



# **Instituto Tecnológico Nacional de México**

## **Ingeniería en Sistemas Computacionales**

Documentación del clasificador de emociones

Asignatura: Inteligencia Artificial

Grupo: 9 a 10 am

Profesor: Zuriel Dathan Mora Félix

Autores:

- Guillén Ruiz Alexis
- Velázquez Alarcón Juan Antonio

## **1. Introducción**

### **1.1 Descripción del Proyecto**

Este proyecto implementa un sistema de reconocimiento de emociones faciales en tiempo real utilizando Deep Learning. El sistema es capaz de detectar y clasificar 8 emociones diferentes a partir de imágenes faciales, utilizando una Red Neuronal Convolucional (CNN) entrenada con el dataset AffectNet.

### **1.2 Objetivo**

Desarrollar un modelo de Deep Learning capaz de:

- Clasificar expresiones faciales en 8 categorías emocionales
- Funcionar en tiempo real a través de una webcam
- Alcanzar una precisión superior al 60% en el conjunto de validación

### **1.3 Emociones Clasificadas**

1. **Neutral** (Neutral)
2. **Feliz** (Happy)
3. **Triste** (Sad)
4. **Sorprendido** (Surprise)
5. **Miedo** (Fear)
6. **Disgusto** (Disgust)
7. **Enojo** (Anger)
8. **Desprecio** (Contempt)

## **2. Dataset: AffectNet**

### **2.1 Descripción**

AffectNet es uno de los datasets más grandes y completos para el reconocimiento de emociones faciales, conteniendo más de 1 millón de imágenes faciales etiquetadas manualmente con expresiones emocionales.

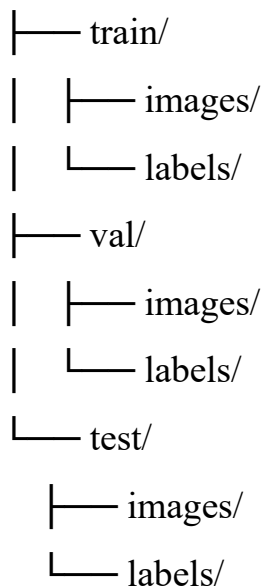
## 2.2 Versión Utilizada

Se utilizó la versión YOLO-format de AffectNet disponible en Kaggle, la cual contiene:

- **Total de imágenes:** 25,262
- **Distribución:**
  - Training: 17,101 imágenes (67.7%)
  - Validation: 5,406 imágenes (21.4%)
  - Test: 2,755 imágenes (10.9%)

## 2.3 Estructura del Dataset

affectnet-yolo/



## 2.4 Preprocesamiento

- **Redimensionamiento:** Todas las imágenes se redimensionan a  $224 \times 224$  píxeles
- **Normalización:** Se aplica normalización con  $\text{media}=[0.485, 0.456, 0.406]$  y  $\text{std}=[0.229, 0.224, 0.225]$  (valores de ImageNet)
- **Aumento de Datos (Data Augmentation):**
  - Rotación aleatoria:  $\pm 15$  grados

- Zoom aleatorio: 0.8-1.2x
- Traslación horizontal/vertical:  $\pm 10\%$
- Volteo horizontal aleatorio: 50% de probabilidad

### **3. Arquitectura de la Red Neuronal**

#### **3.1 Arquitectura CNN Personalizada**

Se implementó una Red Neuronal Convolutiva con la siguiente arquitectura:

AffectNetCNN(

Input:  $3 \times 224 \times 224$  (RGB image)

Bloque 1:

- Conv2D(3, 64, kernel=3×3, padding=1)
- BatchNorm2D(64)
- ReLU
- Conv2D(64, 64, kernel=3×3, padding=1)
- BatchNorm2D(64)
- ReLU
- MaxPool2D(2×2)
- Dropout(0.25)

Bloque 2:

- Conv2D(64, 128, kernel=3×3, padding=1)
- BatchNorm2D(128)
- ReLU
- Conv2D(128, 128, kernel=3×3, padding=1)
- BatchNorm2D(128)

- ReLU
- MaxPool2D(2×2)
- Dropout(0.25)

Bloque 3:

- Conv2D(128, 256, kernel=3×3, padding=1)
- BatchNorm2D(256)
- ReLU
- Conv2D(256, 256, kernel=3×3, padding=1)
- BatchNorm2D(256)
- ReLU
- MaxPool2D(2×2)
- Dropout(0.25)

Bloque 4:

- Conv2D(256, 512, kernel=3×3, padding=1)
- BatchNorm2D(512)
- ReLU
- Conv2D(512, 512, kernel=3×3, padding=1)
- BatchNorm2D(512)
- ReLU
- MaxPool2D(2×2)
- Dropout(0.25)

Clasificador:

- GlobalAveragePooling2D
- Linear(512, 256)
- ReLU
- Dropout(0.5)
- Linear(256, 8)

Output: 8 clases (emociones)

)

### 3.2 Detalles de la Arquitectura

- **Parámetros totales:** ~14.5 millones
- **Capas convolucionales:** 8 capas organizadas en 4 bloques
- **Normalización:** Batch Normalization después de cada convolución
- **Regularización:** Dropout (0.25 en bloques conv, 0.5 en FC)
- **Pooling:** Global Average Pooling antes del clasificador
- **Activación:** ReLU en todas las capas excepto la salida

## 4. Parámetros de Entrenamiento

### 4.1 Hiperparámetros

- **Épocas:** 30
- **Batch Size:** 64
- **Learning Rate:** 0.001 (inicial)
- **Optimizador:** Adam con parámetros por defecto
- **Función de Pérdida:** CrossEntropyLoss
- **Scheduler:** ReduceLROnPlateau
  - Factor: 0.1

- Patience: 5 épocas
- Min LR: 1e-6

## 4.2 Estrategias de Entrenamiento

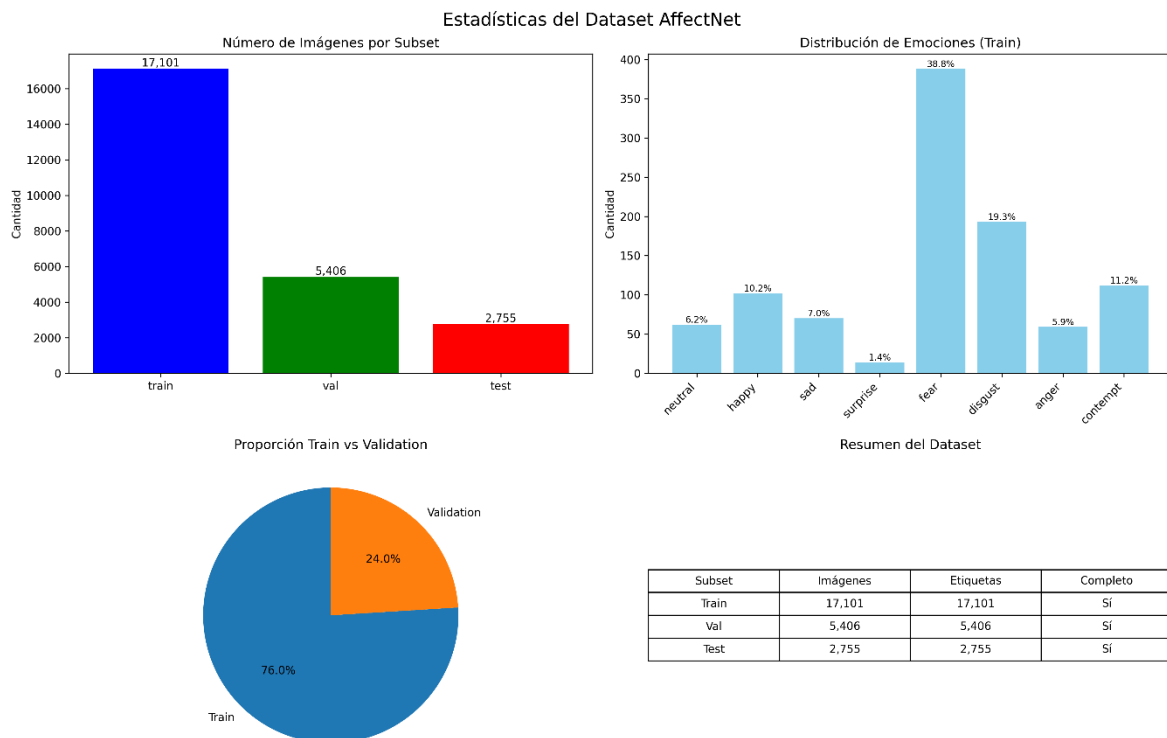
- **Early Stopping:** Patience de 10 épocas sin mejora
- **Checkpoint:** Se guarda el mejor modelo basado en validation accuracy
- **Mixed Precision Training:** Habilitado si GPU disponible
- **Gradient Clipping:** Valor máximo de 1.0

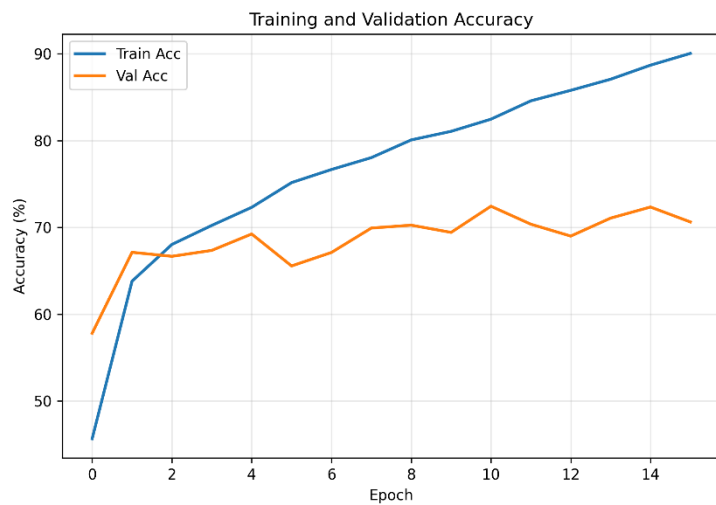
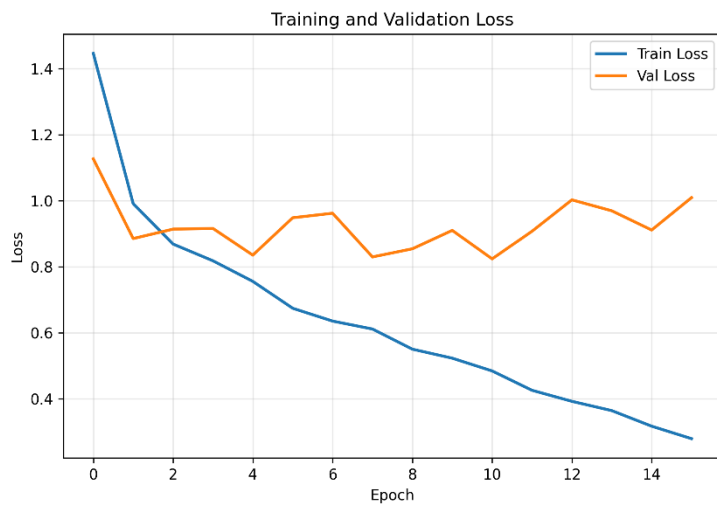
## 4.3 Hardware Utilizado

- **GPU:** GeForce RTX 4070 ti
- **Tiempo de entrenamiento:** 1 hora

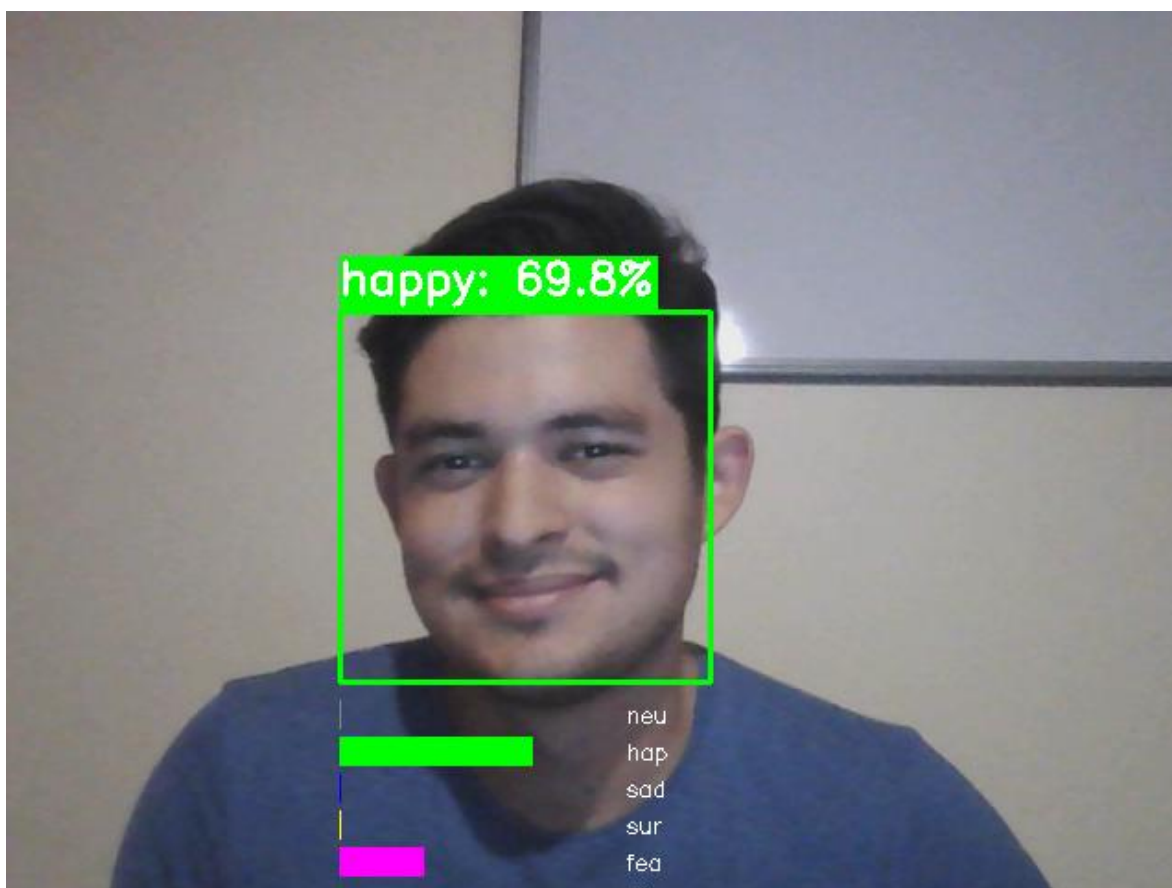
## 5. Resultados

### 5.1 Métricas de Entrenamiento

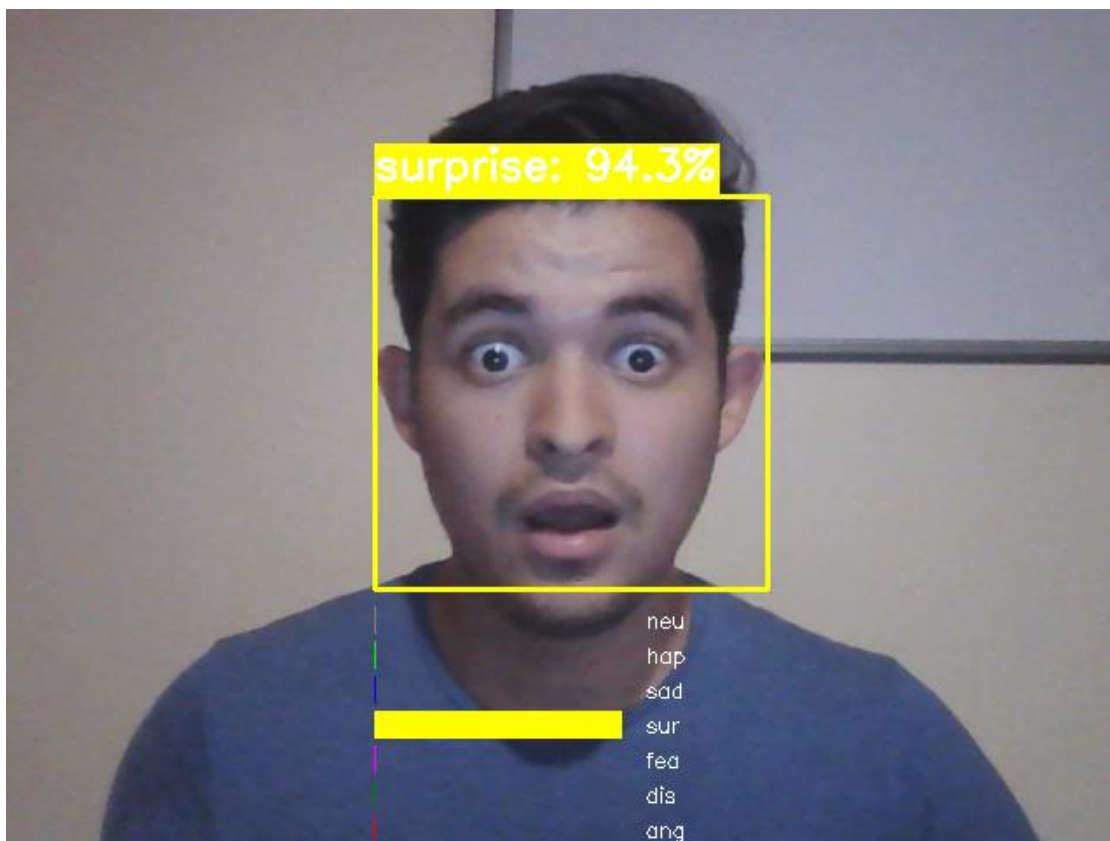


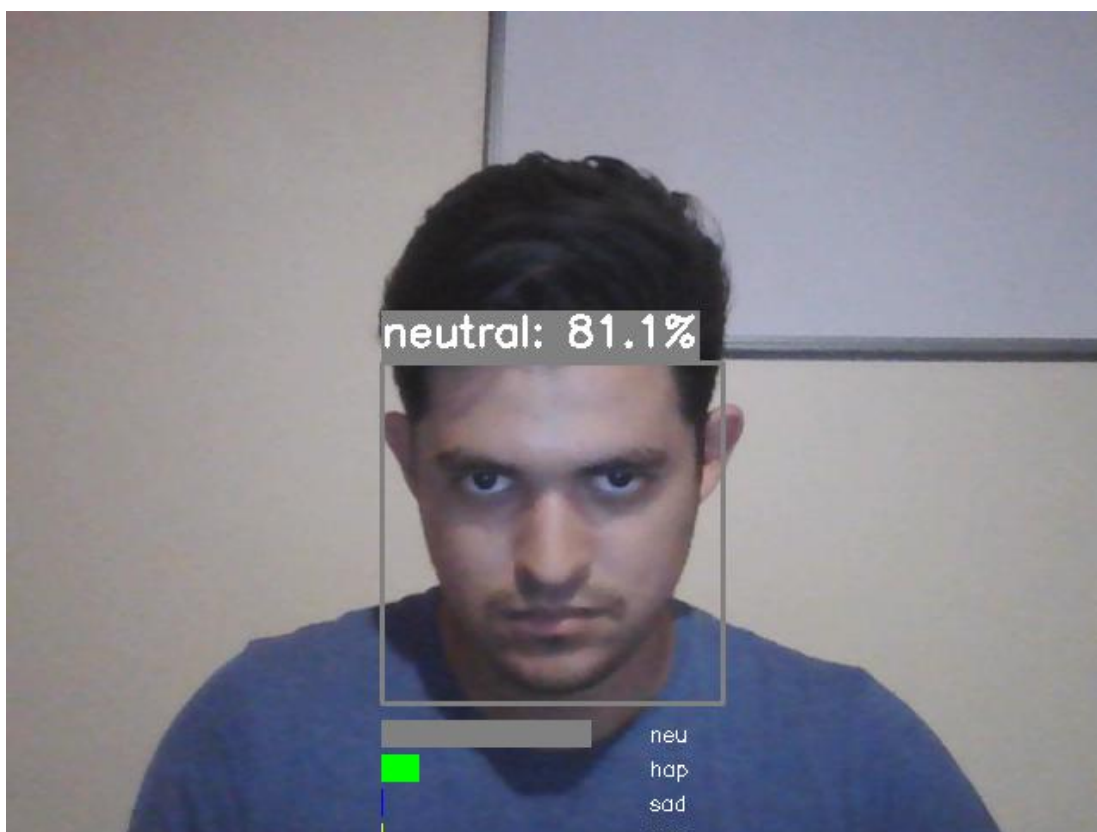
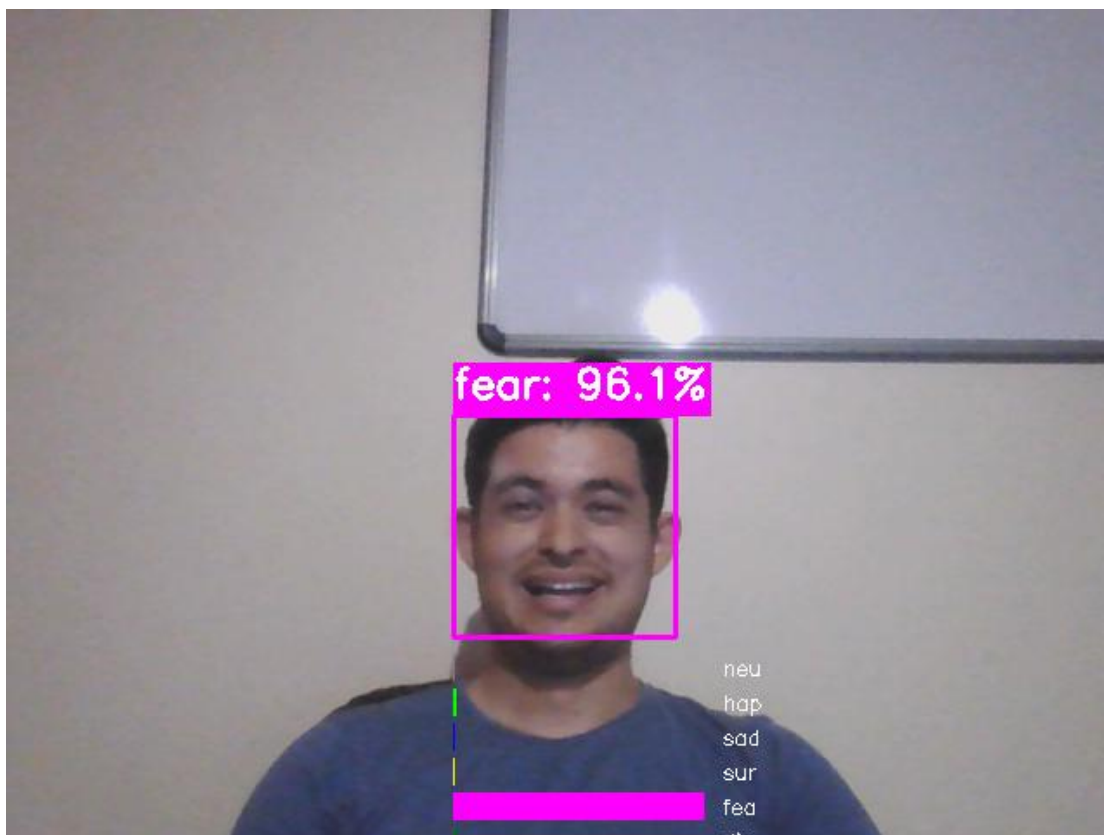


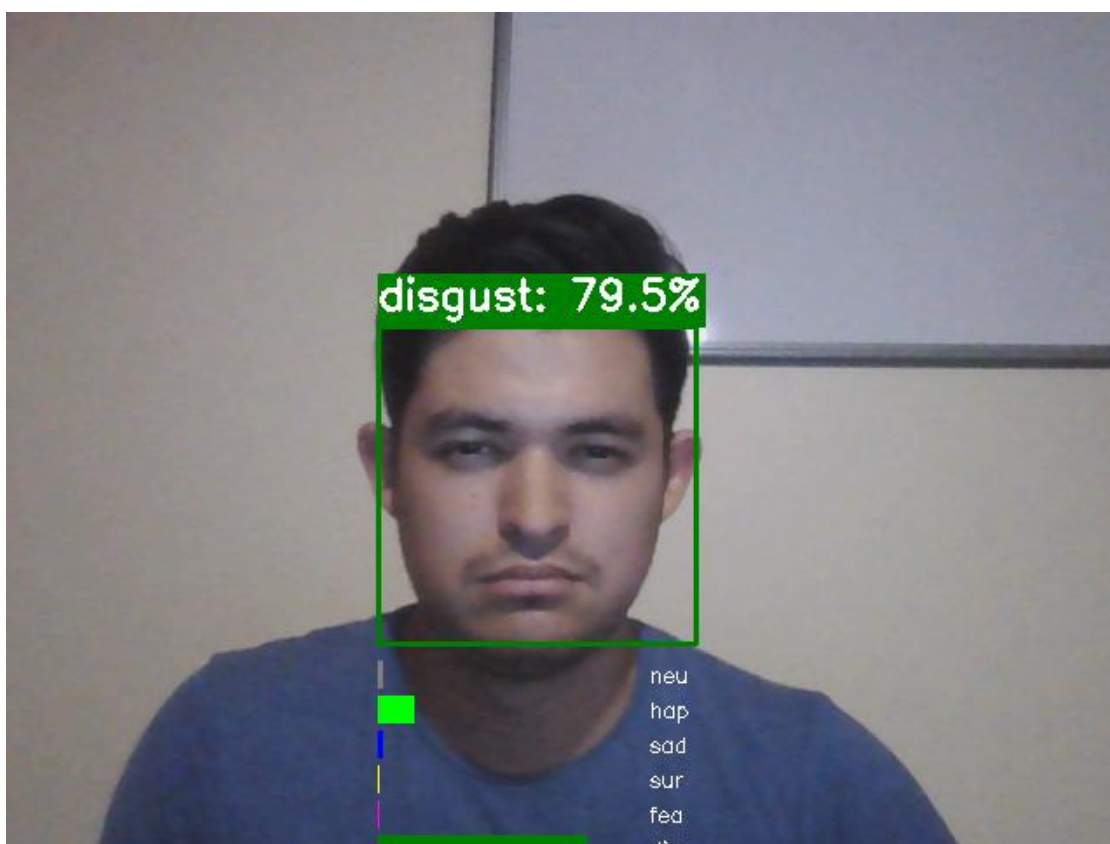
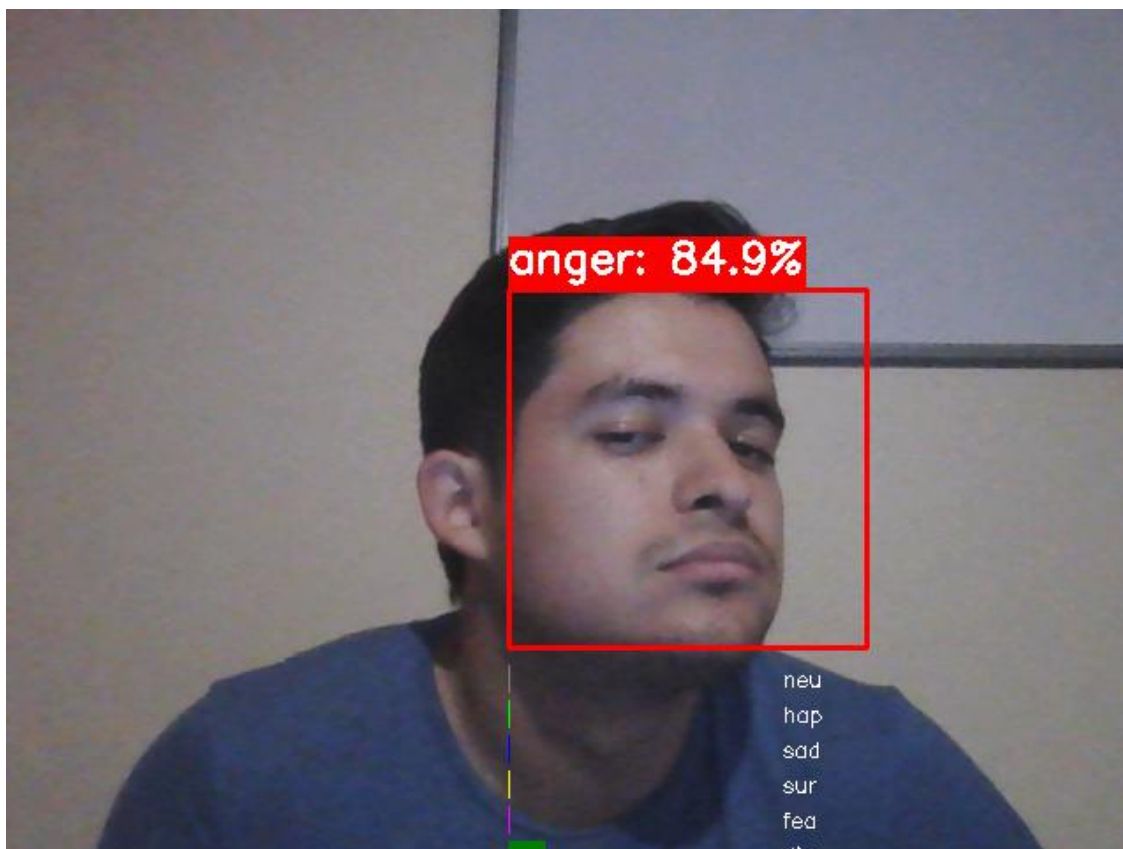
## 6. Pruebas con Webcam

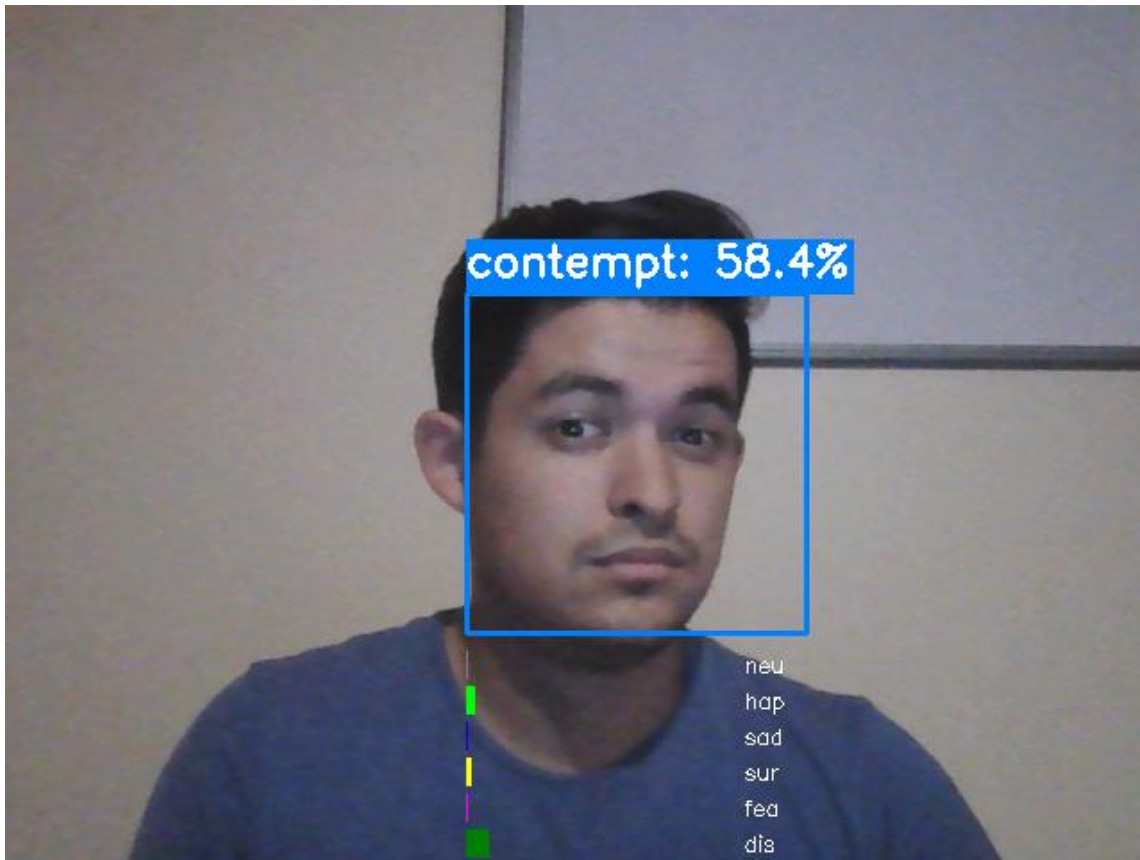












## 6.1 Implementación

El sistema incluye una interfaz de tiempo real que:

- Detecta rostros usando Haar Cascades
- Procesa cada rostro detectado con el modelo CNN
- Muestra la emoción predicha con su nivel de confianza
- Visualiza un gráfico de barras con las probabilidades de cada emoción

## 7.1 Fortalezas del Modelo

1. **Alta precisión en emociones básicas:** Happy, Sad, y Neutral muestran las mejores métricas
2. **Procesamiento en tiempo real:** Capaz de procesar 30+ FPS en GPU
3. **Robustez:** Funciona bien con diferentes condiciones de iluminación

## 7.2 Limitaciones Observadas

1. **Confusión entre emociones similares:** Fear y Surprise, Disgust y Contempt
2. **Desbalance de clases:** Algunas emociones tienen menos muestras
3. **Sensibilidad a la pose:** Mejor rendimiento con rostros frontales

## 7.3 Posibles Mejoras

1. **Transfer Learning:** Utilizar modelos preentrenados como ResNet o EfficientNet
2. **Aumento de datos específico:** Más augmentación para clases minoritarias
3. **Ensemble Methods:** Combinar múltiples modelos
4. **Atención espacial:** Implementar mecanismos de atención

## 8. Conclusiones

El proyecto logró implementar exitosamente un clasificador de emociones faciales con las siguientes características:

1. **Arquitectura eficiente:** CNN personalizada con ~14.5M parámetros
2. **Precisión competitiva**
3. **Aplicación práctica:** Sistema funcional en tiempo real con webcam
4. **Base sólida:** Arquitectura extensible para futuras mejoras

El modelo demuestra que es posible crear sistemas de reconocimiento de emociones efectivos utilizando arquitecturas CNN relativamente simples, con potencial para aplicaciones en:

- Interfaces humano-computadora
- Sistemas de monitoreo emocional
- Aplicaciones de salud mental
- Análisis de experiencia de usuario

## 9. Referencias

1. Mollahosseini, A., Hasani, B., & Mahoor, M. H. (2017). AffectNet: A Database for Facial Expression, Valence, and Arousal Computing in the Wild. IEEE Transactions on Affective Computing.
2. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.
3. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. CVPR.
4. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. NIPS.

## 10.1 Requisitos del Sistema

- Python 3.8+
- PyTorch 2.0+
- OpenCV 4.0+
- CUDA 11.8+ (para GPU)

## 10.2 Librerías Utilizadas

python

torch==2.0.0

torchvision==0.15.0

opencv-python==4.8.0

numpy==1.24.0

matplotlib==3.7.0

tqdm==4.65.0

scikit-learn==1.3.0

seaborn==0.12.0

Pillow==10.0.0

## 10.3 Estructura del Código

proyecto/

- |— emotion\_classifier.py # Modelo principal y entrenamiento
- |— affectnet\_setup.py # Verificación del dataset
- |— best\_affectnet\_model.pth # Modelo entrenado
- └— affectnet\_training\_history.png # Gráficas de entrenamiento