

Instituto Tecnológico y de Estudios Superiores de Monterrey

Inteligencia Artificial Avanzada para la Ciencia de Datos II (Gpo 501)



**Tecnológico
de Monterrey**

**Momento de Retroalimentación Individual: Implementación de un
modelo de Deep Learning**

Jorge Eduardo De León Reyna - A00829759

Noviembre 4, 2023

1. Introducción

En el presente documento se muestra el proceso de implementación de un modelo de Deep Learning para el momento de retroalimentación correspondiente. El principal objetivo del mismo es demostrar el cumplimiento de los puntos indicados en la rúbrica de evaluación de la actividad, así como también explicar el proceso de desarrollo y las herramientas utilizadas.

2. Desarrollo del proyecto

El proyecto se desarrolla bajo la arquitectura VGG 16, la cual fue desarrollada previamente por el Visual Geometry Group de la Universidad de Oxford. Esta misma está diseñada y es conocida por su simplicidad en tareas de clasificación de imágenes y es una de las Redes Neuronales Convolucionales (CNN) pre-entrenadas más famosas y utilizadas en la actualidad.

Ahora bien, como parte de esta actividad se llevó a cabo un proceso de Transfer Learning donde a partir de la arquitectura de la CNN mencionada anteriormente se aplicó un proceso de fine-tuning para ajustarla al contexto del reto y la actividad el cual corresponde a la detección de rostros. Para esto se llevaron a cabo las siguientes modificaciones:

1. Implementación de capa convolucional.
2. Implementación de capa Flatten.
3. Implementación de capa densa de 256 neuronas con activación de tipo ReLU
4. Implementación de capa densa de 16 neuronas con activación Softmax.

Estas modificaciones se hicieron con el objetivo de adaptar la arquitectura inicial al problema analizado el cual tiene como principales diferencias:

1. Clasificación multi-categoría
2. Problema orientado a la clasificación de rostros
3. Re-entrenamiento orientado a la clasificación de rostros.

a. Selección del dataset

Para el problema abordado se seleccionó un set de datos de rostros de personas reales obtenidos de la siguiente fuente:

<https://thinkingneuron.com/face-recognition-using-deep-learning-cnn-in-python/>

El set de datos seleccionado cuenta con imágenes de rostros reales, es de tipo open source y además cuenta con una sección de datos para validación y otra para training, sin mencionar que estos están categorizados para un procesamiento más simple.

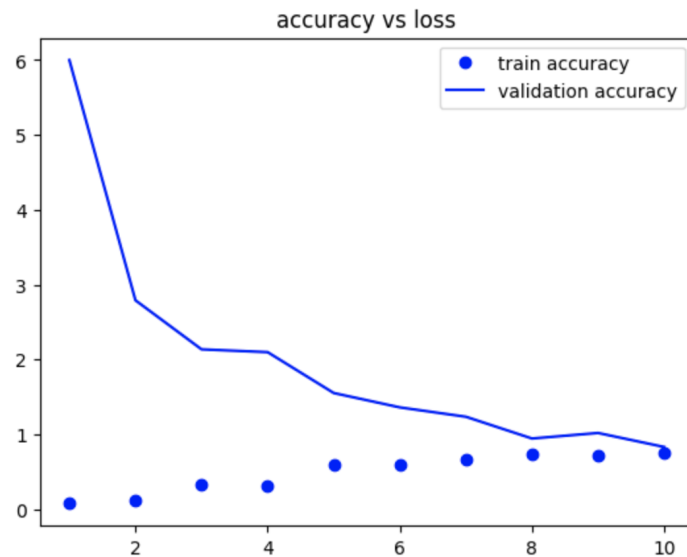
b. Uso de framework

Para el proyecto fueron utilizadas las librerías de tensorflow y keras para la implementación del modelo de aprendizaje profundo. TensorFlow y Keras son dos de las herramientas más populares en el campo del aprendizaje profundo y la inteligencia artificial por lo que gracias a su flexibilidad, facilidad de implementación, escalabilidad y la comunidad de apoyo existente hacen que sean la mejor opción para este tipo de ejercicios.

```
45 conv_base= VGG16(weights='imagenet',  
46 | | | | | include_top = False,  
47 | | | | | input_shape = (150,150,3))  
48  
49 model = models.Sequential()  
50 model.add(conv_base)  
51 model.add(layers.Flatten())  
52 model.add(layers.Dense(256,activation='relu'))  
53 model.add(layers.Dense(16,activation='softmax'))
```

c. Evaluación del desempeño

Debido al tipo de problema abordado donde la clasificación era de tipo multi-categoría se hizo uso de una métrica de evaluación propia de este tipo de problema: categorical accuracy. Esta métrica es comúnmente utilizada en modelos de aprendizaje profundo para problemas de clasificación multiclase como el abordado en el presente trabajo. Mide la proporción de predicciones correctas realizadas por el modelo en relación con el número total de muestras.

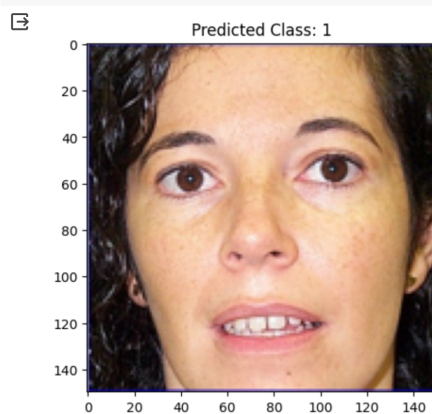


d. Generación de predicciones

Finalmente, se implementó una sección en el código de notebook donde se desarrolló el proyecto donde se puede probar el modelo desarrollado con imágenes específicas de manera que se pueden generar predicciones en base al modelo y de esta manera comprobar si esta es correcta o no.

```
[44]: 1 class_index = np.argmax(predictions)
      2

1 # evaluando la foto resultante
2 import matplotlib.pyplot as plt
3
4 # Display the image
5 plt.imshow(image)
6 plt.title(f'Predicted Class: {class_index}')
7 plt.show()
```



3. Referencias

- a. Set de datos:

<https://thinkingneuron.com/face-recognition-using-deep-learning-cnn-in-python/>

- b. Notebook:

https://colab.research.google.com/drive/1KRAXaMDvNL_cDygBZsBoC8W13Wq57gh7?usp=sharing