









# DETECTOR DE EMOCIONES

Proyecto Final – Visión Artificial 21/22

## Descripción breve

En este documento se presentará la memoria del trabajo en prácticas global de la asignatura de Visión Artificial y Reconocimiento de Patrones del curso 2021/2022.

Daniel Coronado Martín

El desarrollo de este proyecto se apoya principalmente en dos fuentes: el kernel kaggle “      Emotion Detection”<sup>1</sup> y el repositorio GitHub oficial de TensorFlow<sup>2</sup>.

El proceso de implementación y pruebas de este código fue como sigue.

## 1. LANZAR EL CÓDIGO DEL KERNEL

Con ello entender cada paso de imputación, preprocesamiento, transfer learning... del que hace uso su autor Sanskar Hasija.

Una vez lanzado el código obtenemos a través del comando `model.save('/kaggle/working/best_model.h5')` el modelo `best_model.h5`

## 2. CONVERTIR NUESTRO MODELO H5 A TFLITE

Esto lo haremos porque en Raspbian, al igual que en otros sistemas de 32 bits, la instalación de algunas librerías Python como TensorFlow se hace muy tediosa. Por tanto, nosotros utilizaremos como alternativa TensorFlow Lite y necesitaremos ese cambio en el formato del modelo.

## 3. INTRODUCIR LOS METADATOS AL MODELO

A continuación y una vez tengamos listo nuestro `best_model.tflite` necesitaremos introducirle mediante metadatos las etiquetas de la clase.

Eso lo haremos a partir del script `metadata_better_writer.py`<sup>2</sup>

## 4. EJECUCIÓN

Ya con todos los pasos anteriores completados, solamente nos queda probar nuestro modelo con la cámara de la Raspberry pi. Usaremos, para ello, el script `classify.py` que llama a su vez a `image_classifier.py`. Aquí se realiza todo el preprocesamiento de la imagen capturada por la Raspberry según los parámetros del propio modelo. Todo esto gracias a la clase `Interpreter` de la librería TensorFlow Lite<sup>3</sup>.

---

<sup>1</sup> “      Emotion Detection” - Kernel Kaggle: <https://www.kaggle.com/danicoronado00/emotion-detection>

<sup>2</sup> TensorFlow – Repositorio GitHub: <https://github.com/tensorflow/examples>

<sup>3</sup> TensorFlow Lite – Guía Oficial: <https://www.tensorflow.org/lite/guide/python>

## 5. ANEXO

Por último, me gustaría comentar algunas dificultades que he ido teniendo con este entorno nuevo que era para mí la Raspberry pi. Como ya comentaba en el punto segundo la instalación de librerías en Python para Raspberry pi no es siempre tarea sencilla, así que tuve que hacer uso de entornos virtuales Python que hasta ahora no conocía. Estos entornos te ayudan a tener una versión "limpia" de Python en la que poder instalar las librerías que quieras y trabajar de cero.

Además, precisamente en estos meses Raspbian está trabajando en mudar su API Raspicam a Libcamera con muchas mejoras y novedades puestas sobre el papel, pero con prácticamente todo el desarrollo todavía por delante. Así, por ejemplo, la compatibilidad con Python todavía es muy limitada o prácticamente nula. Así que también tuve que acceder al *sudo raspi-config* y activar la versión del legado de la cam.

## 6. CONCLUSIONES

Por tanto, y como conclusión, considero que he aprendido, ya no solo a implementar modelos más o menos eficaces en TensorFlow, sino también a trabajar en un entorno nuevo y de desarrollo contemporáneo como es Raspbian. Con todas las dificultades y toda la labor de investigación que esto supone.