evaluate(X_test, y_test)
print(f"Precisión: {test_acc:.2f}")
```

Demostración: