Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Estado de México

Inteligencia Artificial Avanzada para la Ciencia de Datos II (TC3007C)

M2. Implementación de un modelo de Deep Learning para la detección de rostro

Presenta:

Abraham Gil Félix A01750884

Profesor Módulo 2:

Dr. Julio Guillermo Arriaga Blumenkron

Fecha de entrega:

02 de noviembre de 2022

Resumen:

Con el fin de poner en práctica las habilidades y conocimientos obtenidos en el Módulo 2 “Técnicas y arquitecturas de Deep Learning”, se optó por crear un modelo de reconocimiento de rostro. A continuación, se explica cada una de las etapas del proyecto, así como la implementación general con ayuda de Python.

Face Detection Model

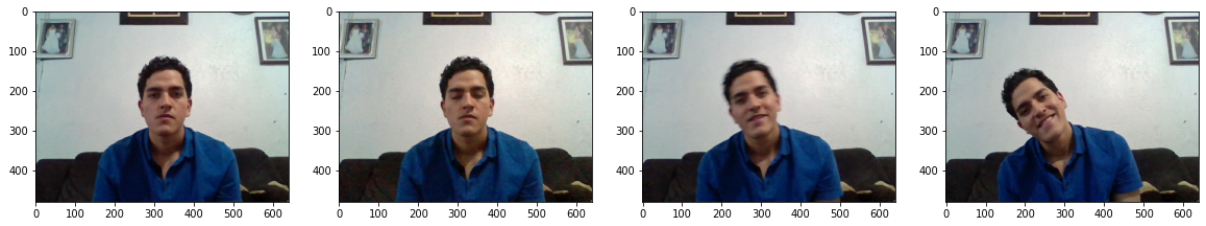
Dependencias

- Labelme
- Tensorflow
- OpenCV
- Albumentations:
- Uuid
- Json
- Numpy
- Matplotlib

Creación del conjunto de datos original

Con ayuda de **OpenCV** y un sujeto de prueba, se lograron obtener 90 imágenes donde se puede y no, apreciar el rostro del sujeto. Esto fue posible gracias a la captura de 3 videos al sujeto en diferentes temporalidades.

Dado el pequeño conjunto de imágenes, se designó el 70% de estas para entrenamiento, el 13% para validación y 17% restante para prueba. A continuación, se muestra una pequeña muestra del conjunto:

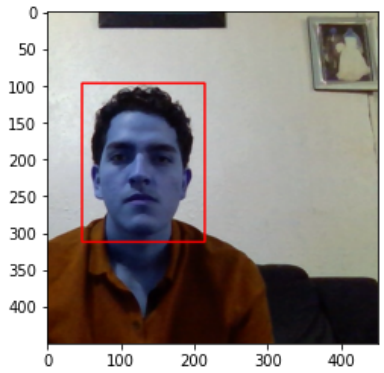


Respecto al etiquetado de cada imagen, se utilizó **Labelme,** consistió en seleccionar aquellas imágenes en las cuales es posible apreciar el rostro del sujeto de prueba. De este modo se logra obtener un problema de clasificación binaria.

Aumento del conjunto de datos

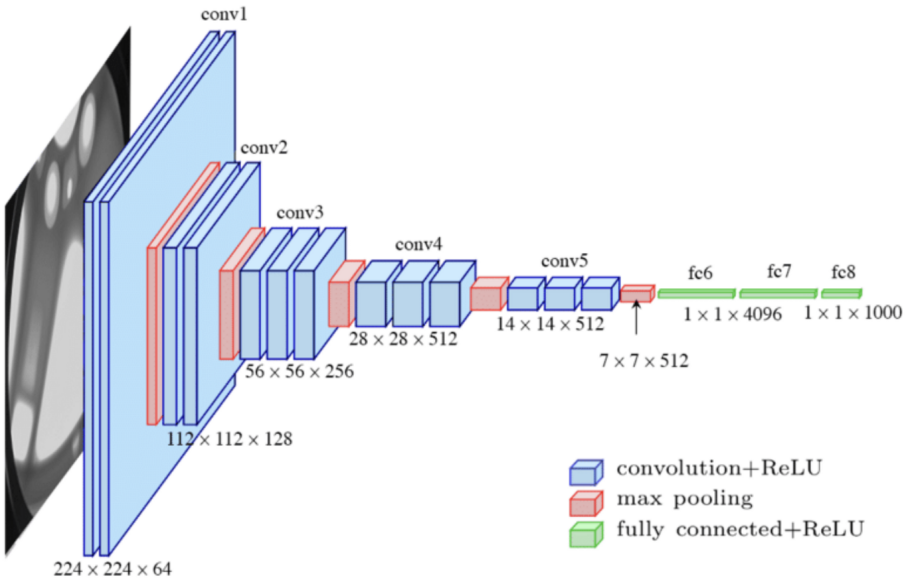
Gracias a **Albumentations**, expandir 60 veces el conjunto de imágenes fue posible. Se tomó en cuenta cada una de las imágenes del conjunto original para recrear, a partir de estás, más imágenes cambiando su contraste, tamaño, orientación horizontal o vertical y tonalidades. Además, se guardan las coordenadas del objeto detectado (en este caso, el rostro del sujeto de prueba). Es así, como el conjunto de datos pasó de tener 90 a 5,400 imágenes para el entrenamiento del modelo de aprendizaje profundo.

Visualización de una de las nuevas imágenes creadas a partir del conjunto original:



Modelación

Para el modelo de aprendizaje profundo a implementar se utilizó como base el algoritmo de clasificación y detección de objetos **VVG-16.** Dicho algoritmo es capaz de clasificar 1,000 imágenes de 1,000 categorías diferentes con precisión mayor al 90%, aunque para la solución de este reto únicamente necesitamos que clasifique en 2 categorías diferentes. Es por esto, que se optó por reemplazar el top de la arquitectura de VVG-16 por un modelo de clasificación y un modelo de regresión.



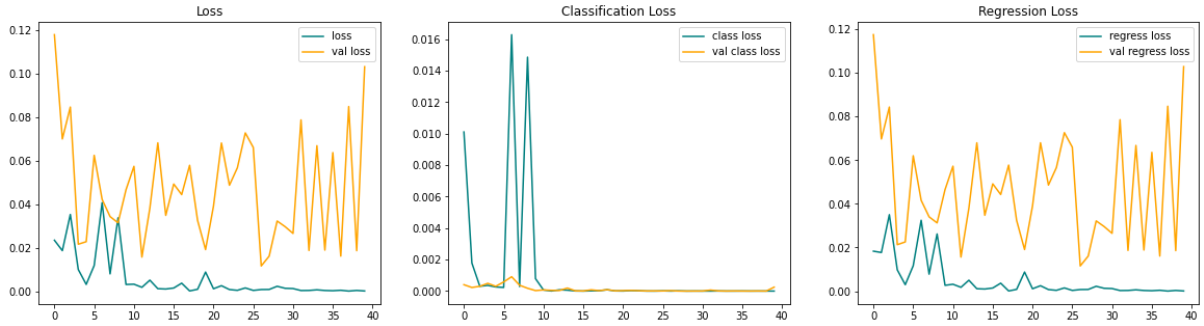
Por su parte, el modelo de clasificación consta de 2 capas densas; la primera de estás con 2,048 neuronas y ReLU como función de activación, y la segunda únicamente con 1 neurona (para la clasificación binaria) y la función de activación sigmoidal. El modelo de regresión posee una arquitectura similar al modelo de clasificación, la diferencia radica en el número de neuronas en la última capa densa, para este es necesario contar con 4 neuronas dado las coordenadas del objeto detectado (rostro del sujeto).

Se optó por Adam (con una tasa de aprendizaje de 0.001) como optimizador, la función de pérdida que se utiliza es similar a la función que emplea el algoritmo YOLO. En general, la arquitectura de red cuenta con un total de 16,826,181 parámetros por entrenar.

Implementación y evaluación del modelo

El entrenamiento de la red neuronal profunda se ejecutó con ayuda de acelerador por hardware en Colab, GPU. Se utilizaron 40 épocas y por supuesto, los conjuntos de entrenamiento y validación.

Las siguientes gráficas corresponden a las funciones de pérdida del modelo. Es posible observar que el rendimiento del modelo no es óptimo, puesto que lo ideal sería que los valores de pérdida disminuyan constantemente. Sin embargo, esto puede ser causado por falta de datos o tiempo de entrenamiento.



Predicciones

Dado que, los resultados para las funciones de pérdida no fueron los ideales, se realizaron numerosas predicciones con el fin de comprobar el buen funcionamiento del modelo de detección de rostro. No obstante, también se implementó en tiempo real con ayuda de OpenCV, en esta última prueba el modelo tuvo un excelente desempeño (mejor de lo esperado). Finalmente, se muestran cuatro predicciones exitosas: