



Campus Monterrey

Inteligencia artificial avanzada para la ciencia de datos II (Gpo 501)

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

Francisco Castorena Salazar

A00827756

## Propósito

Este documento tiene como objetivo explicar el programa titulado `transfer_learning_2.py` el cual es un script con el propósito de realizar clasificación de imágenes de flores por medio de técnicas de transfer learning utilizando la biblioteca TensorFlow y Keras, a continuación se mostrará un reporte explicando el funcionamiento del código

### 1. Importación de Librerías:

- `tkinter`: Se utiliza para crear la interfaz gráfica de usuario (GUI) que permite cargar y predecir imágenes de flores.
- `PIL`: Proporciona herramientas para el manejo y procesamiento de imágenes.
- `tensorflow.keras`: Contiene herramientas para la construcción y entrenamiento de modelos de aprendizaje profundo.
- `matplotlib.pyplot`: Usado para visualizar la precisión y pérdida del modelo a lo largo de las épocas.

### 2. Descarga y Preparación del Conjunto de Datos:

- Se utiliza el conjunto de datos de flores disponible en TensorFlow. Este se descarga y se organiza utilizando `tf.keras.utils.get_file` y `tf.keras.preprocessing.image_dataset_from_directory`.
- El conjunto de datos se divide en conjuntos de entrenamiento y validación.

### 3. Construcción del Modelo de Transfer Learning:

- Se utiliza la arquitectura ResNet50 preentrenada para extraer características de las imágenes.
- Se congela la base preentrenada para evitar que se actualicen sus pesos durante el entrenamiento.
- Se agrega una capa de aplanado, seguida de dos capas densas para realizar la clasificación.
- Se compila el modelo utilizando la función de pérdida categórica cruzada y el optimizador Adam.

### 4. Entrenamiento del Modelo:

- Se entrena el modelo utilizando el conjunto de datos de entrenamiento y se valida con el conjunto de datos de validación.
- Se establece un número de épocas y se observa el progreso del modelo en términos de precisión y pérdida.

### 5. Interfaz Gráfica de Usuario (GUI):

- Se utiliza Tkinter para crear una interfaz amigable que permite cargar imágenes y realizar predicciones.

- La función `load_img` abre una ventana de selección de archivos, carga la imagen seleccionada y la redimensiona.
- La función `img_pred` realiza el preprocesamiento de la imagen y utiliza el modelo entrenado para hacer una predicción.
- La predicción se muestra en la interfaz gráfica.

#### 6. Uso de Recursos Externos:

- Se hace referencia a varios recursos externos utilizados en el desarrollo del modelo, como la documentación oficial de Keras, tutoriales de TensorFlow, un cuaderno de Jupyter para Transfer Learning con ResNet, y videos explicativos.