ARQUITECTURA NEURONAL RECOMENDABLE POR LA COMUNIDAD CIENTÍFICA

Para la detección de emociones, usar una arquitectura CNN (Red Neuronal Convolucional) es una buena opción. Ahora, las CNN proporcionan un enfoque más escalable para las tareas de clasificación de imágenes y reconocimiento de objetos, aprovechando principios del álgebra lineal, específicamente la multiplicación de matrices, para identificar patrones dentro de una imagen. Sin embargo, pueden ser muy exigentes desde el punto de vista informático, ya que requieren unidades de procesamiento gráfico (GPU) para entrenar los modelos.
Las CNN comenzaron a desarrollarse gracias a los trabajos pioneros de Kunihiko Fukushima y Yann LeCun en las décadas de 1980 y 1990. LeCun destacó por aplicar exitosamente el algoritmo de propagación inversa al reconocimiento de códigos postales escritos a mano, lo que llevó al desarrollo de la arquitectura LeNet-5. A partir de entonces, surgieron nuevas arquitecturas más profundas y eficientes, impulsadas por conjuntos de datos como MNIST, CIFAR-10 e iniciativas como el ImageNet Challenge. Entre las arquitecturas más influyentes se encuentran AlexNet, VGGNet, GoogLeNet, ZFNet y ResNet.

Para nuestro proyecto utilizamos ResNet18, una red neuronal convolucional de 18 capas diseñada para facilitar el entrenamiento de modelos profundos mediante el uso de conexiones residuales. Elegimos esta arquitectura por su balance entre rendimiento y eficiencia computacional.

Entrenamos el modelo utilizando PyTorch en CPU, ya que nuestra tarjeta gráfica no era compatible con el entorno de entrenamiento. Esta limitación influyó en la nuestra elección de ResNet18, ya que versiones más profundas como ResNet50 ofrecen un mejor rendimiento, pero requieren mayor capacidad de procesamiento. De hecho, intentamos usar ResNet50, pero el proceso de entrenamiento en CPU era demasiado lento y tardaba varias horas, por lo que elegimos una arquitectura más ligera y manejable para nuestra configuración.

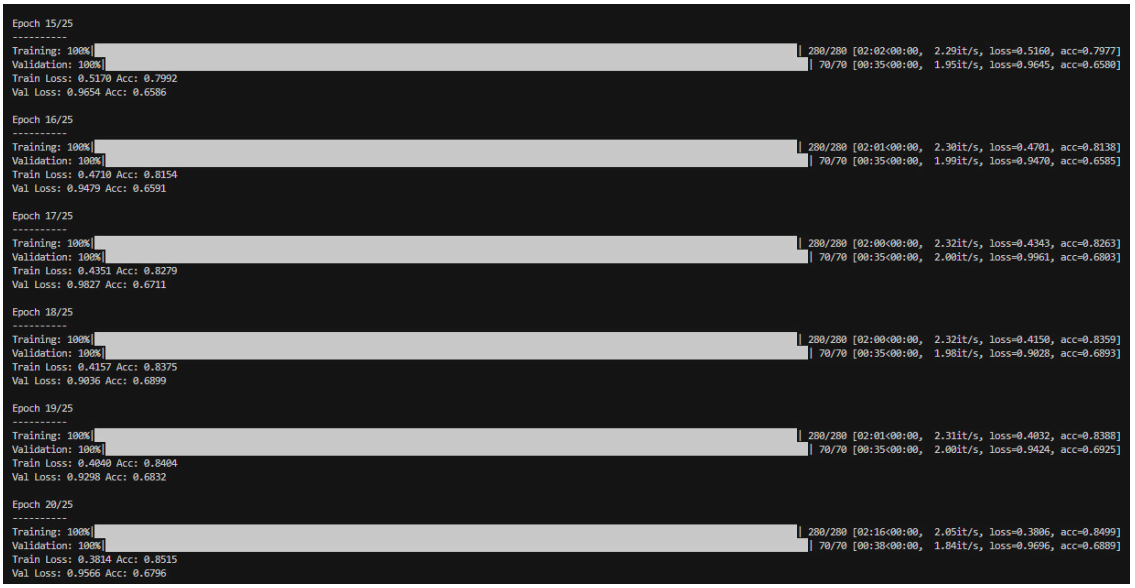
PARÁMETROS DE ENTRENAMIENTO

- Modelo base: ResNet18.
- Dispositivo de entrenamiento: CPU (por incompatibilidad con la GPU).
- Épocas de entrenamiento: 25
- Tamaño de lote (batch size): 32
- Tamaño de imagen: 64 × 64 píxeles

Transformaciones aplicadas:

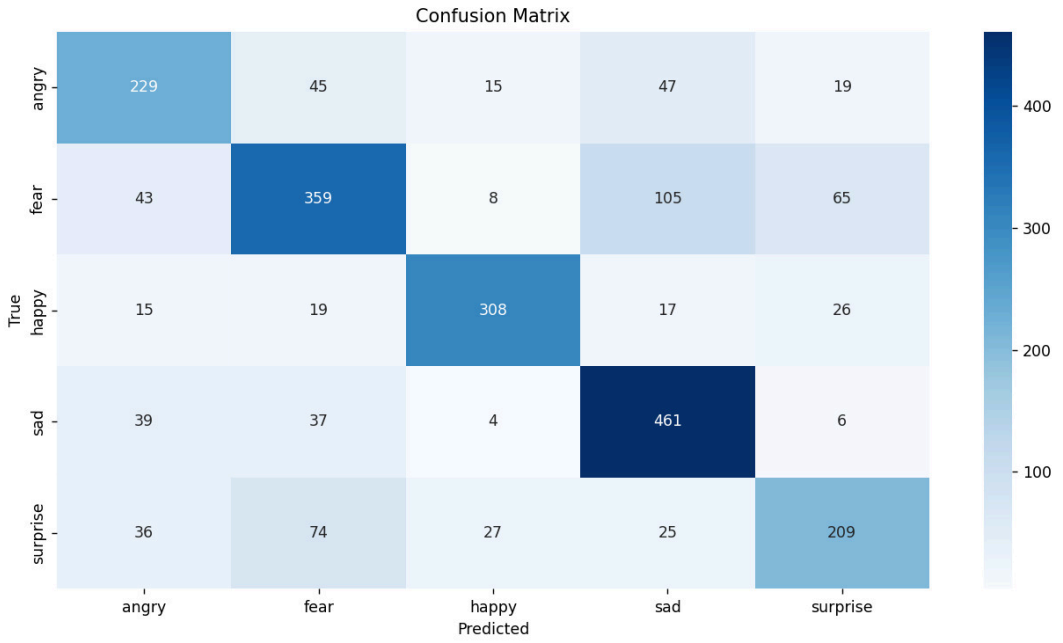
- Redimensionamiento a 64 × 64
- Rotación aleatoria
- Recorte y cambio de escala (RandomResizedCrop)
- Volteo horizontal (RandomHorizontalFlip)
- Ajuste de brillo, contraste y saturación (ColorJitter)

Proceso de entrenamiento



Final de entrenamiento

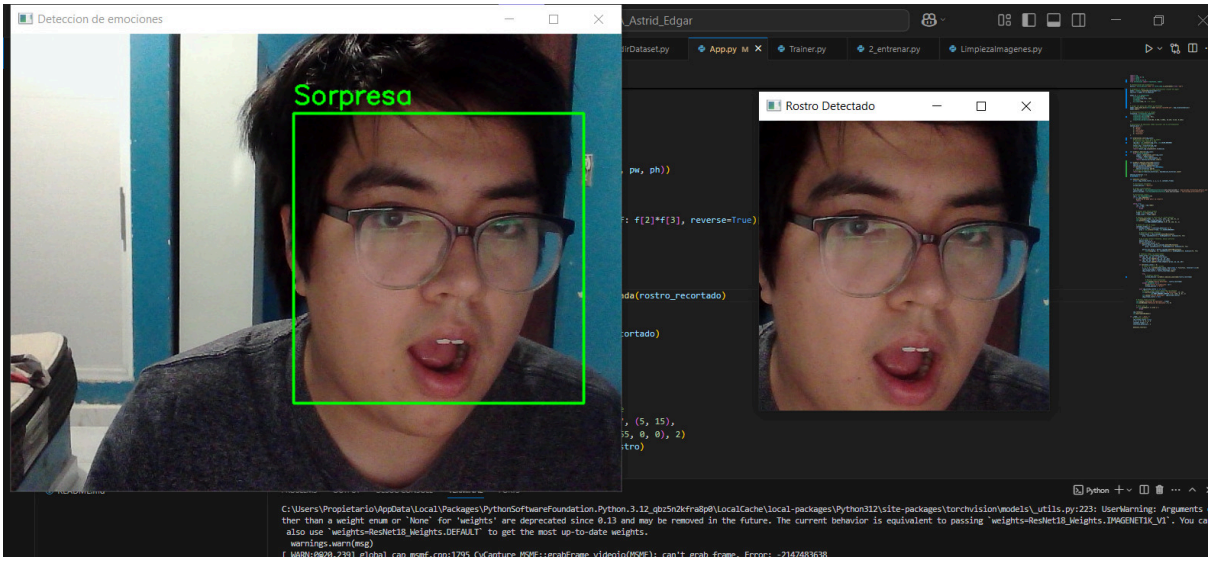
Matriz de confusión

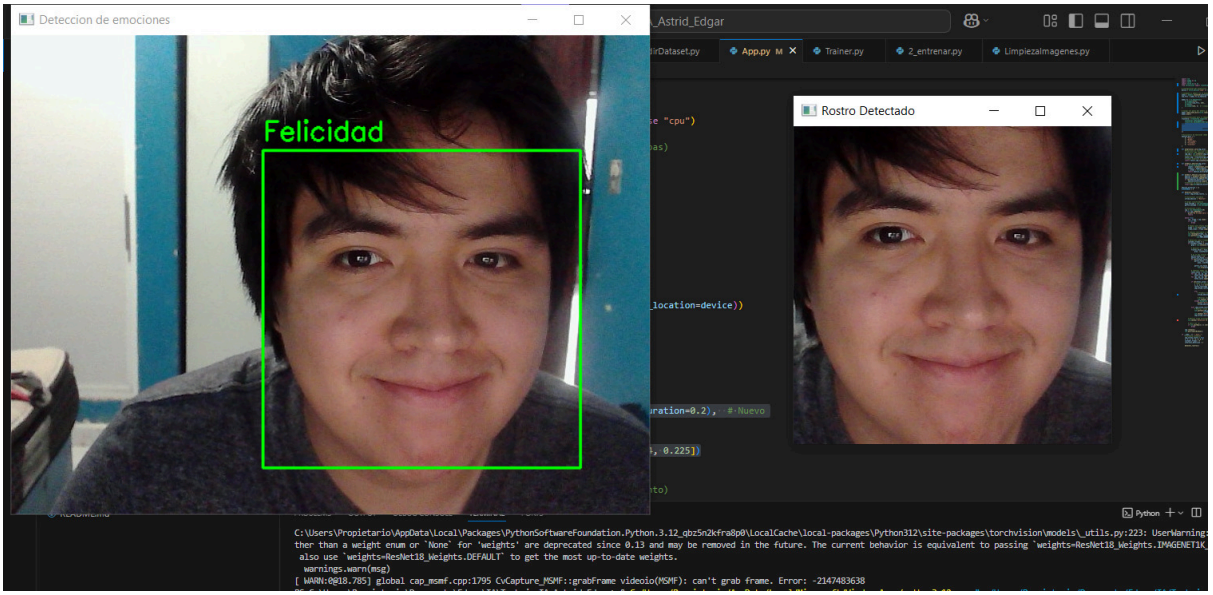


Al finalizar el entrenamiento, se generó una matriz de confusión para analizar el desempeño del modelo. Lo que nos permitió identificar con mayor precisión las clases que el modelo tiende a confundir.

Los resultados indican que, aunque el modelo logra reconocer correctamente la mayoría de las emociones, el modelo tiende a confundir "miedo" y "tristeza" con frecuencia.

PRUEBAS FINALES





VIDEO DE PRUEBAS: https://youtu.be/iXd80nji9gs

BIBLIOGRAFÍA

https://www.ibm.com/mx-es/think/topics/convolutional-neural-networks#:~:text=Ahora%2C%20las%20CNN%20proporcionan%20un,patrones%20dentro%20de%20una%20imagen.

https://www.geeksforgeeks.org/resnet18-from-scratch-using-pytorch/