

# **Introducción a la visión por ordenador**



**SOCIEDAD  
ECUATORIANA  
DE ESTADISTICA**

**Katherine Morales**



- PhD candidate in TSP - Institut Polytechnique de París, Francia.
- Máster Matemática Aplicada - Ciencia de Datos.
- Ingeniera Matemática - Estadística e Investigación Operativa
- Analista de datos
- Consultora
- Capacitadora SEE, Programa Python Full stack, EPN



**GitHub**











**LinkedIn**

# Python

Accesible, fácil y se puede usar en varios entornos.

Grandes empresas como Google, Instagram, Pinterest, Facebook, Netflix o Dropbox siguen utilizando Python en su desarrollo tecnológico.

Big Data, AI, Data Science, frameworks de pruebas y desarrollo web

Language Ranking: IEEE Spectrum					
Rank	Language	Type			Score
1	Python▼				100.0
2	Java▼				95.3
3	C▼				94.6
4	C++▼				87.0
5	JavaScript▼				79.5

- **Interpretado:** significa que Python “interpreta” el código del programador, es decir, lo traduce y lo ejecuta a la vez.
- **Multiparadigma:** porque es un lenguaje de programación que admite el uso de varios paradigmas de programación (modelos de desarrollo), por lo que no exige a los programadores un estilo único para programar. ¿Cuáles son los paradigmas de programación que permite Python? Programación orientada a objetos, programación imperativa y programación funcional.
- **Multiplataforma:** el lenguaje Python puede ejecutarse en diferentes sistemas operativos como Unix, Linux, macOS y Windows.



# Introducción

- ¿Qué es la visión por ordenador?
- Problemas, métodos y algoritmos que ayudan a la comprensión automática (informática) de las características de alto nivel de las imágenes y los vídeos.
  - Contornos, esquinas y otras características
  - Análisis del movimiento
  - Identificación de objetos

Objetivo: el ordenador puede comprender y analizar su entorno



# Segmentación

- La segmentación es uno de los pasos de procesamiento o preprocesamiento más comunes en la visión por ordenador
- Importante para la comprensión semántica de una escena
  - Robótica: ¿dónde estoy, qué hay delante de mí, qué obstáculos hay?
  - Imágenes médicas: diferentes partes del cuerpo

Hay muchos enfoques para la segmentación:

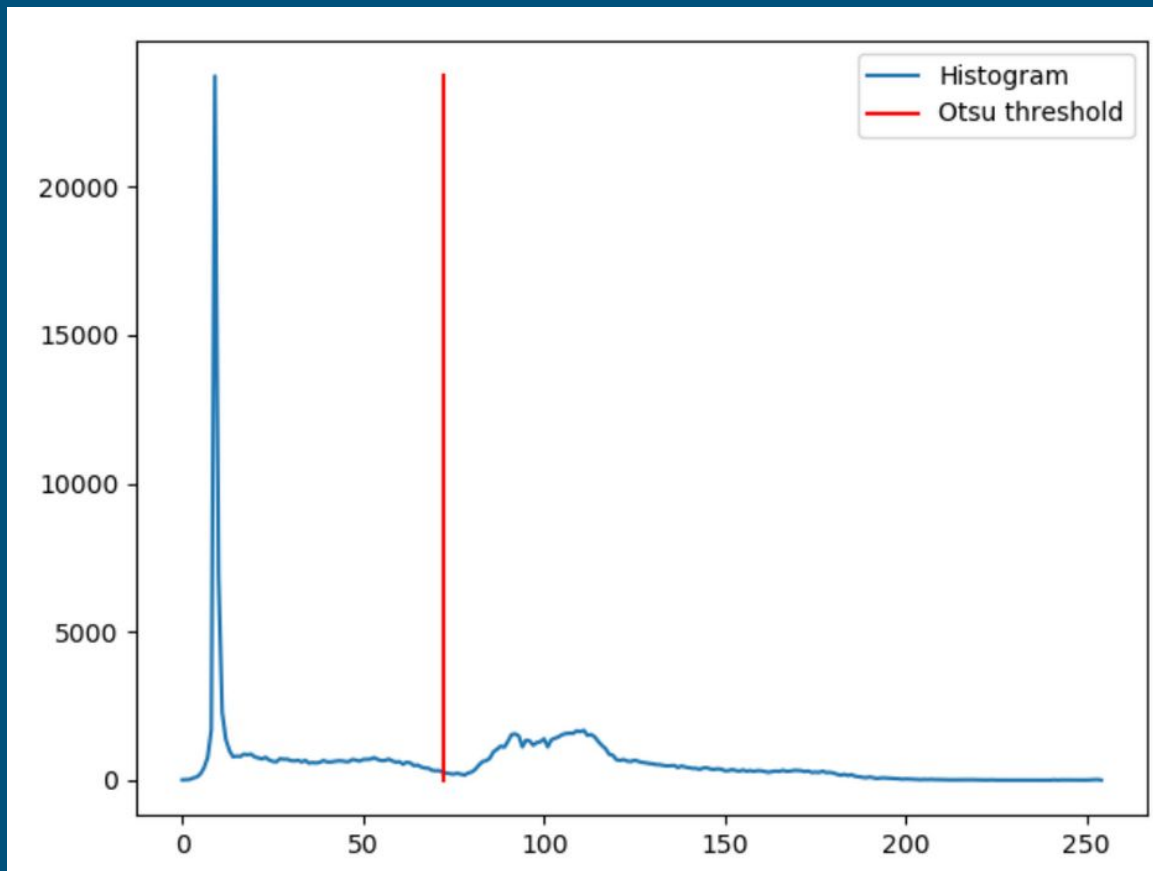
- Thesholding
- Region growing
- Active contours (differential geometry)
- Motion-based

# Thesholding: Otsu

Ejemplo de Otsu segmentation: Input



# Ejemplo de Otsu segmentation: histograma





# Ejemplo de Otsu segmentation: segmentación



Algunas desventajas del método de Otsu :

- Umbral global, los objetos segmentados pueden no estar conectados
- Basado enteramente en los valores de los píxeles, no existe la noción de coherencia geométrica

Otra idea intuitiva es utilizar métodos **region growing**

# Región en crecimiento (region growing)

- Comenzamos eligiendo inicialmente algunos seed points
  - Puede ser una entrada del usuario o aleatoria
- A continuación, el algoritmo intenta incluir más y más píxeles en una vecindad  $\Psi$
- Los píxeles se incluyen en función de una decisión relativa a la intensidad del píxel
- Criterio más sencillo: diferencia absoluta entre el nuevo valor del píxel  $I(x, y)$  y la media de la región,  $\mu$

# Seed inicial



# Segmentación



# Seed inicial



# Segmentación



Ventaja del crecimiento de la región growing: noción de geometría/topología, la región debe estar conectada

Inconvenientes del cultivo de la región :

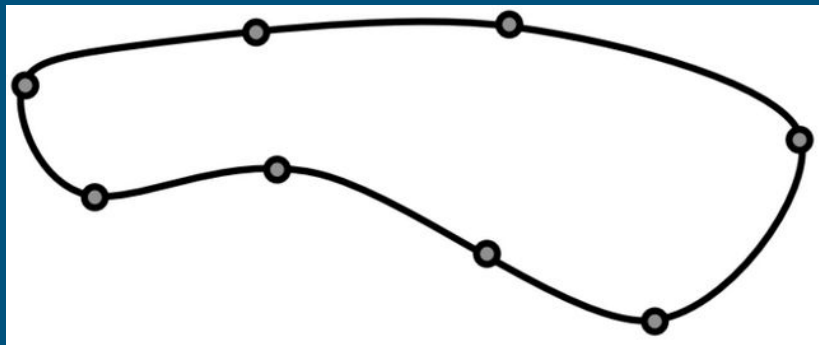
- No es robusto a los valores atípicos (ruido)
- No hay noción de suavidad en el límite del segmento

Para ello, recurrimos a enfoques basados en la geometría. El más conocido es contornos activos -active contours- (también conocido como el modelo de las serpientes)



# Active contours

- Los contornos activos se utilizan para segmentar objetos específicos
  - Comience por elegir una segmentación inicial
  - Refinar iterativamente la segmentación minimizando una energía
- El segmento está determinado por una curva cerrada  $\gamma$ , parametrizada por un conjunto de puntos  $a_i, i = 0, \dots, N-1$
- Todo lo que está dentro de la línea es un segmento, todo lo que está fuera es el otro segmento



# Active contours

$$E(v) = \int_0^1 E_{int}(v(s)) + E_{image}(v(s)) ds$$

- $E_{int}$  es la energía interna, lo que requiere que la curva sea lo más suave posible
- $E_{image}$  es la energía de la imagen, que requiere que la curva se contraiga alrededor de los bordes
- Este enfoque utiliza herramientas de la geometría diferencial y del análisis funcional
- Un tema bastante técnico, por lo que no entramos en grandes detalles
- La cuestión principal: ¿cómo minimizar la energía de la ecuación? Esto requiere el cálculo de variaciones
- También es necesario en el flujo óptico (estimación del movimiento)

# Active contours

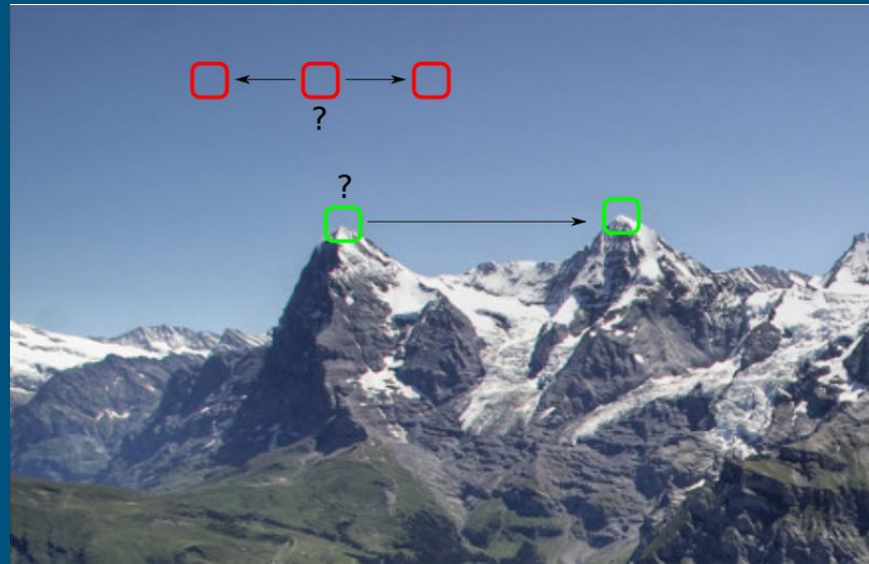


# Active contours - resumen

- Hay bastantes variantes de la idea de los contornos activos (Chan y Vese),
  - Ventajas: formulación rigurosa, minimización del funcional, geométricamente menesteroso
  - Desventajas: bastante técnico para entender/implementar, los parámetros del algoritmo pueden ser difíciles de elegir
- En general, los algoritmos de contorno activo se utilizan cuando es necesario detectar un objeto específico (por ejemplo, imágenes médicas)
- Más recientemente, el aprendizaje profundo se ha convertido en el estado del arte (como en muchas áreas)

# Feature detection

- A menudo, deseamos detectar características en las imágenes. ¿Qué son las características?
  - Líneas, esquinas y otros puntos de interés
- Para tareas como el seguimiento, necesitamos puntos inequívocos e identificables
- Los puntos sin textura son muy difíciles de seguir en una imagen



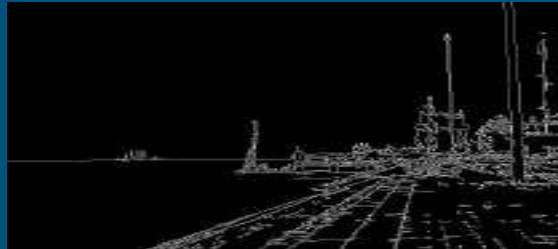
- A menudo es necesario detectar líneas y segmentos en las imágenes
  - Detección de puntos de fuga, estructura, detección de objetos
- ¿Cómo encontramos las líneas? Dos pasos :
  - Detección de bordes (filtros sobel o de otro tipo)
  - Una forma de agrupar colecciones fuertes de aristas: la transformada de Hough

# Transformación de Hough: Detección de líneas

- 1 Utilizar un detector de bordes por píxel para detectar posibles líneas
- 2 Calcular la transformada de Hough de la imagen de borde
- 3 Transformación de Hough con umbral para encontrar líneas



Imagen de entrada



Detección de bordes



Líneas detectadas

- Nuestra última tarea es agrupar los puntos de la línea en segmentos
- Esto se suele hacer mediante un algoritmo heurístico, especificando :
  - Una longitud de segmento mínima
  - Una longitud máxima de agujeros en un segmento



Imagen de entrada



Detección de bordes



Líneas detectadas

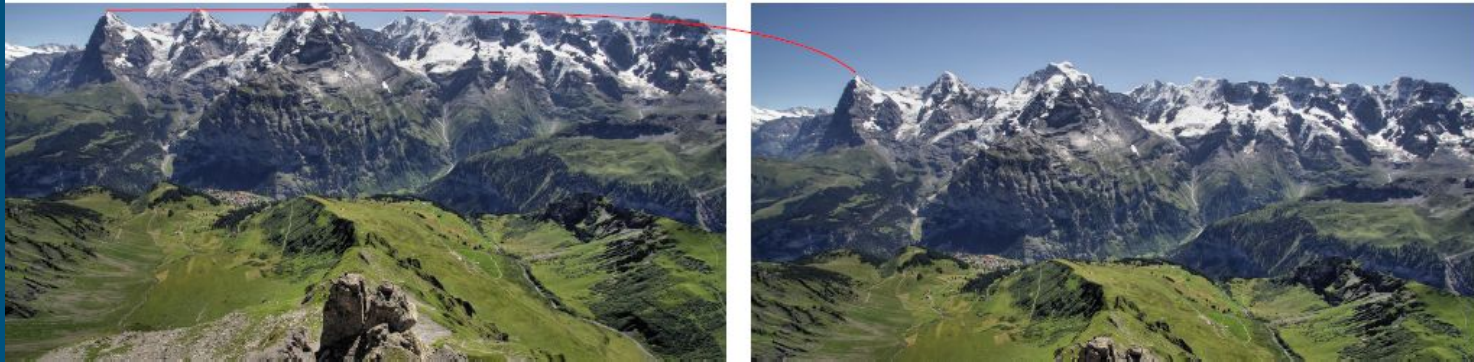


Segmentos detectados



# Motion estimation

- Una tarea común en la visión por ordenador es la estimación del movimiento
- Útil para: el seguimiento, la segmentación de objetos
- La principal hipótesis utilizada para la estimación del movimiento es la suposición de constancia del brillo



- Esto puede ser falso, por ejemplo en presencia de cambios de luz/sombras

- Block matching: es el enfoque más sencillo en términos prácticos.  
Problema: de apertura.
- Optical flow: es un método para estimar el movimiento con un enfoque continuo

Ejemplo del problema de apertura.



Optical flow



Block matching:

Ventajas: fácil de entender y aplicar

Desventajas: tiempo de ejecución largo, sin regularización  
(problema de apertura)

Optical flow:

Ventajas: regularización, campo de flujo suave, más robusto que el emparejamiento de bloques (block matching)

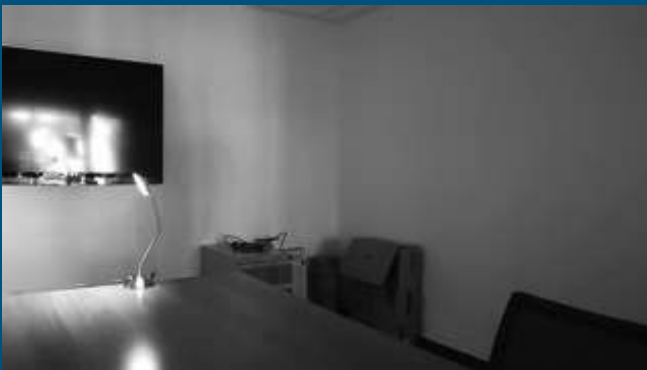
Desventajas: la teoría es más difícil de entender, los esquemas numéricos son difíciles de aplicar

# Background estimation

En los datos de vídeo, es necesario separar el primer plano en movimiento y el fondo estático

- Esto es necesario para detectar y analizar personas/objetos
- Ejemplo: un robot que navega alrededor de obstáculos, aislando los objetos en movimiento para el entrenamiento

También se conoce como detección/substracción de fondo, detección de primer plano, etc. (en la mayoría de los casos se trata del mismo problema)



- Simple temporal filters :  
mean, median
- Gaussian mixture models
- Robust principle component analysis

# Object detection

Un problema muy común en la visión por ordenador es la detección de objetos

Por detección de objetos se entiende, de hecho, varias cosas:

- Detección de la presencia

- Localización de objetos (¿dónde está el objeto?)

- Reconocimiento de objetos (por ejemplo, ¿de quién es esta cara?)

La detección de objetos, es decir, el análisis de objetos en una escena, es uno de los problemas más difíciles en la visión por ordenador. ¿Por qué? Las variaciones en :

- Pose, punto de vista, zoom, rotación

- Condiciones de iluminación, color

- Ruido, oclusión

Este es un campo muy amplio, y puede depender del tipo de objeto

# Face detection:

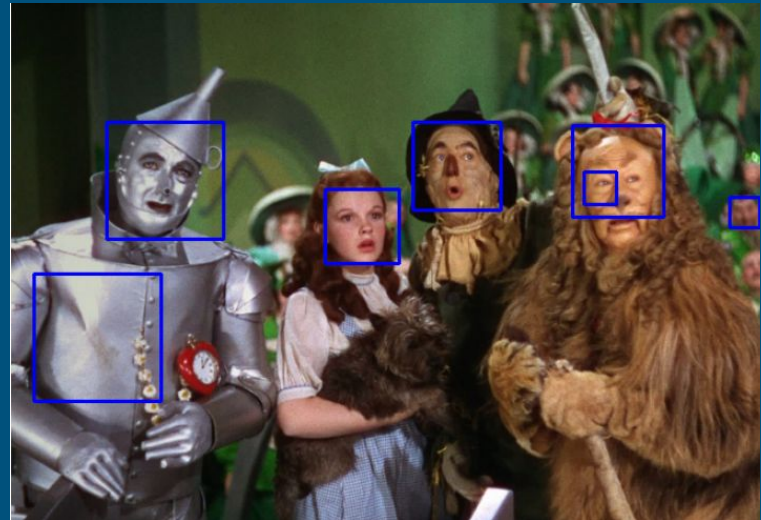
Al ser una de las tareas más complejas, no existe un marco unificador

Existen varios tipos de enfoques generales:

- Basado en características
- Características de bajo nivel (borde, textura de la piel) Modelos de forma
- Mezcla de características, Viola-Jones

Basado en imágenes

- Enfoque estadístico: PCA, SVM Eigenfaces
- Redes neuronales



*Conclusión*

Hemos examinado los siguientes problemas fundamentales de la visión por ordenador

- Segmentación

- Detección de características

- Estimación del movimiento

- Estimación del fondo

- Detección/reconocimiento de objetos

A menudo se emplean varias herramientas/técnicas matemáticas destacadas:

- Cálculo de variaciones (enfoques variacionales)

- Álgebra lineal (dirección principal de la variación)

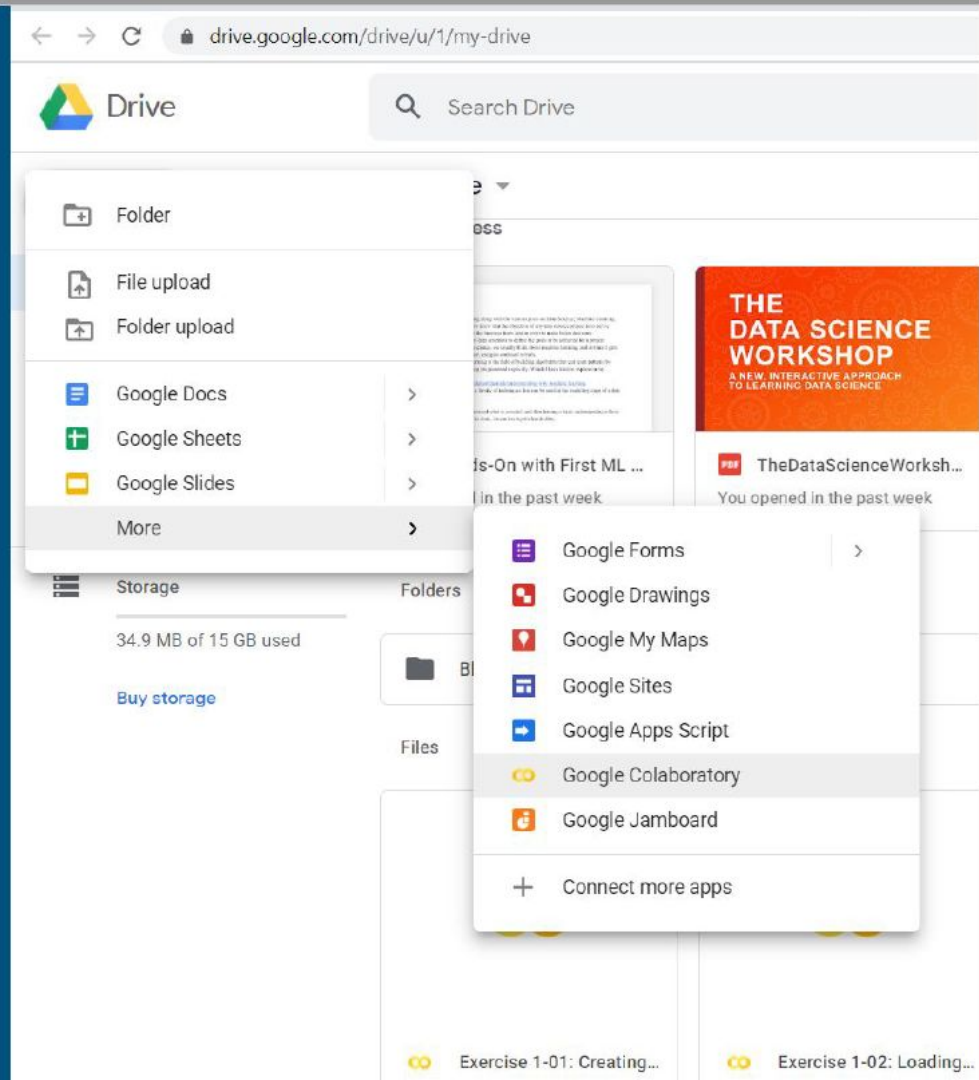
- Análisis de componentes principales (PCA)

- Detección de rasgos + clasificadores estándar



# GOOGLE COLAB

**Colab** es un servicio cloud, basado en los Notebooks de Jupyter, que permite el uso gratuito de las GPUs y TPUs de **Google**, con librerías como: Scikit-learn, PyTorch, TensorFlow, Keras y OpenCV.



*Muchas  
gracias*