Detección automática de estructuras anómalas en imágenes arbitrarias

## Introducción:

La detección automática de estructuras anómalas en imágenes arbitrarias se ocupa del problema de encontrar patrones que non se adecuan a la normalidad de una imagen. Este problema plantea un desafío en visión por computador, ya que no existe una definición clara y directa de lo que es (a)normal para una imagen arbitraria dada. La detección automática de anomalías tiene una gran importancia en la industria, la teledetección, la videovigilancia y la medicina. Esta medida de rareza generalmente se llama mapa de saliencia. Requiere un umbral empírico para decidir qué píxeles o regiones son lo suficientemente destacados como para llamarse anomalías. Asumiremos que las anomalías se pueden detectar en y desde una sola imagen, o desde un pequeño conjunto de imágenes. Aprender el modelo de *normal* en imágenes que contienen anomalías implica que estas sean pequeñas en proporción con el tamaño de las imágenes procesadas.

### Tareas a realizar

De entre las distintas aproximaciones del estado de la técnica, nos centraremos en modelos de saliencia visual, y métodos basados en *self-supervised learning*. En la carpeta Código puedes encontrar nuestros dos modelos saliencia (AWS y WMAP), y el código con un ejemplo de uso de uno de los modelos (basado en residuo espectral) que viene con las últimas versiones de OpenCV. Estos modelos habría que dotarlos de una etapa final de clasificación binaria a nivel de píxel.

En cuanto a las técnicas de *self-supervised learning*, la idea sería implementar el método, basado en *inpainting*, descrito en uno de los artículos de la correspondiente carpeta de Referencias. Para realizar el entrenamiento se puede utilizar un dataset genérico y también imágenes de las subcarpetas de la carpeta Datasets. Parte de estas imágenes vienen con su *ground-truth* y otras no. No hay muchos datasets de imágenes de anomalías públicos. Podemos trabajar con lo que hay básicamente para evaluar los métodos. Los casos que no tienen el *ground-truth* asociado se pueden emplear para entrenar con la idea del *self-supervised learning*.

### Plan de trabajo

La primera semana la puedes aprovechar para instalarte, leer los trabajos en las dos carpetas de Referencias (algunas son introductorias del problema y de revisiones, que se pueden utilizar para hacer la presentación final), y probar el código de la carpeta Código.

La segunda semana podrías ponerte con la implementación del método basado en *inpainting*, y comenzar con el entrenamiento, que continuaríamos la siguiente semana.

En la cuarta semana podríamos ir viendo los resultados y construyendo las tablas/gráficas asociadas.

Los últimos días estarían dedicados a preparar la presentación del trabajo realizado.