

## Punto 1.

### Primera aproximación

A priori el planteamiento del enunciado podría parecer una buena idea suponiendo que el patrón de tono del color de un girasol fuera claramente amarillento y diferenciado del de las rosas. Con esta característica de imágenes podríamos pensar en un etiquetado acertado sobre las imágenes de girasoles y suponer que el resto de imágenes son de rosas. Sin embargo dada la variedad de imágenes de girasoles la utilización de una sola característica de entrada **auguramos daría muy malos resultados** si no llevamos a cabo una alternativa.

### La importancia de las ‘features’

¿Es realmente suficiente una característica como el tono amarillento de una imagen para determinar si es o no un girasol? Sin necesidad de llevar implementar este código intuimos que **es realmente poca información**. Podríamos simplemente, pensar que una imagen tiene que tener un porcentaje de sus píxeles más o menos amarillos para que fuese un girasol y hacer que esta condición fuese necesaria y suficiente para etiquetar las imágenes. Intuimos que este proceso daría una baja tasa de acierto.

Hemos de tener en cuenta la importancia de la extracción de características de una imagen. En el campo del tratamiento digital de imágenes **previo al ML**, la segmentación de objetos, reconocimiento de caracteres, seguimiento de caras etc. era un proceso bastante rudimentario. Existían algunas herramientas que utilizaban las diferentes capas de color, otras que aplicaban unos pocos filtros e incluso algoritmos más o menos primitivos para encontrar formas muy concretas como círculos/elipses, rectángulos/cuadrados, patrones verticales/horizontales...

La implementación redes neuronales y la mayor capacidad de procesamiento hicieron que el tratamiento digital de imágenes diera un salto cualitativo. La clave principal de dicho cambio radica en la capacidad de estas redes de extraer casi infinitas características mucho más allá de cómo podría pensar un humano. Ponderar y comparar estas características abstractas en multitud de imágenes para, en última instancia, dar una respuesta “humana”.

---

## Clusterización por color amarillo de girasol.

El proceso propuesto en el enunciado consiste en utilizar el color amarillo para determinar si una imagen es o no un girasol.



Entendiendo que queremos extraer la información del color de cada imagen podríamos elegir el mejor espacio de color en el que trabajar. De entre todos los que conocemos quizás el más interesante podría ser el **espacio de color CMYK** ya que independiza la componente de color amarillo de la imagen. Estimaríamos a continuación cantidad de 'amarillo-girasol' que hay realmente en la imagen. Una vez asignado un valor de porcentaje de píxeles de color 'amarillo-girasol' respecto al total de píxeles de la imagen, según la propuesta de nuestro compañero de equipo apasionado por la botánica, cabría determinar un umbral a partir de cual consideraríamos que se trata de un girasol.

Esta propuesta es realmente pobre por varios motivos. Entre los más importantes:

1. Existen imágenes de girasoles ocupando **diferentes porcentajes** de la imagen. Esto imposibilita encontrar un umbral consistente.
2. Tal y como dice el enunciado, **existen tipos de rosa de color amarillo** que, a todas luces, serían etiquetados como girasoles.
3. En el dataset de imágenes proporcionado existe un considerable porcentaje de **imágenes que no son ni rosas ni girasoles** y que, según el etiquetado propuesto, serían marcadas como unas u otras en función de la componente de color amarillo.

## Clusterización Alternativa.

Utilizar los píxeles en bruto como features no nos dará buenos resultados ya que la información de una imagen está contenida en la imagen completa. Extraer una por una cada una de estas variables con filtros (ya sea de color, escala de grises, formas...) nos daría resultados muy limitados.

La propuesta para solventar este problema podríamos resumirla en los siguientes pasos:

---

1. **Pasar las imágenes a través de una CNN pre-entrenada para extraer features.** Utilizar la última capa de la CNN (justo antes de la capa totalmente conectada) para extraer las features de la imagen.

2. **Reducir el número de features** al menor número posible sin perder información relevante para resolver el problema.

3. **Clustering:** Una vez obtenidas las features de cada imagen podemos agruparlas para realizar el etiquetado.

## Punto 2.

Archivo adjunto en código Python.

## Punto 3.

### Enriquecimiento de dataset desde clustering

Inicialmente agruparíamos todo el conjunto de imágenes del dataset utilizando nuestro algoritmo de clusterización. Para cada clúster resultante, se selecciona un representante, por ejemplo, el centroide. Además de las features originales, podríamos utilizar estos **representantes para calcular nuevas características** utilizando una medida de distancia determinada. Tanto las características originales como las que podríamos añadir formarían un dataset aumentado con el que entrenar el clasificador. Habría además que **almacenar el conjunto de representantes** para transformar los futuros datos no etiquetados para extraer estas nuevas features.

Cuando queramos etiquetar un nuevo dato, se evalúa su distancia con respecto a cada uno de los representantes de cada clúster y se utiliza para calcular las nuevas features.

### Refinar el método de etiquetado

Podríamos realizar el proceso de clusterización y etiquetado de todo nuestro dataset. Entendemos que en su mayoría está bien etiquetada y que por lo tanto podría entrenar de manera eficiente un **Clasificador Multicapa**.

Para poder afinar con más detalle este etiquetado podríamos pensar en entrenarlo con nuestro dataset dejando una imagen fuera de dicho entrenamiento. A continuación este **algoritmo entrenado con N-1 podría etiquetar la imagen que hemos dejado fuera** y

---

comparar el resultado con el proceso de clustering. En función de los resultados de probabilidad estamos capacitados de afinar la decisión cluster.

Este proceso podría implicar la **elaboración de N modelos para refinar N imágenes**. Para aligerar este proceso, ¿habría mucha diferencia entre un modelo entrenado con N-1 imágenes y otro entrenado con todas? Evidentemente no debería haberla. Deberemos, eso sí, **evitar que nuestro clasificador memorice las imágenes**. Para ello tendremos que intentar elaborar un modelo con la menor cantidad de capas posibles.

## Punto 4.

Archivo adjunto en código Python.