

Implementación de un modelo de Deep Learning.

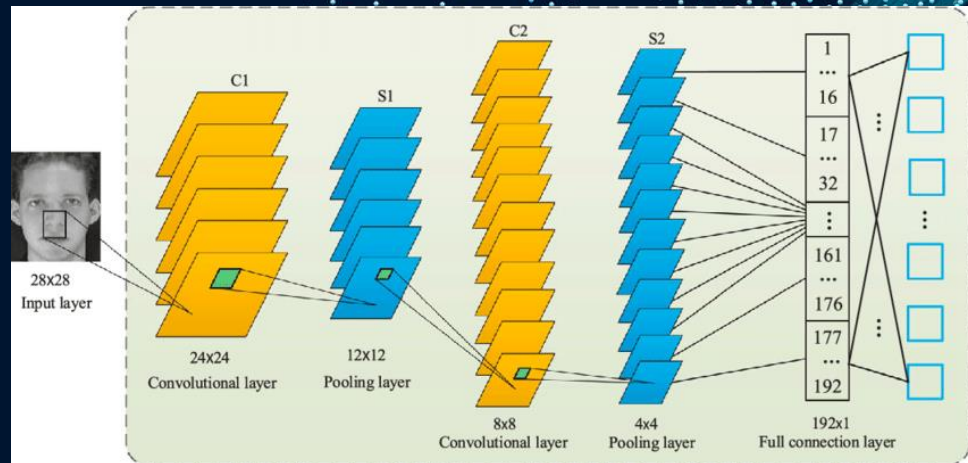
Equipo 6

Selección del Modelo

El problema de reconocimiento facial en asistencia en aulas involucra la identificación precisa de individuos en imágenes o secuencias de video, a menudo en tiempo real.

Las CNN se han destacado en tareas de visión por computadora, incluido el reconocimiento facial, debido a su capacidad inherente para extraer características relevantes de las imágenes.

Las capas convolucionales de una CNN actúan como detectores de características, identificando patrones locales como bordes, texturas y formas en las imágenes. Estas características aprendidas permiten al modelo comprender eficazmente las características distintivas de los rostros humanos, como ojos, narices y bocas.



Diversos artículos científicos de vanguardia respaldan esta elección. Por ejemplo, FaceNet (Schroff et al., 2015) presenta un enfoque que utiliza CNN para mapear rostros en un espacio de características comunes, lo que permite una comparación precisa y eficiente entre rostros. De manera similar, DeepFace (Taigman et al., 2014) ha demostrado un rendimiento equiparable al humano en tareas de verificación facial.

Selección del DataSet

Para este modelo en específico usamos un dataset de acceso público y libre que contenía las caras de personas famosos como Henry Cavill, Robert Downey Jr, etc con el fin de entrenar el modelo para el reconocimiento facial de estas personas y así usarlo posteriormente con nuevas caras de estudiante o cualquier persona.

Este Dataset puede encontrarse en Kaggle como Face Recognition Dataset con una cantidad total de 2562 imágenes de rostros el cual concluimos que es una cantidad suficiente para el entrenamiento de los datos

Link: <https://www.kaggle.com/datasets/vasukipatel/face-recognition-dataset/data>

Face Recognition Dataset

Face Data of 31 different classes.

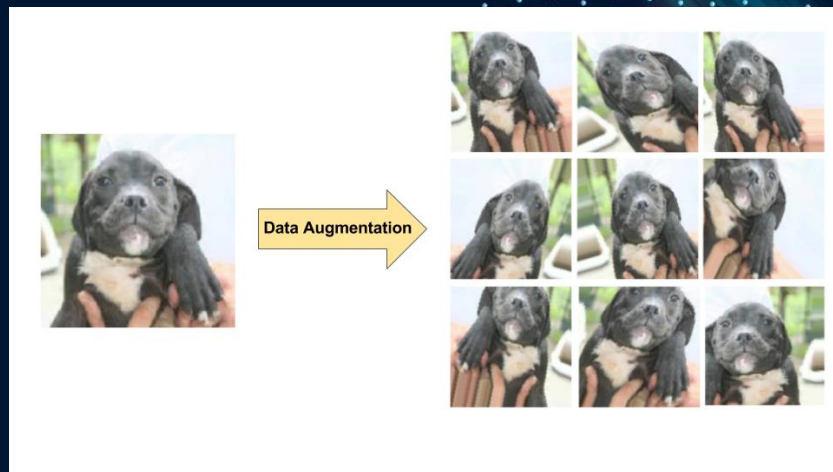


Procesamiento de Datos

Se utilizó la biblioteca ImageDataGenerator de Keras para crear un generador de datos a partir de un directorio que contiene imágenes. Estas imágenes se redimensionaron a un tamaño de 224x224 píxeles y se agruparon en lotes de 32. Además, se obtuvieron las clases a partir de los nombres de las subcarpetas en el directorio. Este proceso permite cargar y preparar datos de imágenes de manera eficiente para su posterior uso en un modelo de aprendizaje profundo.

```
train_dir="/content/drive/MyDrive/face-recognition-dataset/Original Images/Original Images"
generator = ImageDataGenerator()
train_ds = generator.flow_from_directory(train_dir,target_size=(224, 224),batch_size=32)
classes = list(train_ds.class_indices.keys())
```

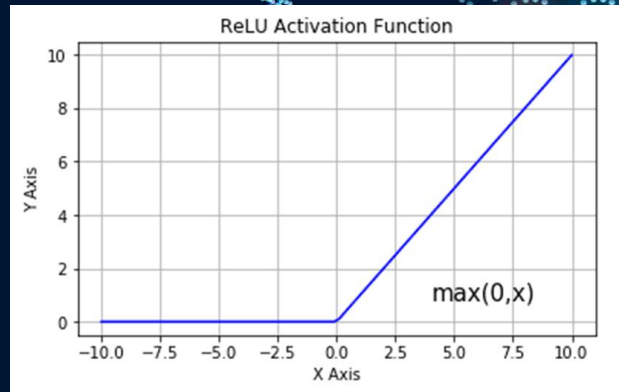
Found 2572 images belonging to 31 classes.



Creación del Modelo

Hemos creado un modelo de red neuronal convolucional (CNN) con varias capas Conv2D que extraen características importantes de las imágenes. Usamos funciones de activación 'relu' para mejorar el aprendizaje. También incluimos capas de normalización (BatchNormalization) y capas de dropout para mejorar la estabilidad y evitar el sobreajuste. El modelo se ajusta a los datos de entrenamiento para realizar el reconocimiento facial de manera precisa.

Posteriormente entrenamos nuestro modelo utilizando los datos de entrenamiento. Realizamos 30 épocas de entrenamiento, lo que significa que el modelo aprenderá de los datos 30 veces. Cada época, procesamos un lote de 32 imágenes a la vez para ajustar el modelo y mejorar su rendimiento.



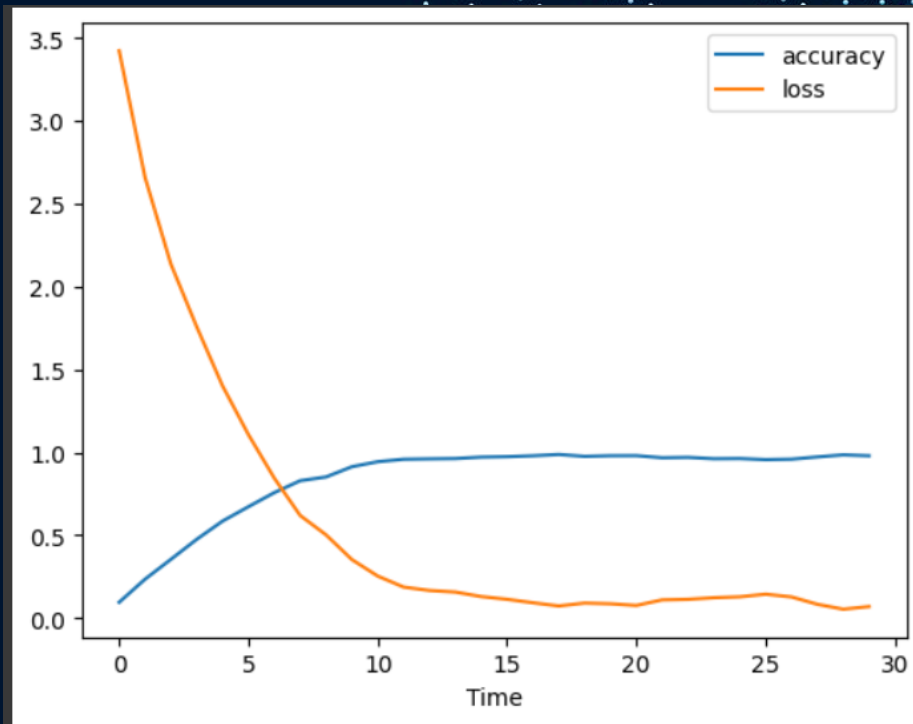
```
81/81 [=====] - 44s 532ms/step - loss: 0.1300 - accuracy: 0.9712
Epoch 16/30
81/81 [=====] - 45s 542ms/step - loss: 0.1135 - accuracy: 0.9740
Epoch 17/30
81/81 [=====] - 43s 533ms/step - loss: 0.0926 - accuracy: 0.9794
Epoch 18/30
81/81 [=====] - 44s 548ms/step - loss: 0.0733 - accuracy: 0.9872
Epoch 19/30
81/81 [=====] - 43s 528ms/step - loss: 0.0908 - accuracy: 0.9759
Epoch 20/30
81/81 [=====] - 43s 534ms/step - loss: 0.0866 - accuracy: 0.9798
Epoch 21/30
81/81 [=====] - 44s 536ms/step - loss: 0.0765 - accuracy: 0.9802
Epoch 22/30
81/81 [=====] - 44s 550ms/step - loss: 0.1101 - accuracy: 0.9673
Epoch 23/30
81/81 [=====] - 43s 535ms/step - loss: 0.1134 - accuracy: 0.9697
Epoch 24/30
81/81 [=====] - 43s 534ms/step - loss: 0.1231 - accuracy: 0.9615
Epoch 25/30
81/81 [=====] - 43s 532ms/step - loss: 0.1286 - accuracy: 0.9627
Epoch 26/30
81/81 [=====] - 43s 534ms/step - loss: 0.1444 - accuracy: 0.9561
Epoch 27/30
81/81 [=====] - 44s 541ms/step - loss: 0.1278 - accuracy: 0.9588
Epoch 28/30
81/81 [=====] - 42s 525ms/step - loss: 0.0834 - accuracy: 0.9724
Epoch 29/30
81/81 [=====] - 43s 530ms/step - loss: 0.0539 - accuracy: 0.9848
Epoch 30/30
81/81 [=====] - 44s 542ms/step - loss: 0.0696 - accuracy: 0.9794
```

Rendimiento del Modelo

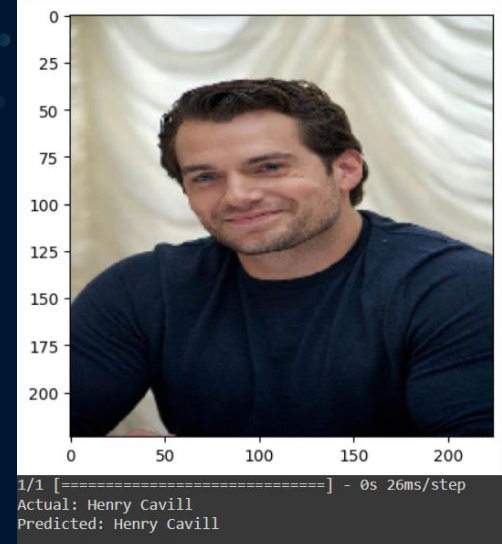
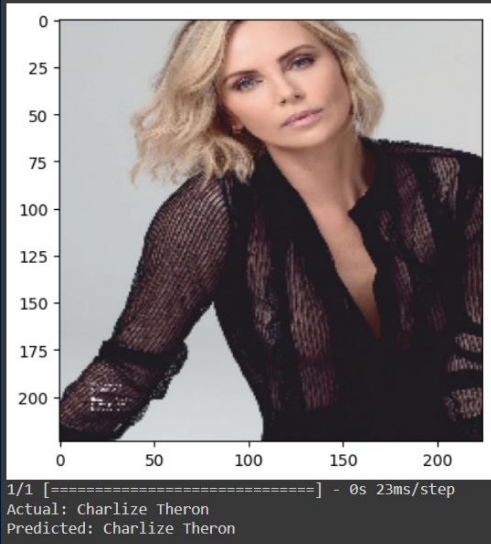
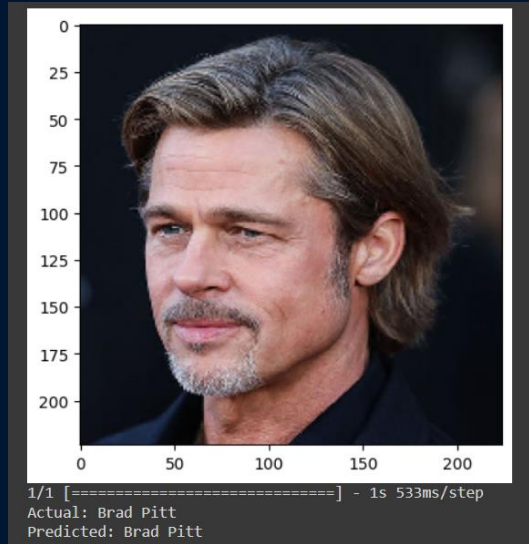
Visualización del rendimiento del modelo a lo largo del tiempo. Graficamos la precisión (accuracy) y la pérdida (loss) en función del número de épocas. Esto nos permite evaluar cómo nuestro modelo está mejorando con el tiempo.

La precisión debería aumentar, mientras que la pérdida debería disminuir indicando una buena predicción y entrenamiento de imágenes o rostros

accuracy: 0.9794



Pruebas del Modelo



De los tres diferentes actores famosos, el modelo CNN pudo identificar correctamente la cara de cada individuo junto a su respectivo nombre probando la eficacia de nuestro modelo para el reconocimiento de rostros en famosos.



Modelo Pre Entrenado utilizado en el proyecto

Face Recognition

Pese a que el modelo en general arrojó buenos resultados, si se compara con librerías que contienen un modelo pre-entrenado (Face-recognition de python) podemos observar que en general:

- Tiene mejor precisión (99.38% vs 97.34%)
- Mayor soporte e integración con otras librerías (opencv)
- Tiene una comunidad que está familiarizada con este modelo pre-entrenado

Link: https://github.com/5100-chap/TC3007C.501_EQ6/blob/main/Anexos/FaceRecognition.py

