

Introducción a la visión computacional

Andrés Escobar Cotán



ÍNDICE

1. ¿Qué es la visión computacional?
2. Pipeline básico
3. Ejemplo práctico usando OpenCV

ÍNDICE

1. ¿Qué es la visión computacional?

1. Definición

2. ¿Qué veo yo, qué ve el ordenador?

3. Áreas de aplicación

4. Modelo de color

5. Problema matemático/estadístico

¿QUÉ ES LA VISIÓN COMPUTACIONAL?

- DEFINICIÓN



La **Visión Computacional** (o **visión artificial**) , es una disciplina que trata de adquirir, procesar, analizar y comprender las imágenes del mundo real con el fin de producir información numérica o simbólica para que puedan ser tratados por un ordenador.

Es decir, es un campo que a partir de una imagen intenta extraer información susceptible de ser interpretada y tratada numéricamente.

Imagen de test de Lena Söderberg, recorte de Playboy Noviembre 1972[1]

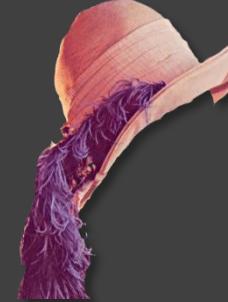
¿Qué veo yo, qué ve el ordenador?



Foto completa



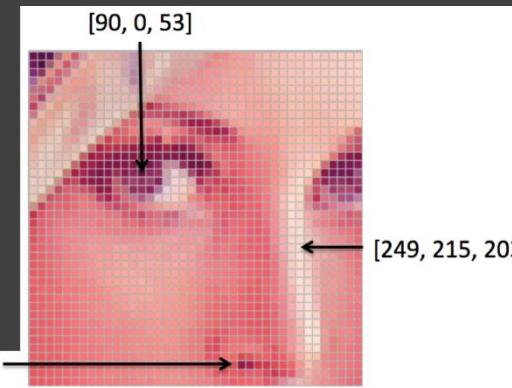
Una modelo



El sombrero con
plumas

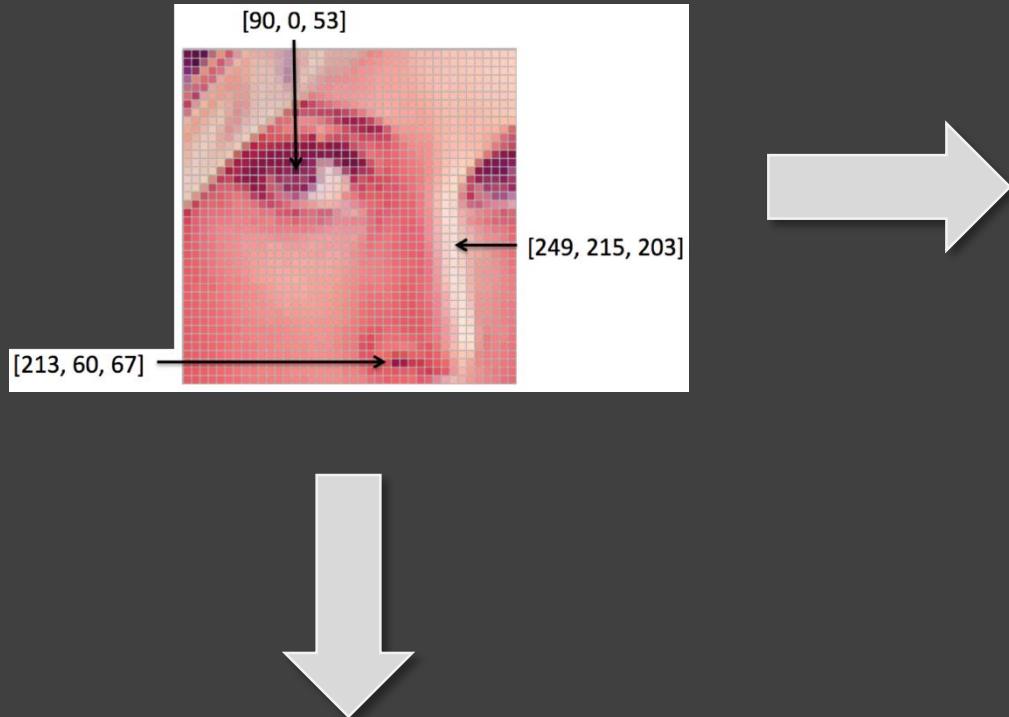


Foto completa



Representación
matricial (RGB)

¿Qué veo yo, qué ve el ordenador?



Una imagen, desde el punto de vista de un TAD , es una matriz (**H** x **W**) donde:

- **W o (Width)** , es el ancho de la imagen o número de columnas de la matriz, su unidad es px o pixel.
- **H o (Height)**, es la altura de la imagen o número de filas de la matriz, su unidad es px o pixel.

A su vez cada celda (o pixel) , dependiendo del modelo de color en el que esté, tendrá los siguientes valores (8 bits):

- Si la imagen es en escala de grises, un único entero comprendido entre [0 ... 255], donde 0 es el negro absoluto, y 255 el blanco.
- Si la imagen es en color, una tupla de tres valores, todos comprendido entre [0..255], donde el primer valor de la tupla se corresponde al Rojo, el segundo valor al Verde, y el tercero al Azul.

¿Qué veo yo, qué ve el ordenador?



Un video, desde el punto de vista de un TAD , va a ser un array de imágenes, donde a mayores está **el factor FPS: Frames por segundo**, que es el número de capturas (imágenes) que hay por cada segundo.

A diferencia de la imagen, es más susceptible al ruido y a artefactos de compresión

Áreas de Aplicación

La visión computacional, tiene su aplicación en multitud de áreas:

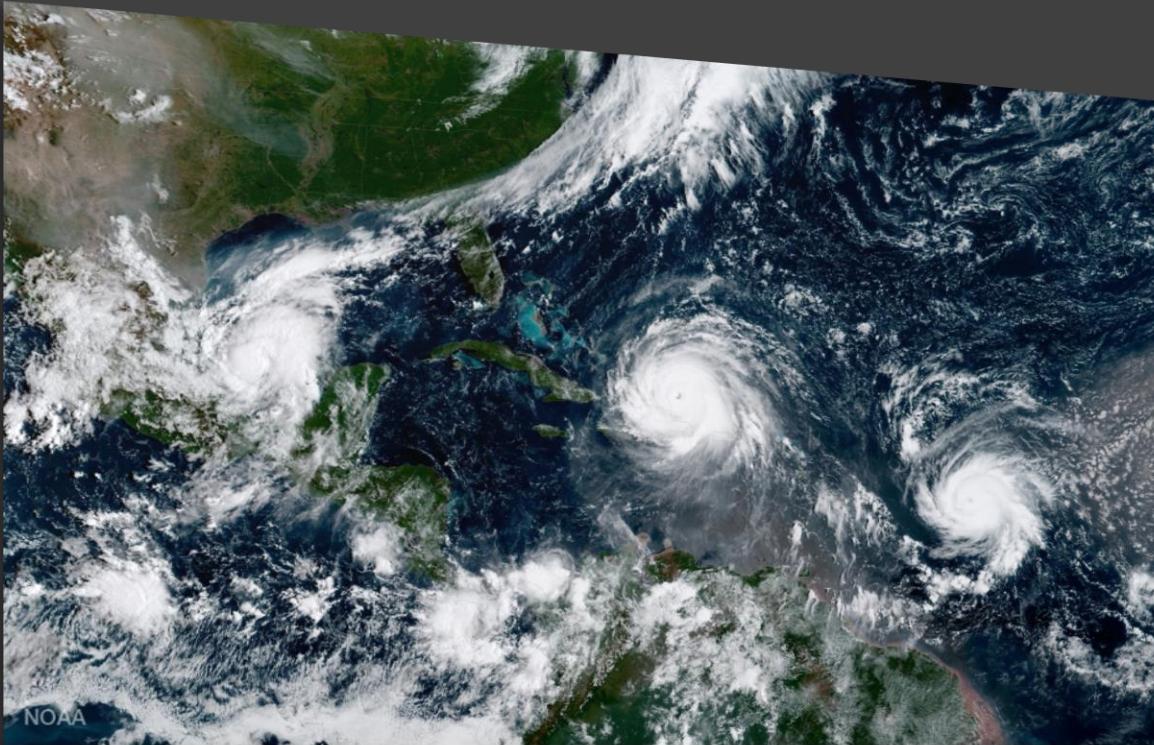


Imagen de dos huracanes aproximándose a la costa este
de E.E.U.U. [2]

Imágenes satelitales:

- Control del clima
- Monitorización agrícola
- Diseño de mapas
 - Gestión de desastres
- Seguridad nacional

Áreas de Aplicación

La visión computacional, tiene su aplicación en multitud de áreas:

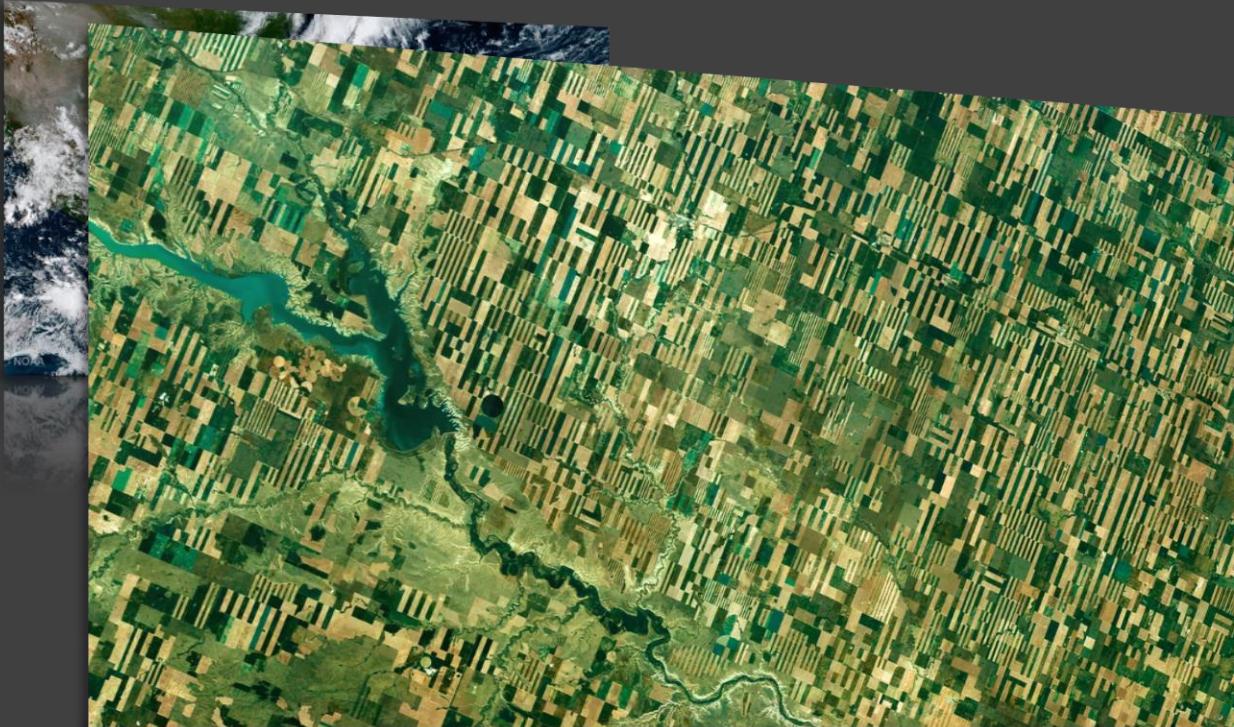


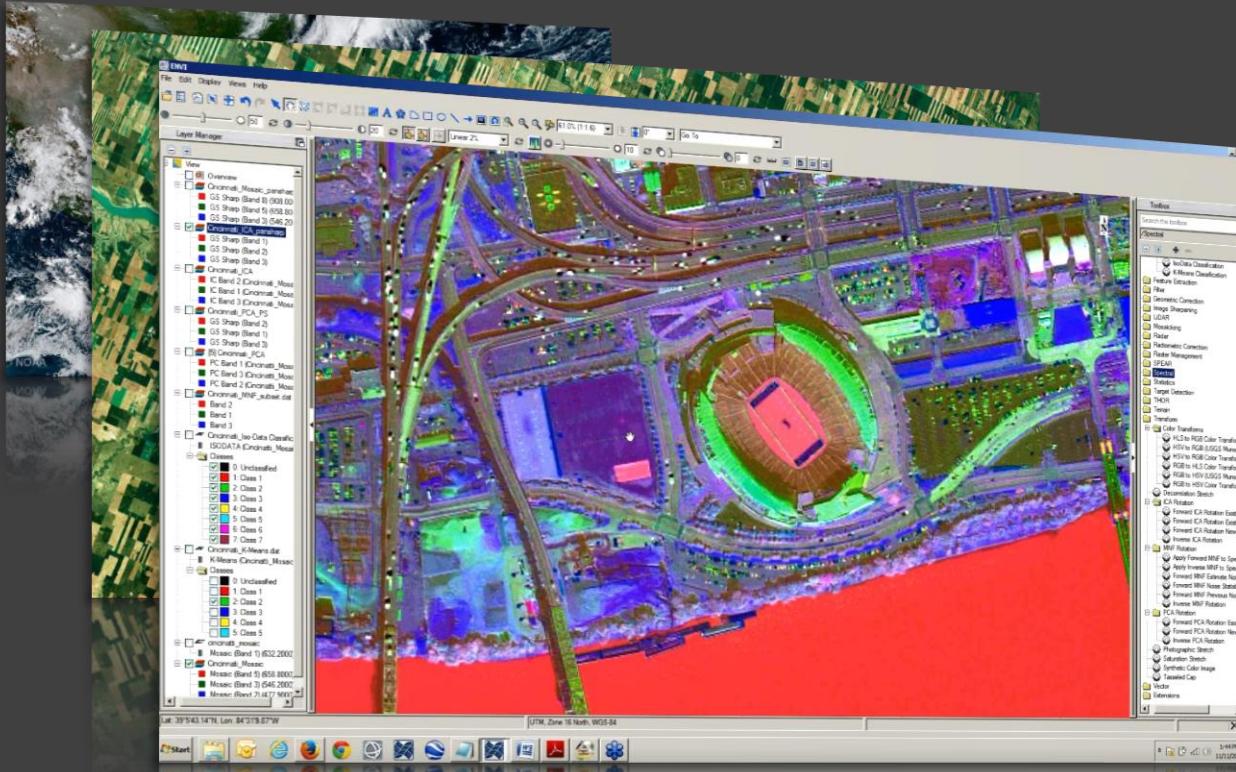
Imagen del 45% de campos de trigo en Montana
capturado por la camara Theia de UrtheCast [3]

Imágenes satelitales:

- Control del clima
- Monitorización agrícola
- Diseño de mapas
 - Gestión de desastres
- Seguridad nacional

Áreas de Aplicación

La visión computacional, tiene su aplicación en multitud de áreas:



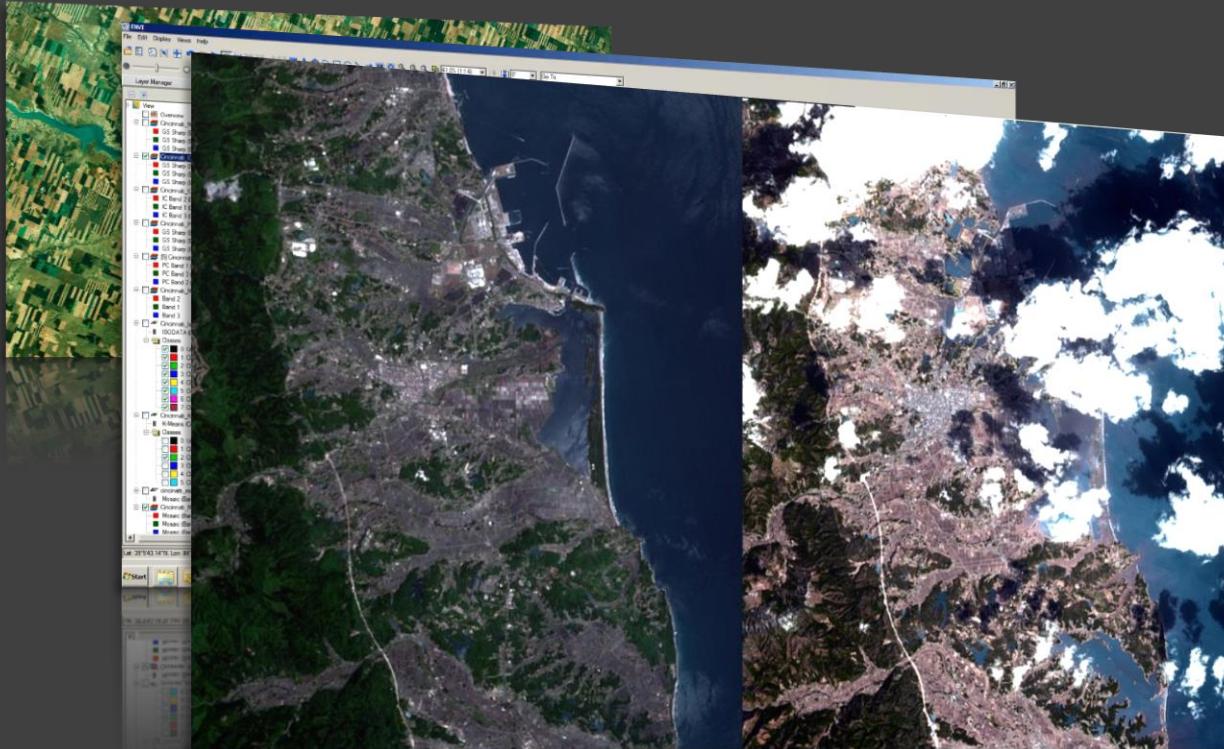
Captura del programa ENVI, para análisis de imágenes sobre una imagen de LandSat8 [4]

Imágenes satelitales:

- Control del clima
- Monitorización agrícola
- Diseño de mapas
 - Gestión de desastres
 - Seguridad nacional

Áreas de Aplicación

La visión computacional, tiene su aplicación en multitud de áreas:



Capturas del satélite español Deimos-1, antes y después del tsunami de Japón de 2011 [5]

Imágenes satelitales:

- Control del clima
- Monitorización agrícola
- Diseño de mapas
 - Gestión de desastres
- Seguridad nacional

Áreas de Aplicación

La visión computacional, tiene su aplicación en multitud de áreas:

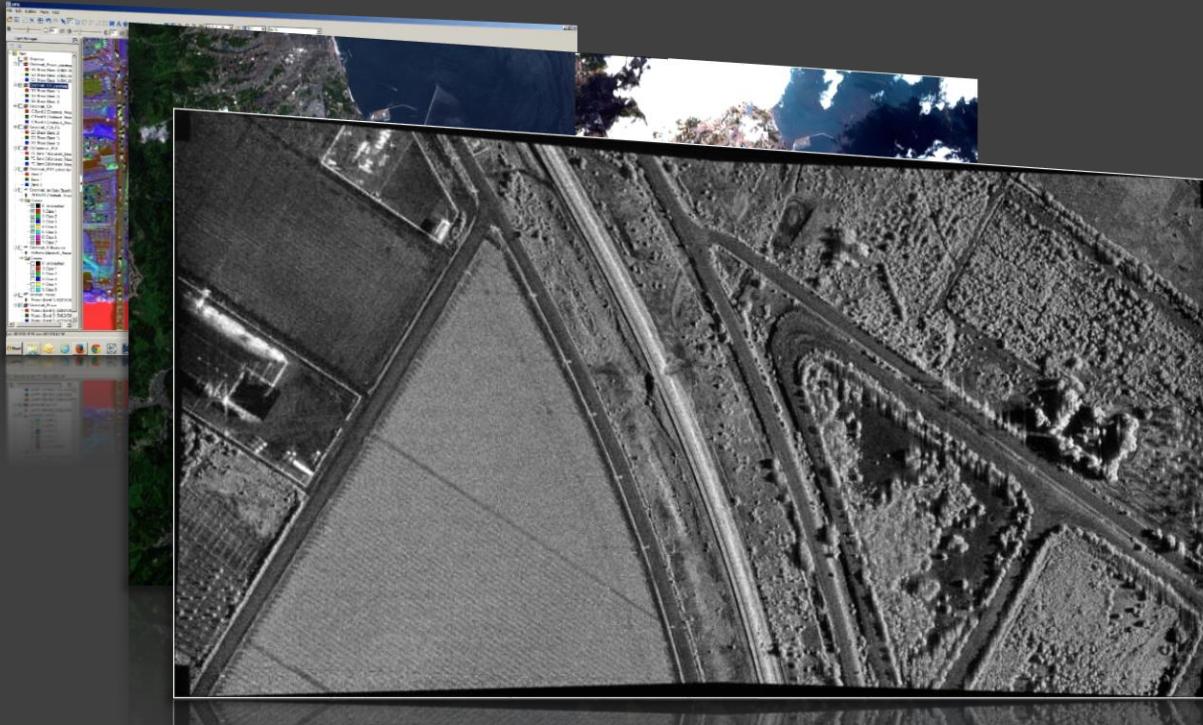


Imagen SAR de Lynx [6]

Imágenes satelitales:

- Control del clima
- Monitorización agrícola
- Diseño de mapas
 - Gestión de desastres
- Seguridad nacional

Áreas de Aplicación

La visión computacional, tiene su aplicación en multitud de áreas:

Arte y asociados:

- **Reconstrucción 3D [7]**
- Color Transfer

Medicina:

- Reconocimiento de patologías
- Ecografías

Inteligencia Artificial:

- Vehículos autónomos
- Reconocimiento de individuos



Áreas de Aplicación

La visión computacional, tiene su aplicación en multitud de áreas:

Arte y asociados:

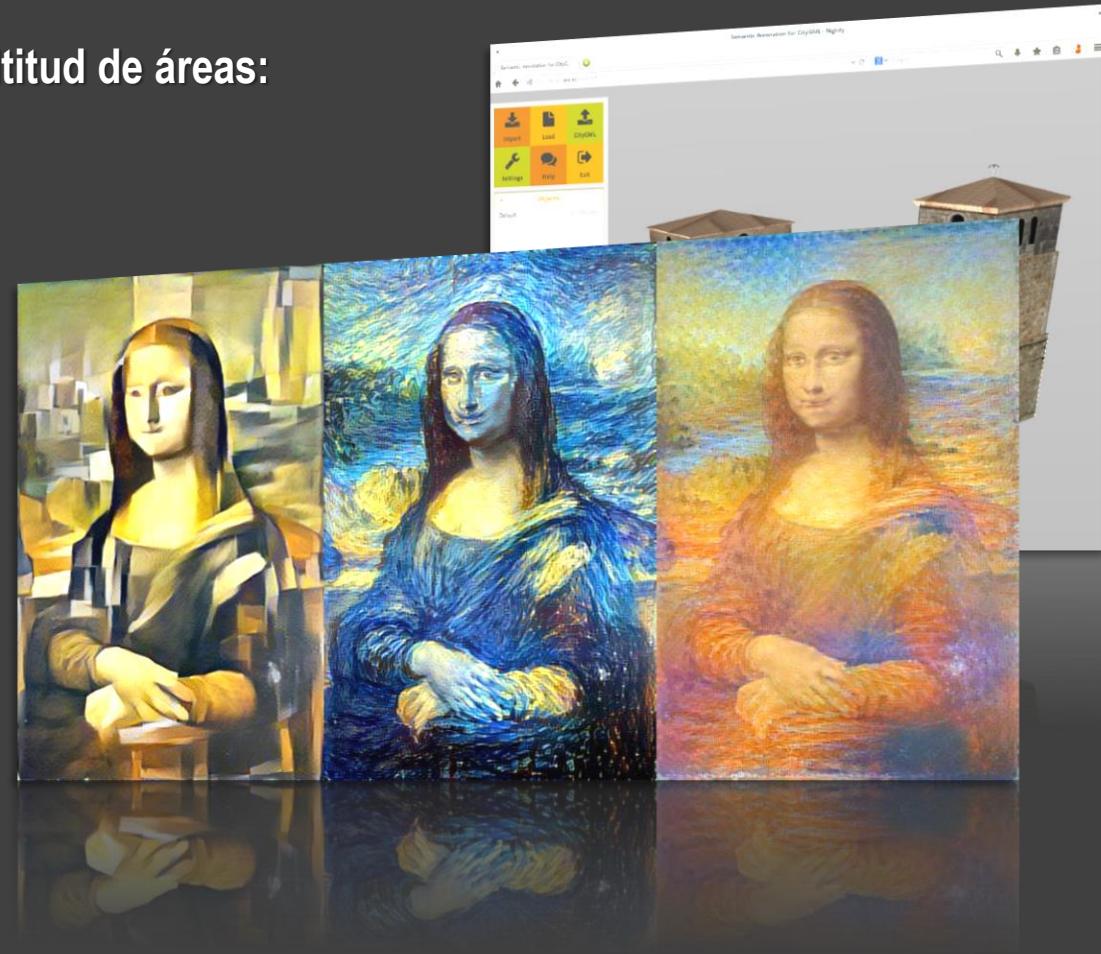
- Reconstrucción 3D
- Color Transfer [8]

Medicina:

- Reconocimiento de patologías
- Ecografías

Inteligencia Artificial:

- Vehículos autónomos
- Reconocimiento de individuos



Áreas de Aplicación

La visión computacional, tiene su aplicación en multitud de áreas:

Arte y asociados:

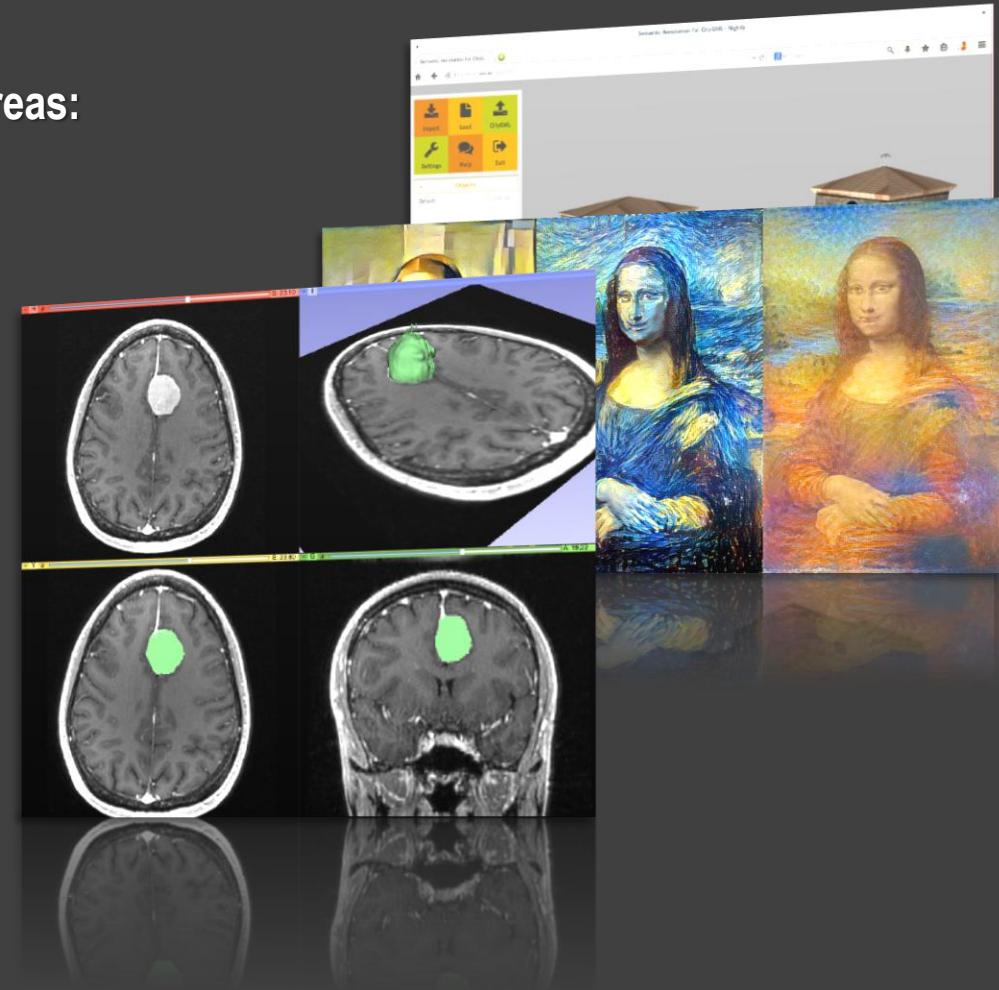
- Reconstrucción 3D
- Color Transfer

Medicina:

- **Reconocimiento de patologías [9]**
- Ecografías

Inteligencia Artificial:

- Vehículos autónomos
- Reconocimiento de individuos



Áreas de Aplicación

La visión computacional, tiene su aplicación en multitud de áreas:

Arte y asociados:

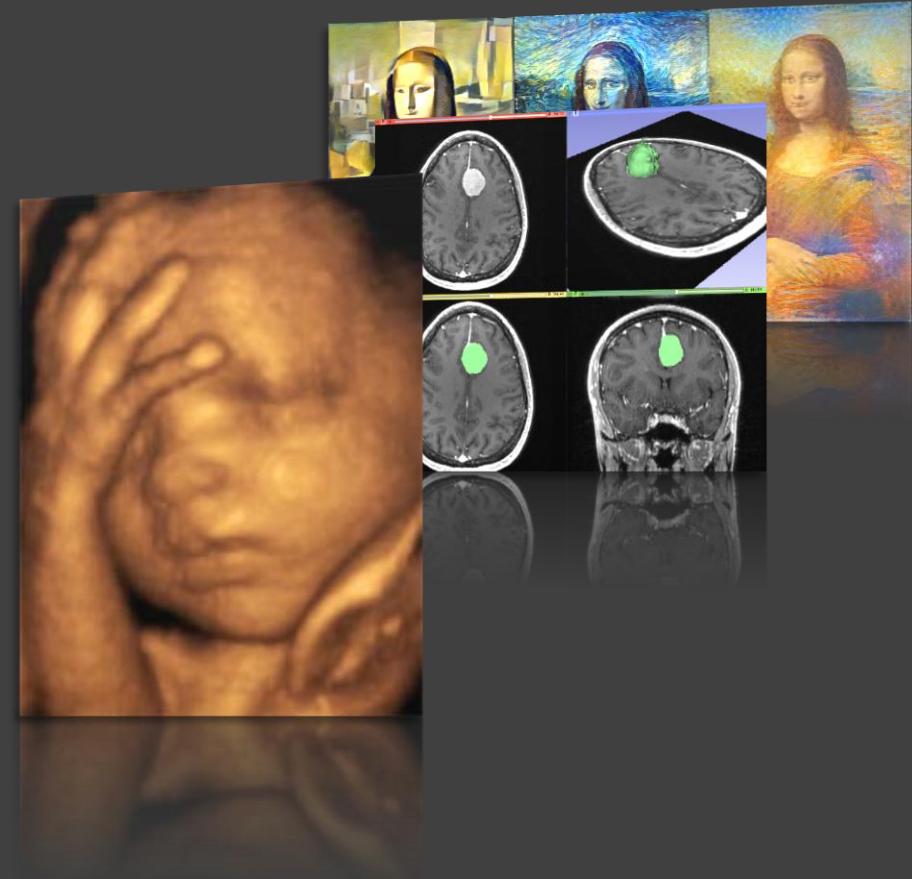
- Reconstrucción 3D
- Color Transfer

Medicina:

- Reconocimiento de patologías
- Ecografías [10]

Inteligencia Artificial:

- Vehículos autónomos
- Reconocimiento de individuos



Áreas de Aplicación

La visión computacional, tiene su aplicación en multitud de áreas:

Arte y asociados:

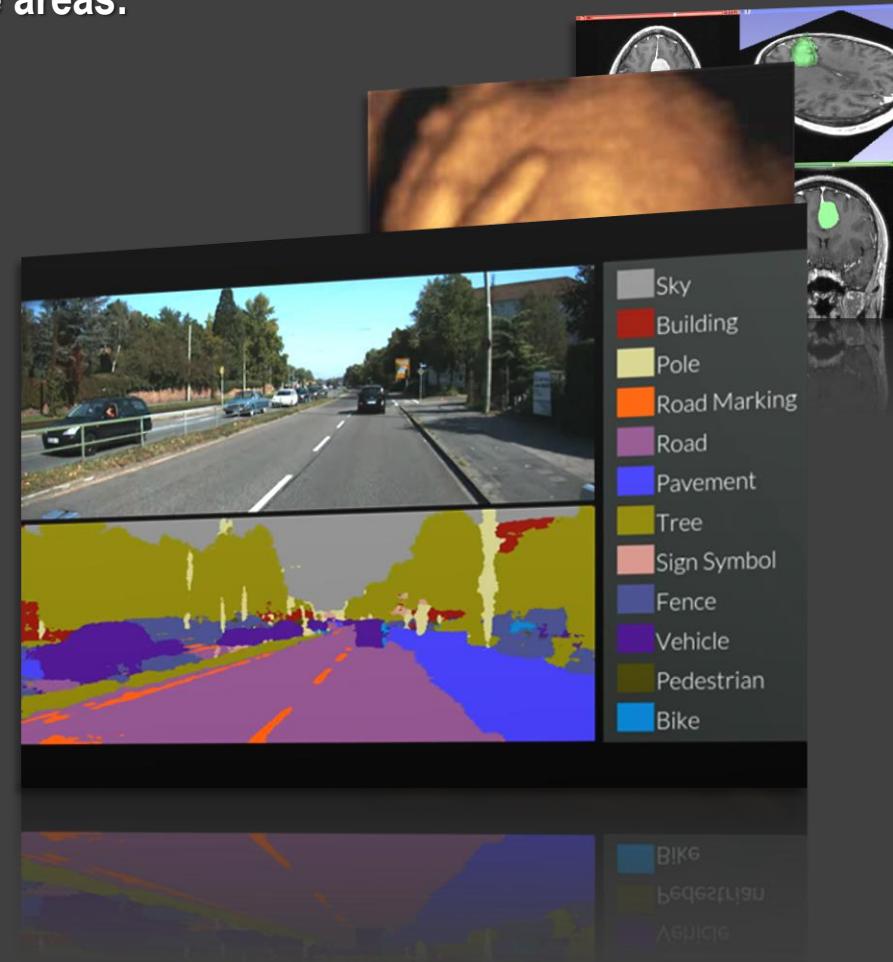
- Reconstrucción 3D
- Color Transfer

Medicina:

- Reconocimiento de patologías
- Ecografías

Inteligencia Artificial:

- Vehículos autónomos [11]
- Reconocimiento de individuos



Áreas de Aplicación

La visión computacional, tiene su aplicación en multitud de áreas:

Arte y asociados:

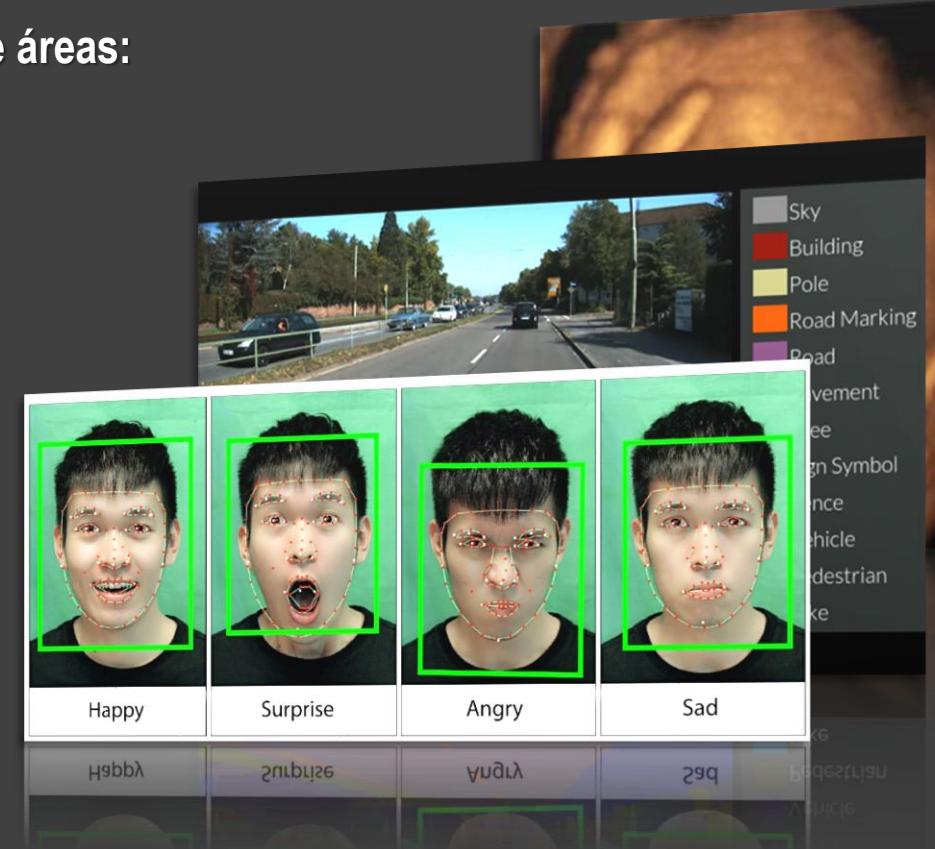
- Reconstrucción 3D
- Color Transfer

Medicina:

- Reconocimiento de patologías
- Ecografías

Inteligencia Artificial:

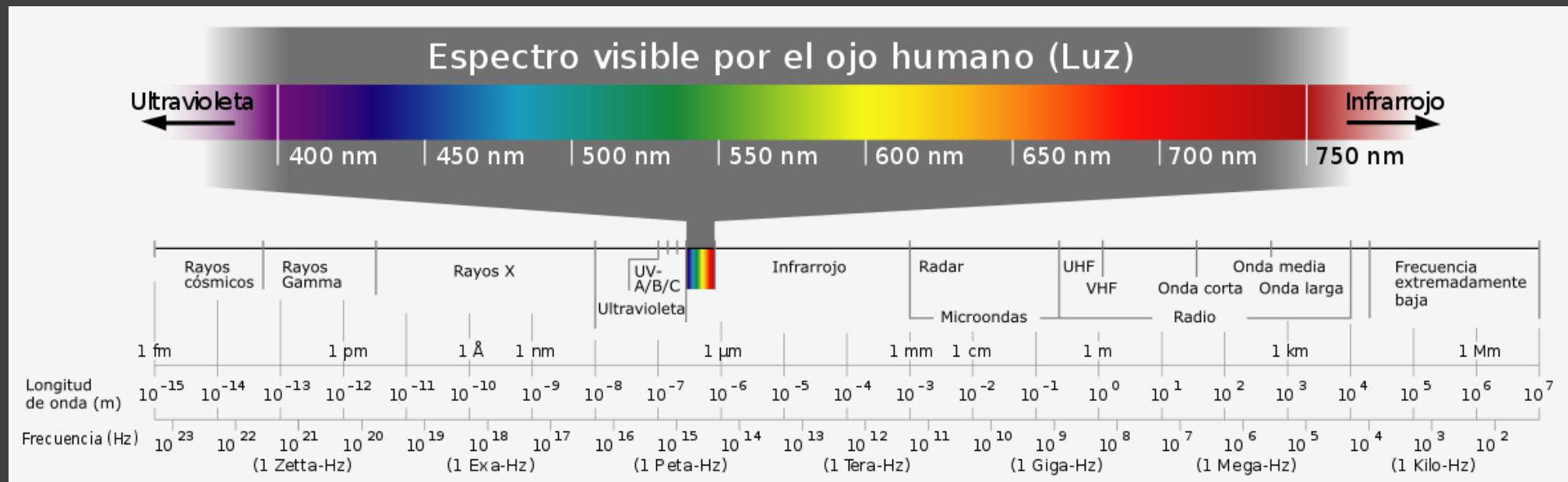
- Vehículos autónomos
- **Reconocimiento de individuos [12]**



Modelos de color

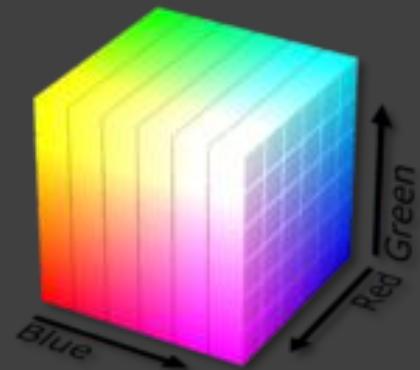
La luz es la parte de la radiación electromagnética que puede ser percibida (tiene componente corpuscular y ondulatoria) por el ojo. [13]

El espectro visible y no visible, de la luz es el siguiente:

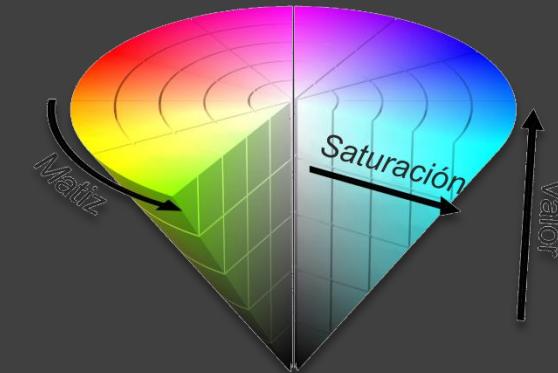


Modelos de color

A la hora de traducir el color a un ordenador (para su tratamiento y visualización), se necesita establecer un modelo de color:



RGB (Red Green Blue) [14]



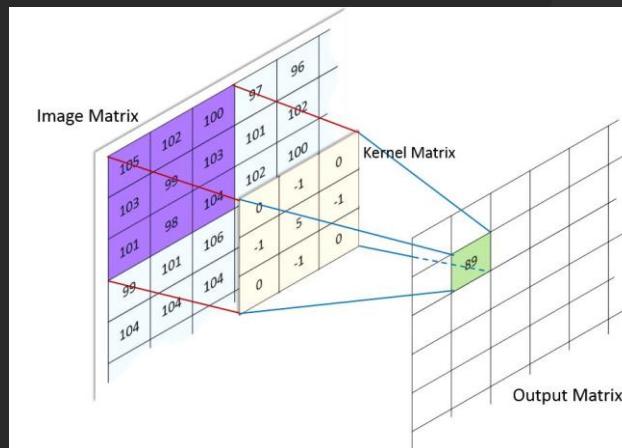
HSV (Hue Saturation Value) [15]

Convolución atómico/estadístico

La **convolución** es el proceso matemático [17] por el que se le **aplica un filtro a una imagen**:

- Precisa de un **kernel**, esto es una matriz simétrica MxM (donde M es un número impar)
 - Se **recorre la imagen**, como si fuera una **matriz solapando** cada **pixel** con el **centro del kernel** (los bordes tienen un comportamiento diferente):

- Se multiplica el número de la imagen por su correspondiente número en la matriz.
 - Se suman los resultados, y el resultante se sustituye en el centro de la imagen

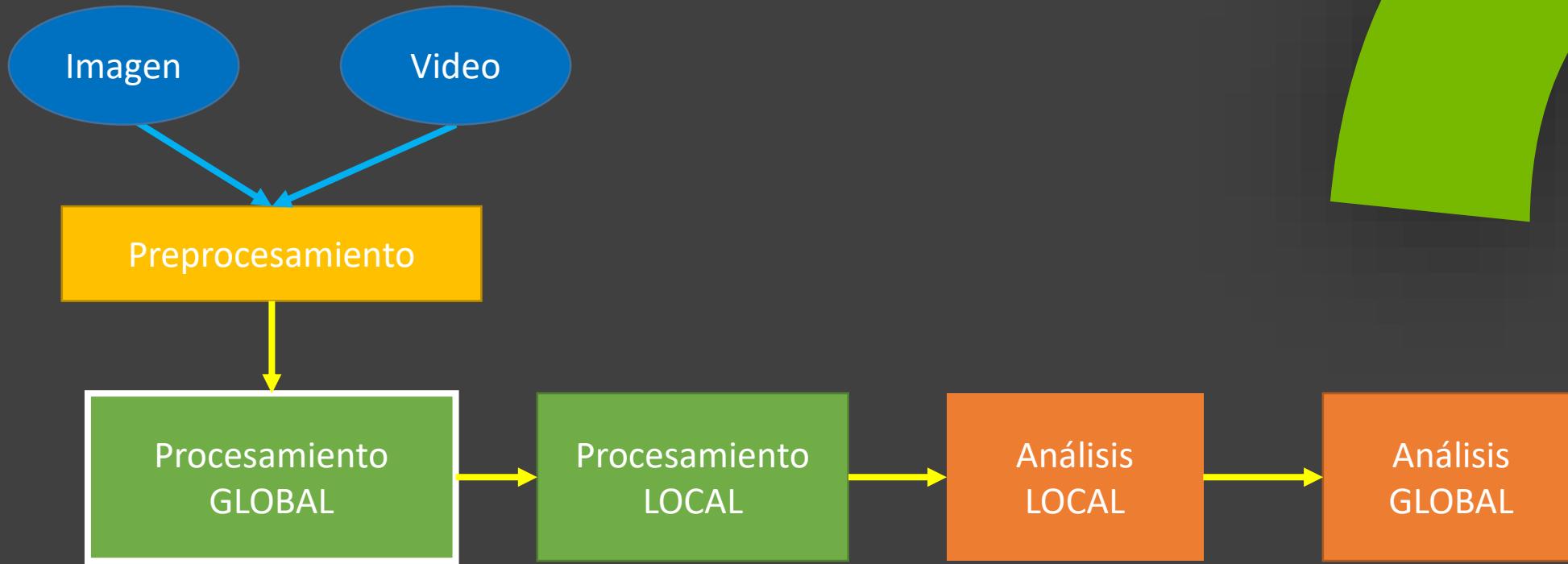


ÍNDICE

2. Pipeline Básico

1. Procesamiento Global
2. Procesamiento Local
3. Análisis Local
4. Análisis Global

Procesamiento Global

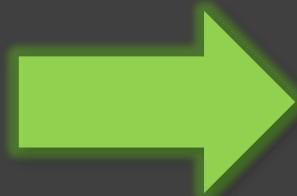


Procesamiento Global

El procesamiento global de imagen es una colección de **métodos y herramientas** que se aplican a toda la imagen por igual.

El **objetivo de esta etapa del pipeline**, va a consistir en **eliminar** aquella información de la imagen que **no nos interese**, y segmentar la que si.

Para ello se presentan las siguientes herramientas, basadas en **operaciones aritméticas**, en **convoluciones** y en **derivadas** de primer orden.



Filtros básicos (Operaciones Aritméticas) [18] [19]

Imagen A



$A + B$



$A - B$



Imagen B



Imagen A (negativa)



$A * B$

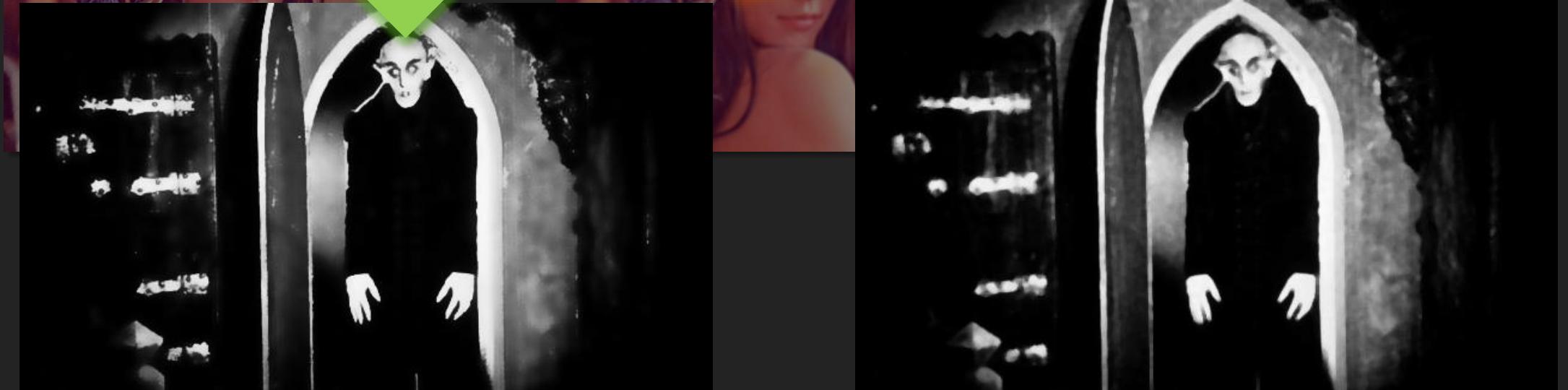
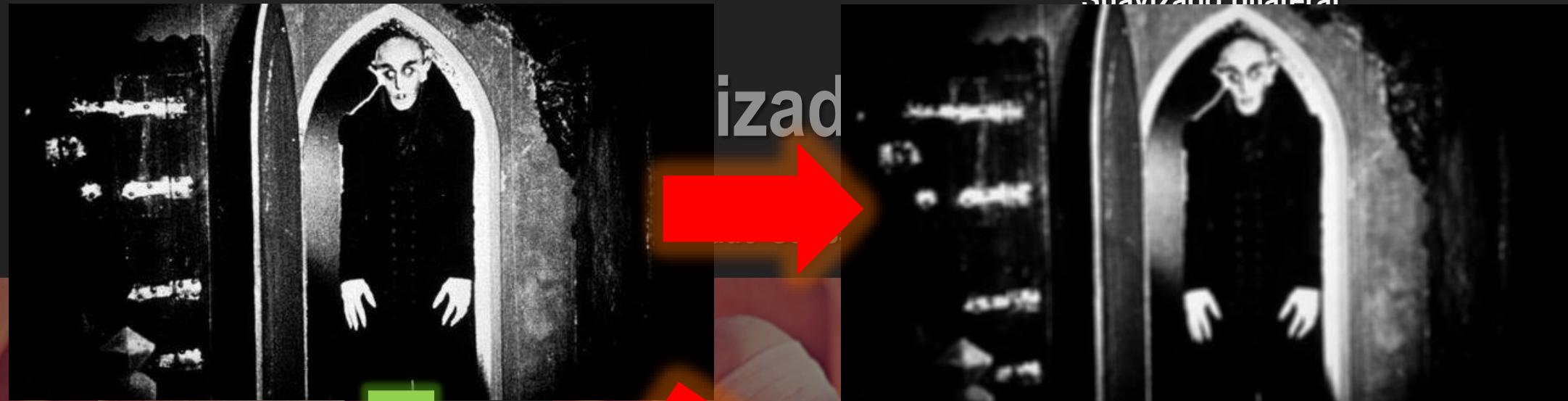


A / B

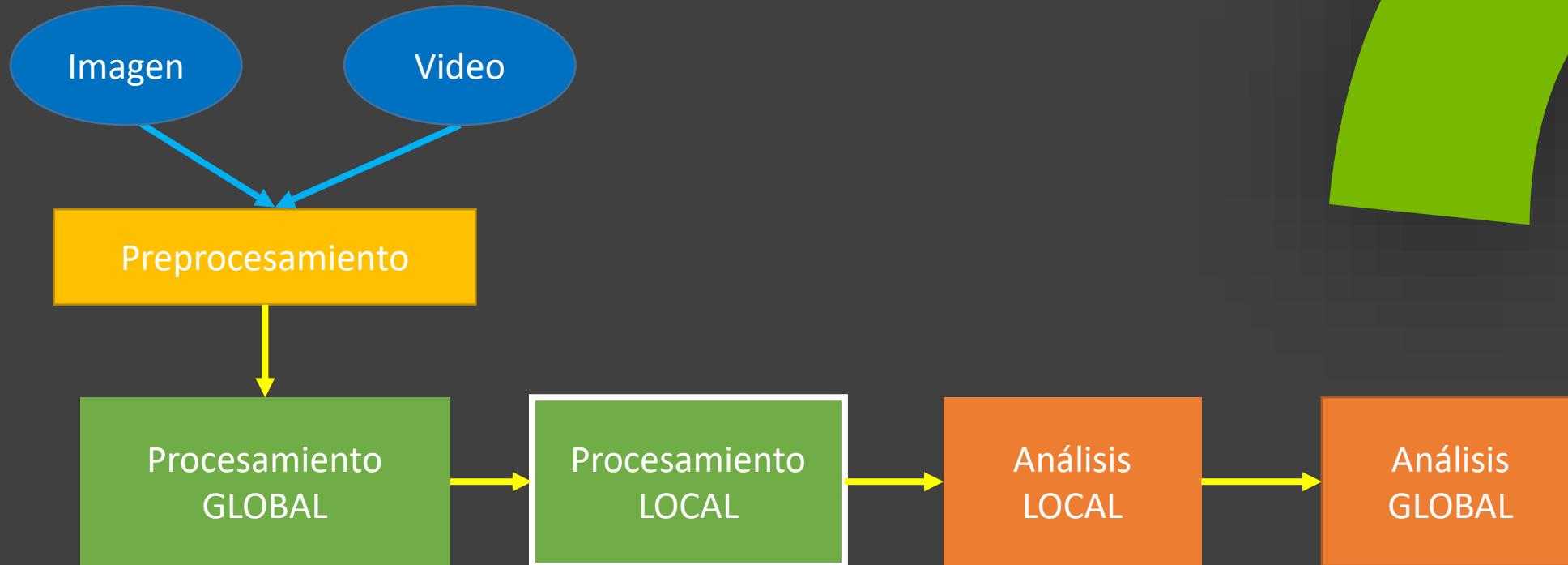


Suavizado bilateral

izad.



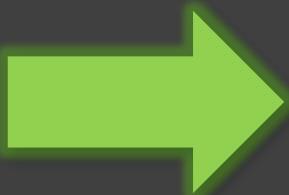
Procesamiento Local



Procesamiento Local

El procesamiento local a diferencia del global pretende:

- Detectar, acentuar y describir **hechos** o discontinuidades.
- Simplificar la imagen, mediante la descripción de los hechos citados anteriormente.
- Se trata por tanto de a partir de datos **radiométricos** crear datos **geométricos**:
 - **Puntos** (1D)
 - **Minirectas o Segmentos** (2D)



Filtros básicos (Enmascarado) [21]

Imagen normal



Umbralizado básico (canal azul)



Umbralizado adaptativo(canal azul)



Filtros básicos (Extracción de bordes) [22]

Imagen normal



Laplace



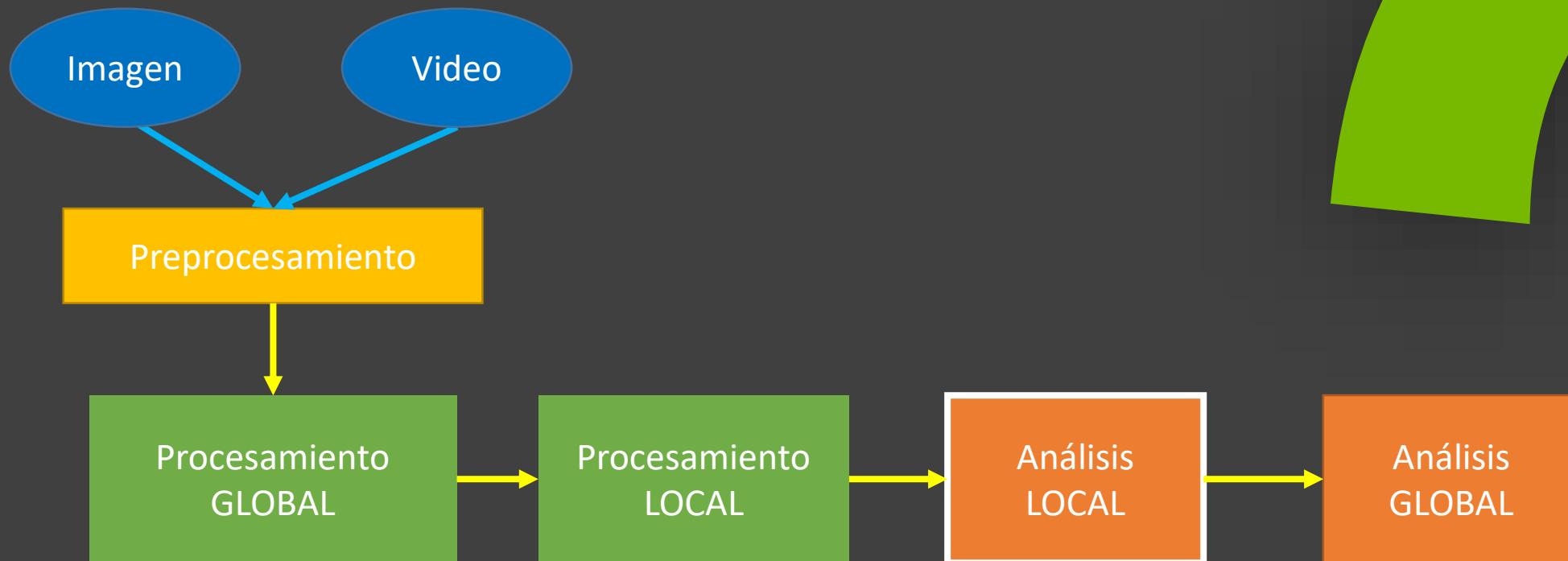
Sobel



Canny



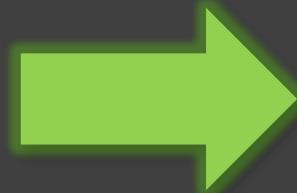
Análisis Local



Análisis Local

El análisis local pretende a raíz de los hechos geométricos descubiertos en el caso anterior:

- Reconstruir más segmentos, mediante la agrupación de los mismos y discriminación de ángulos.
- Detectar en función de los segmentos, junturas estables y detección de puntos de fuga
 - Puntos (1D)
 - Minirectas o Segmentos (2D)

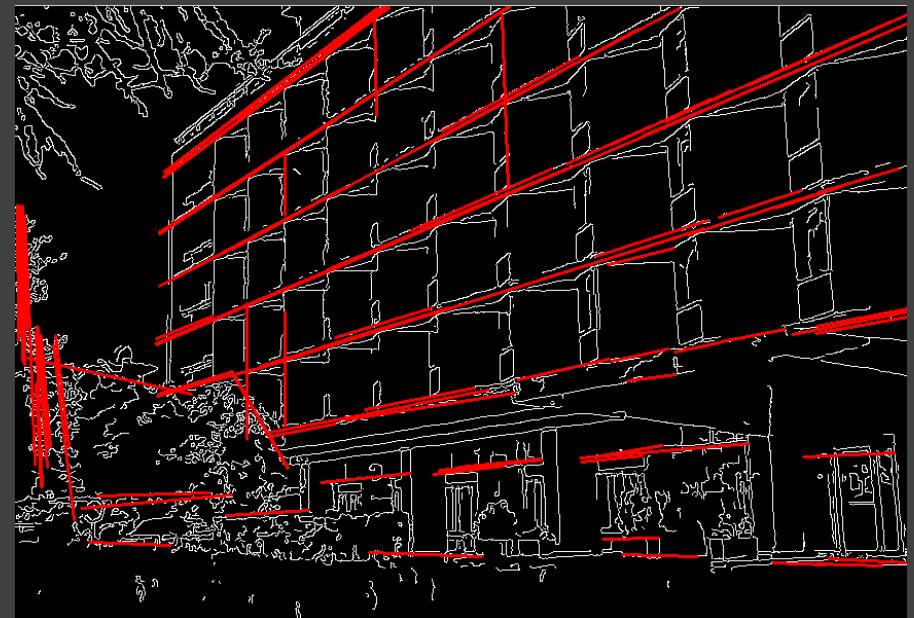


Análisis Local

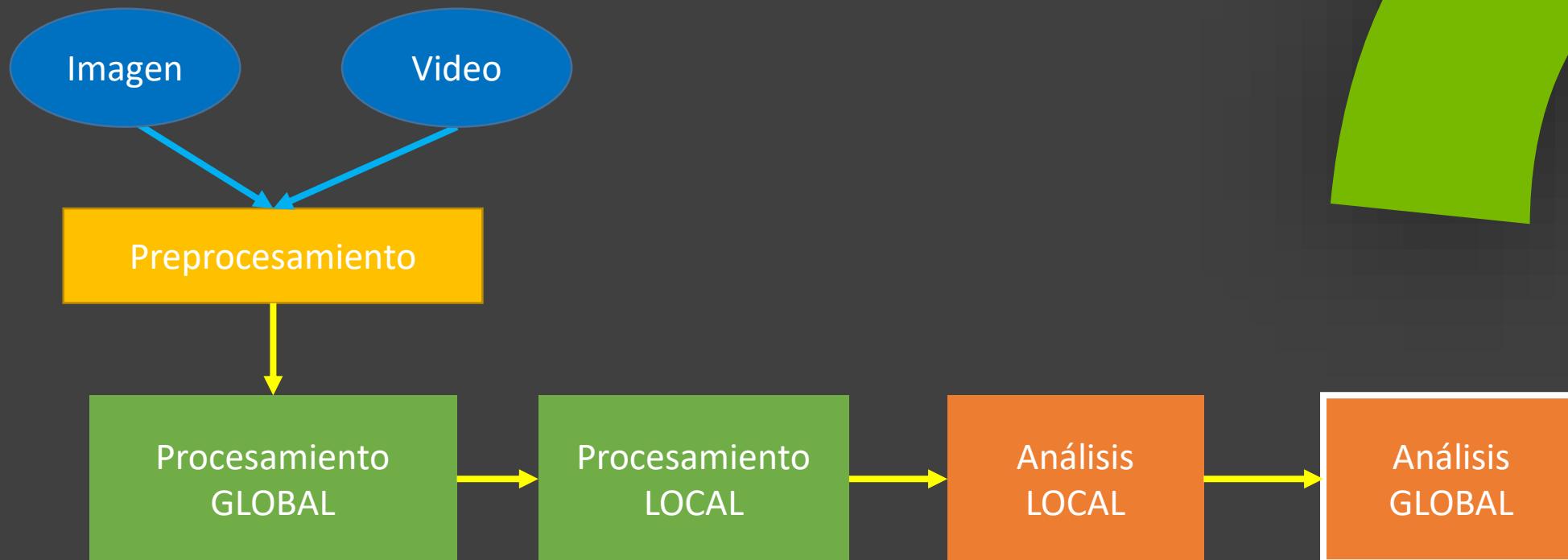
El operador básico en esta etapa es la **transformada de Hough** [23]

Este va a traducir bordes, en distintos **elementos geométricos (rectas y círculos)**

El fundamento matemático que hay detrás, es que coge pares de puntos que estima que son una recta (coordenadas cartesianas) , y los traduce a **coordenadas polares. (Números complejos)**.



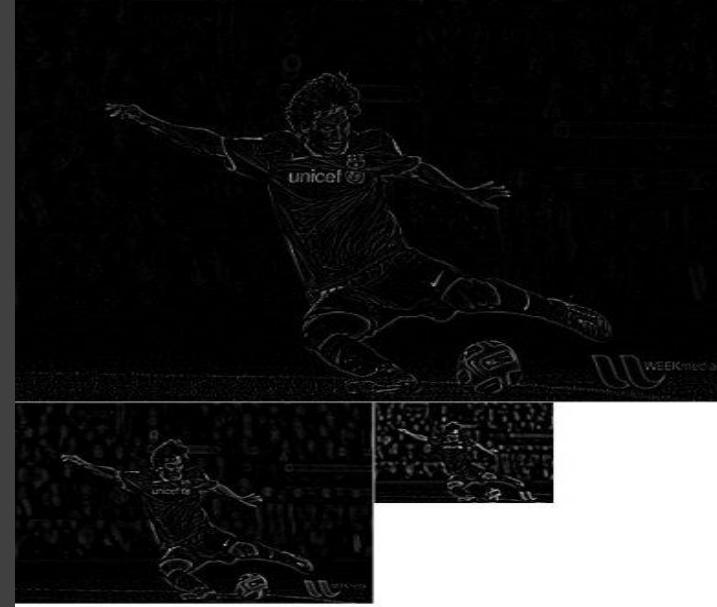
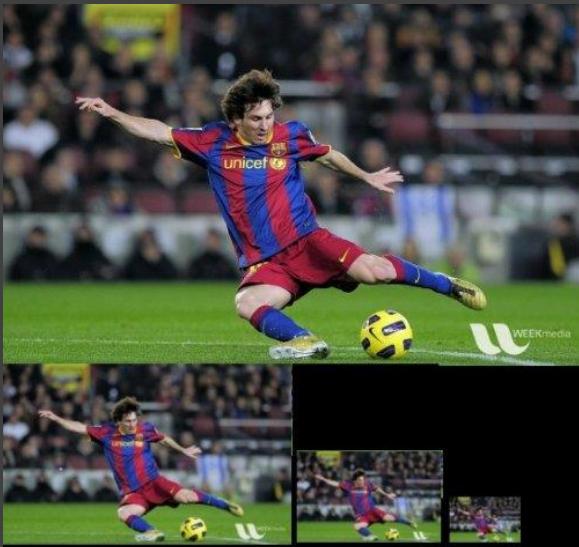
Análisis Global



Análisis Global

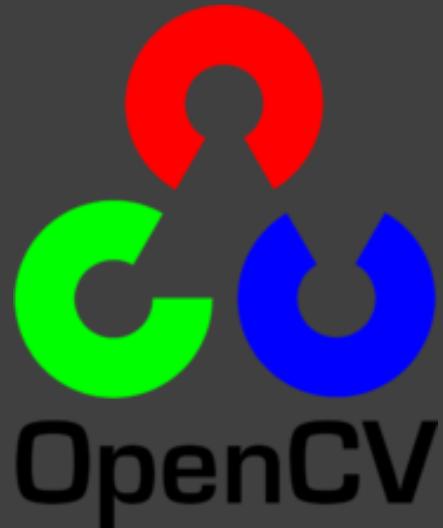
El análisis global, pretende mejorar los resultados anteriores, usando técnicas del procesamiento global sobre **distintas copias de la imagen original**, aplicando un solape de los distintos resultados.

La técnica más común es el uso de **pirámides de imágenes** [24], esto es, generar un conjunto n de imágenes más pequeñas en resolución que la imagen original y repetir las etapas del pipeline por cada una de ellas:



Ejemplo práctico usando OpenCV

OpenCV, es una librería de Software libre, desarrollada por Intel, y se utiliza para tareas de visión computacional. Posee un amplio abanico de herramientas de IA, así como diferentes wrappers en distintos lenguajes de programación . Tiene un carácter multiplataforma.



www.opencv.org

Ejemplo práctico usando OpenCV

En este ejemplo práctico nos vamos a descargar los siguientes ficheros:

En primer lugar Python:

- <https://www.python.org/downloads/release/python-371/>

Después de instalarse, abrimos en línea de comandos y escribimos:

- `pip install numpy opencv-python`

Por último descargamos el siguiente repositorio de git:

- <https://github.com/andresescobarcotan/Taller-OpenCV>

Ejemplo práctico usando OpenCV

Una vez descargados y instalados los ficheros, vamos a resolver el siguiente ejercicio ([ejercicioSinResolver.py](#)):

- El ejercicio, consiste en **eliminar el fondo de un video**, de tal forma que **queden** los objetos que se mueven sobre un fondo negro.
- Por tanto en el video elegido ('[cochesPasando.mp4](#)') , que muestra un conjunto de coches transitando en una carretera, se trata de eliminar la carretera y que los coches circulen sobre el vacío.
- Existen varias aproximaciones para resolver este ejercicio, alguna de ellas son:
 1. Usar alguno de los métodos de extracción de fondos disponibles de OpenCV ([ejercicioResueltoBMOG.py](#))
 2. Crear una máscara, a partir de una muestra del fondo y usando la operación de la resta ([ejercicioResuelto.py](#))
 3. Igual que el anterior, pero calcularemos los bordes de los vehículos y usaremos el área rellenada de los contornos para calcular la máscara. ([ejercicioResueltoContornos.py](#))

Referencias

- [1]: <http://www.lenna.org/>
- [2]: <http://fortune.com/2017/09/06/satellite-hurricane-irma-jose-and-katia-atlantic/>
- [3]: <https://gallery.urthecast.com/rectangular-crop-fields>
- [4]: <https://www.harrisgeospatial.com/Support/Maintenance-Detail/ArtMID/13350/ArticleID/16201/Analysts-Learn-New-Image-Analysis-Techniques-at-the-ENVI-Rapid-Learning-Series>
- [5]: <https://www.elnortedecastilla.es/20110318/local/valladolid/deimos-captó-imágenes-tsunami-201103181947.html>
- [6]: <https://www.sandia.gov/media/NewsRel/NR1999/Lynx.htm>
- [7]: <https://www.mobivap.es/2012/react/>
- [8]: <https://www.lauritz.com/en/auction/reyaz-nadi-color-transfer-on-wood-mona-lisa-1993-limit/i4625260/>
- [9]: <http://news.mit.edu/2018/faster-analysis-of-medical-images-0618>
- [10]: <https://nuit-blanche.blogspot.com/2013/07/fourier-domain-beamforming-path-to.html>

Referencias

- [11]: <https://www.shutterstock.com/es/video/clip-33521560-autonomous-driverless-car-computer-vision-object-detection>
- [12]: <https://www.pyimagesearch.com/2015/03/30/accessing-the-raspberry-pi-camera-with-opencv-and-python/>
- [13]: <https://www.vix.com/es/btg/curiosidades/2011/10/02/el-espectro-visible-de-luz>
- [14]: <https://es.wikipedia.org/wiki/RGB>
- [15]: https://es.wikipedia.org/wiki/Modelo_de_color_HSV
- [16]: https://es.wikipedia.org/wiki/Tri%C3%A1ngulo_de_Pascal
- [17]: <https://es.wikipedia.org/wiki/Convoluci%C3%B3n>
- [18]: <http://www.iro.umontreal.ca/~mignotte/TestImages.html>
- [19]: https://docs.opencv.org/3.4/dd/d4d/tutorial_js_image_arithmetics.html
- [20]: https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_imgproc/py_filtering/py_filtering.html
- [21]: [https://en.wikipedia.org/wiki/Thresholding_\(image_processing\)](https://en.wikipedia.org/wiki/Thresholding_(image_processing))
- [22]: https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_imgproc/py_gradients/py_gradients.html
- [23]: https://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/hough_lines/hough_lines.html
- [24]: https://docs.opencv.org/3.4/dc/dff/tutorial_py_pyramids.html