



INSTITUTO TECNOLÓGICO DE CULIACÁN

Ingeniería en Sistemas Computacionales

Inteligencia artificial

Hora: 09:00 am – 10:00 am

Tarea:

Sistema de reconocimiento facial de emociones en tiempo real

Equipo:

Peña López Miguel Ángel

Robles Rios Jacquelin

Docente:

Zuriel Dathan Mora Félix

Culiacán, Sinaloa

29/05/2025

En la actualidad, la inteligencia artificial y la visión por computadora han permitido avances significativos, ya que tienen aplicaciones en áreas como educación, salud mental, atención al cliente y seguridad. Este proyecto presenta el desarrollo de un sistema de reconocimiento facial de emociones en tiempo real, capaz de identificar 7 estados emocionales a partir de las expresiones faciales captadas por una cámara web.

El sistema utiliza técnicas modernas de procesamiento de imágenes junto con modelos de aprendizaje profundo, específicamente redes neuronales convolucionales (CNN), para analizar y clasificar las emociones de forma precisa. Además, se incorpora un detector de rostros basado usando Haar Cascades.

¿Qué es una CNN?

Una red neuronal convolucional (CNN), es un tipo especializado de algoritmo de aprendizaje profundo diseñado principalmente para tareas que requieren reconocimiento de objetos, como la clasificación, la detección y la segmentación de imágenes.

Las CNN están inspiradas en el funcionamiento de la corteza visual humana, donde pequeñas regiones del campo visual son procesadas de manera jerárquica para reconocer formas, texturas y objetos.

¿Por qué usar CNN (Redes Neuronales Convolucionales)?

Las Redes Neuronales Convolucionales (CNN) son la mejor opción para el reconocimiento de emociones faciales debido a su capacidad para procesar información visual de manera jerárquica y automática. Las CNN aprenden patrones directamente de los píxeles, lo que las hace ideales para captar los sutiles cambios en expresiones faciales.

Sus ventajas son:

- Extracción automática de características
- Robustez frente a variaciones
- Precisión en clasificación
- Eficiencia en tiempo real

Proceso de desarrollo

1. Recopilación de Datos

Para el entrenamiento del modelo se utilizó un conjunto de datos obtenido desde la plataforma Roboflow, la cual proporciona datasets especializados para tareas de visión por computadora. La descarga se realizó a través de su API, seleccionando un dataset de clasificación de emociones faciales. El conjunto de datos descargado estaba estructurado en carpetas separadas por clase (por ejemplo: *angry*, *happy*, *sad*, etc.) y subdividido en conjuntos de entrenamiento, validación y prueba, lo que facilitó su uso en el desarrollo del modelo.

2. Preprocesamiento de Imágenes

Para mejorar la capacidad de generalización del modelo, se aplicaron técnicas de aumento de datos (data augmentation). Entre las transformaciones realizadas se incluyeron:

- Variación de brillo aleatoria.
- Rotaciones aleatorias de hasta $\pm 15^\circ$.
- Escalado (zoom in/out) aleatorio.

Estas modificaciones se realizaron utilizando las librerías OpenCV y NumPy, y las imágenes resultantes se almacenaron en una carpeta separada llamada *preprocesado*

3. Organización del Dataset

El dataset fue organizado en tres carpetas principales:

- train/: Contiene las imágenes utilizadas para entrenar el modelo.
- valid/: Incluye los datos para la validación durante el entrenamiento.
- test/: Se utilizó para evaluar el rendimiento final del modelo.

Dentro de cada una, las imágenes están clasificadas en subcarpetas por clase emocional.

4. Construcción y Entrenamiento del Modelo

Se diseñó una red neuronal convolucional (CNN) profunda compuesta por múltiples bloques que incluyen:

- Capas Conv2D para la extracción de características.
- Capas MaxPooling2D para la reducción de dimensionalidad.
- BatchNormalization para estabilizar y acelerar el entrenamiento.
- Dropout para reducir el sobreajuste.

Durante el entrenamiento se emplearon varios callbacks:

- EarlyStopping: para detener el entrenamiento si no había mejora en la validación.
- ReduceLROnPlateau: para reducir la tasa de aprendizaje si el rendimiento se estancaba.
- ModelCheckpoint: para guardar el mejor modelo obtenido.

El rendimiento se evaluó usando el conjunto de prueba, generando métricas como la precisión (accuracy), la función de pérdida (loss) y la matriz de confusión.

5. Implementación en Tiempo Real

Una vez entrenado, el modelo fue implementado en una aplicación que detecta emociones faciales en tiempo real. Los principales pasos de esta implementación fueron:

- Carga del modelo utilizando `keras.models.load_model`.
- Captura de video a través de la cámara web con OpenCV.

- Detección de rostros usando un clasificador Haar Cascade preentrenado.
- Preprocesamiento del rostro, que incluye recorte, redimensionado y normalización de la imagen antes de la predicción.
- Suavizado de predicciones, utilizando una estructura tipo historial para estabilizar la salida y evitar fluctuaciones rápidas.
- Visualización en pantalla del nombre de la emoción detectada y su porcentaje de confianza.

6. Librerías Utilizadas

Principales librerías y herramientas empleadas en el proyecto:

- tensorflow y keras: para la construcción, entrenamiento y carga del modelo.
- opencv-python (cv2): para procesamiento de imágenes y video en tiempo real.
- numpy: para operaciones numéricas.
- matplotlib y seaborn: para la visualización de resultados y métricas.
- scikit-learn: para generar la matriz de confusión y otras métricas.
- roboflow: para la obtención y gestión del dataset.

Documentación, desarrollo y pruebas

Selección de dataset y preprocesamiento:

- <https://github.com/KingSplatt/CourselA/blob/main/Unidad4/Tarea1/preprocesar/magenes.ipynb>

Desarrollo del modelo y pruebas (matriz de confusión, métricas):

- <https://github.com/KingSplatt/CourselA/blob/main/Unidad4/Tarea2/Modelo.ipynb>

Implementación:

- <https://github.com/KingSplatt/CourselA/blob/main/Unidad4/Tarea2/camara.ipynb>

Pruebas de funcionamiento (vídeo):

- <https://youtu.be/8kCoOt-08Is>

Conclusión

En conclusión, el desarrollo de un sistema de reconocimiento de imágenes requiere contar con un dataset de calidad, bien procesado y balanceado, ya que de ello depende en gran medida la precisión del modelo. Un conjunto de datos deficiente puede afectar negativamente el rendimiento del sistema.

Por otro lado, el uso de redes neuronales convolucionales en nuestro sistema de reconocimiento facial de emociones ha demostrado ser fundamental, ya que permite identificar expresiones humanas con alta precisión y en tiempo real. Su capacidad para aprender y detectar patrones visuales complejos las convierte en una herramienta ideal para este tipo de aplicaciones.