Andrés Marín Galán

# Reconocimiento de Gemas

Se ha recogido un conjunto de imágenes de 87 tipos distintos de gemas para realizar la clasificación de las mismas. Se disponen de 2856 imágenes para train y 363 para test.

Primeramente se ejecutaría el programa **script_tamano.py** que estandariza el tamaño de todas las imágenes a 256x256 pixeles.
Tras esto el programa principal **program_v2.5.py** realiza la función principal de clasificación.
En primer lugar ya que no disponemos de un conjunto de validación se divide el conjunto de train dejando el 80% en train y el 20% en validación, mediante la función **Obt_val.**

En segundo lugar dado que las imágenes no son muy pesadas se ha optado por generar imágenes mediante Data Augmentation y guardarlas en memoria para así observarlas bien y poder determinar con exactitud los cambios que se realizan a las imágenes. Esto se realiza mediante la función **DataAug_save** se ha encontrado que los mejores parámetros para el data augmentation son:

- ☐ Flip vertical y horizontal
- ☐ Rotación de máximo 20 grados
- ☐ Movimiento de la imagen máximo 20% tanto para el ancho como para el alto
- ☐ Zoom del 20% como máximo
- ☐ Cambio de brillo de entre 50% y 90% del original

A partir de aquí se emplea un modelo de clasificación, se ha optado por una **EfficientNetB4** con un pooling max, tras esto un **BatchNormalization** una Densa de 256 neuronas con activación relu y un Dropout, por último una densa del número de clases con activación softmax. Se compila con Adamax y se usa un learning rate a 0.001 sin cambios.
Se usan 3 epoch porque tenían una duración de más o menos una hora por cada una y por dar con estas un buen valor de accuracy en validación, pero si se usan más epoch es posible que se mejore el valor obtenido.

```
Epoch 1/3
522/522 [==============================] - 3864s 7s/step - loss: 7.2919 - accuracy: 0.5738 - val_loss: 4.5127 - val_accuracy: 0.7028 - lr: 0.0010
Epoch 2/3
522/522 [==============================] - 3773s 7s/step - loss: 2.8840 - accuracy: 0.8935 - val_loss: 2.3930 - val_accuracy: 0.7640 - lr: 0.0010
Epoch 3/3
522/522 [==============================] - 3860s 7s/step - loss: 1.3881 - accuracy: 0.9595 - val_loss: 1.6933 - val_accuracy: 0.8042 - lr: 0.0010
```

Como vemos el mejor valor de accuracy en training es del 95.59%, en validación **80.42**%.

```
11/11 [==============================] - 23s 2s/step - loss: 1.7588 - accuracy: 0.7631
```

El accuracy en test es de **76.31**%

## Bibliografía

Paguina de keras con los datos: https://www.kaggle.com/lsind18/gemstones-images
Inspirado en: https://www.kaggle.com/gpiosenka/augment-and-balance-dataset-f1-score-81