

# Procesamiento de imágenes con Python

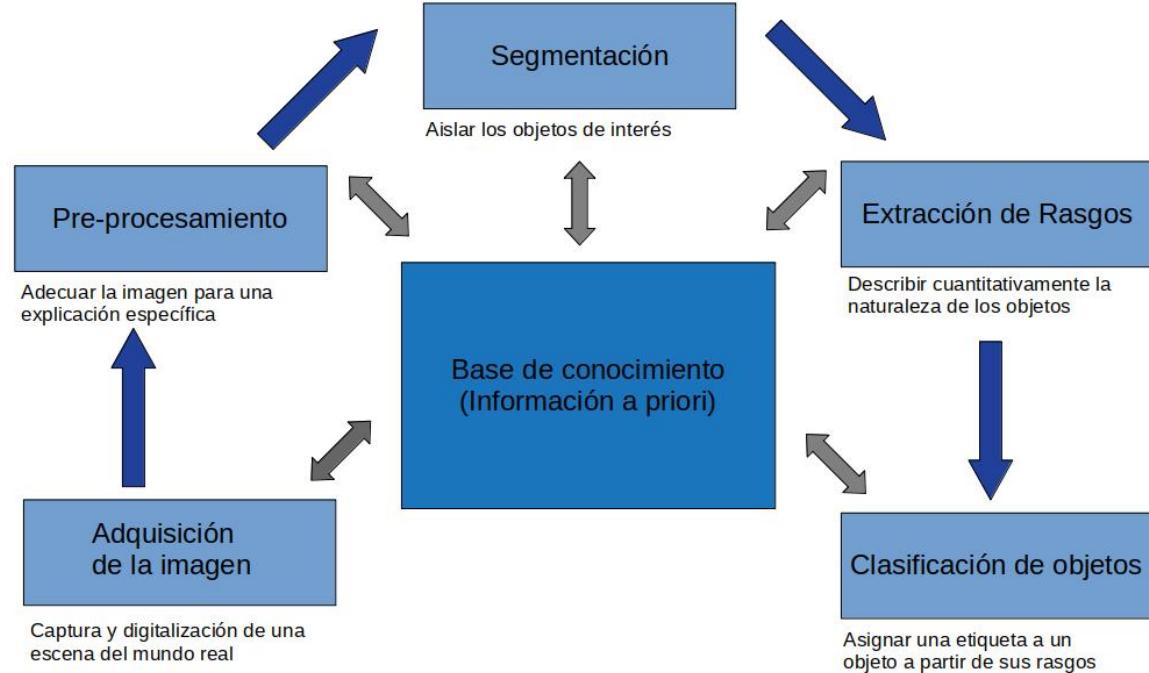
<Contacto>

Mail: [cabustillo13@hotmail.com](mailto:cabustillo13@hotmail.com) Github: cabustillo13

</Contacto>

## Agenda

- Visión por computadora
- Modelo / Objetivo
- Descripción de etapas
- Desarrollo en Python
- Otros ejemplos



¿Qué es esto?



¿Perro? ¿Ratón?



¿Muffin? ¿Quequito?

<Contacto>

Mail: [cabustillo13@hotmail.com](mailto:cabustillo13@hotmail.com) Github: [cabustillo13](https://github.com/cabustillo13)

</Contacto>

## ¿Qué es esto?



¿Perro? ¿Ratón? ¿Muffin? ¿Quequito?

<Contacto>

Mail: [cabustillo13@hotmail.com](mailto:cabustillo13@hotmail.com) Github: [cabustillo13](#)

</Contacto>

## Nos adentramos al problema

Yo vivo en la capital del vino Mendoza, Argentina. La provincia, representa más del 60% de la producción de vino del país.

Hay una gran variedad de varietales, y para alguien que no es de la región es difícil reconocer cada uno a simple vista.

Se pueden clasificar a partir de distintas características: color, pH, acidez, momentos de Hu, etc.



Fuente imagen: <https://revistasumer.com/comenzó-la-cosecha-2020/>

## Nos adentramos al problema

Propongo un modelo de juguete para analizar uvas blancas, rosadas y azules. En el cual le presentamos una imagen al algoritmo y éste nos diga a qué clase de uva pertenece.



[Input]



Es una Grape Blue / Uva azul

[Output]

## Modelo / Objetivo

**Tipo de agente:** Es un agente que aprende, racional, no omnisciente y autónomo.

**Propiedades del entorno:** Totalmente observable, determinista, episódico, estático, discreto e individual.

**Rendimiento:** Cantidad de predicciones correctas de las imágenes que se están evaluando.

**Entorno:** Es el contexto plasmado en la imagen de entrada.

**Actuadores:** Es el medio de notificación del usuario donde se muestra la predicción del algoritmo.

**Sensores:** Cámara.

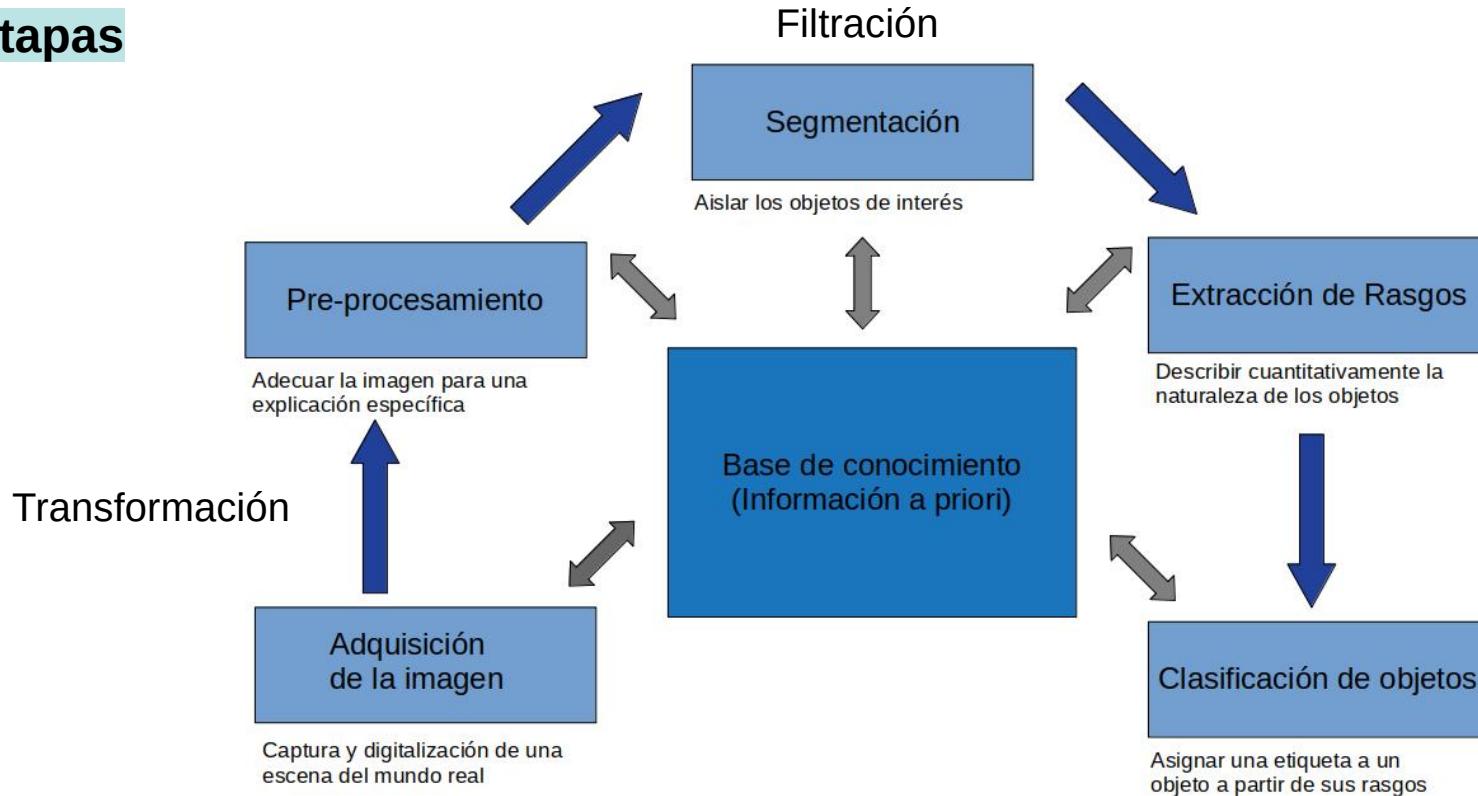
# Procesamiento de imágenes con Python

<Contacto>

Mail: [cabustillo13@hotmail.com](mailto:cabustillo13@hotmail.com) Github: cabustillo13

</Contacto>

## Etapas



<Contacto>

Mail: [cabustillo13@hotmail.com](mailto:cabustillo13@hotmail.com) Github: [cabustillo13](#)

</Contacto>

## Paso 1/6 : Adquisición de imágenes

Vamos a utilizar un dataset de Kaggle: Fruits 360.

Y solo utilizaremos las clases grapeBlue, grapeWhite y grapePink.



Link: <https://www.kaggle.com/moltean/fruits>

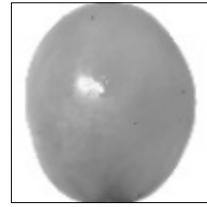
## Paso 2/6 : Transformación

Inicialmente cada imagen es de 100x100 píxeles.



## Paso 3/6 : Preprocesamiento

Conversión a escala de grises cada imagen.



## Paso 4/6 : Filtración

Aplicar un filtro para eliminar ruido.



<Contacto>

Mail: [cabustillo13@hotmail.com](mailto:cabustillo13@hotmail.com) Github: [cabustillo13](#)

</Contacto>

## Paso 5/6 : Extracción de rasgos

A veces previo a esta etapa, se hace una segmentación.

Definir los momentos de Hu 1,2 y 4.

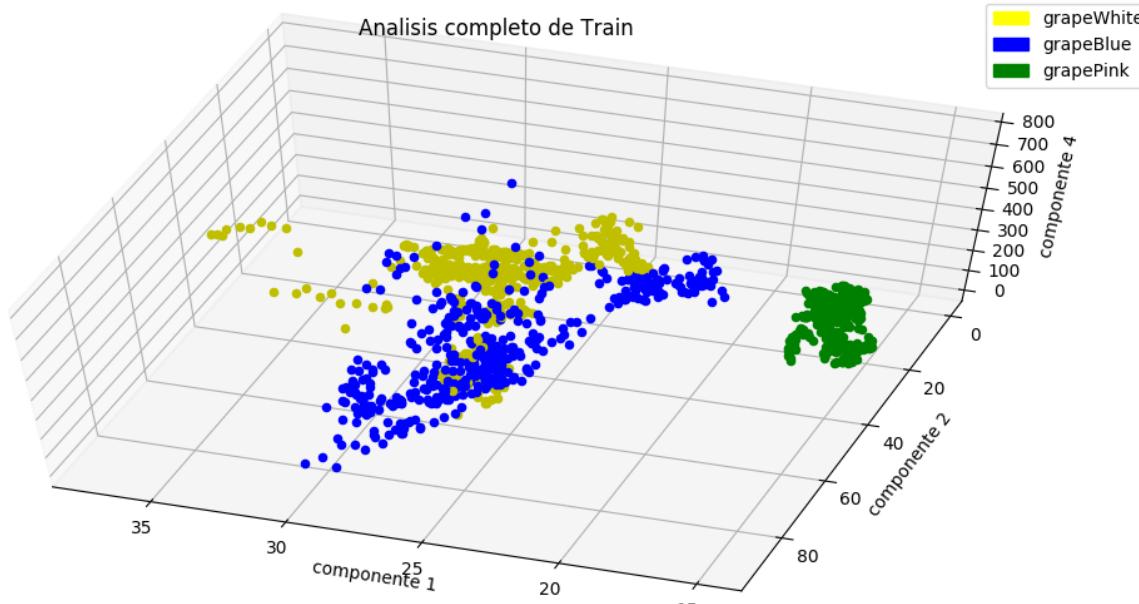
Blue Grape : [7.266208318798105, 0.024048648552466356, 0.6853083786356352]

Pink Grape : [4.7386009969045215, 0.13936792354833638, 0.6428959835480056]

White Grape: [8.903935254086356, 0.005172179956379898, 0.0464418568479963]

## Gráfico Momentos de Hu

Para 1470 imágenes  
¿eso es bueno?



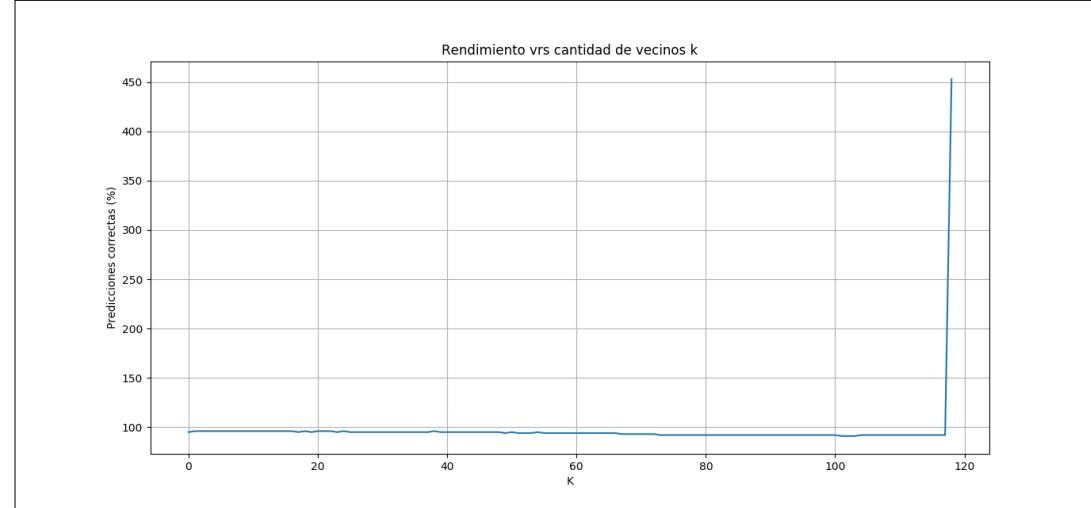
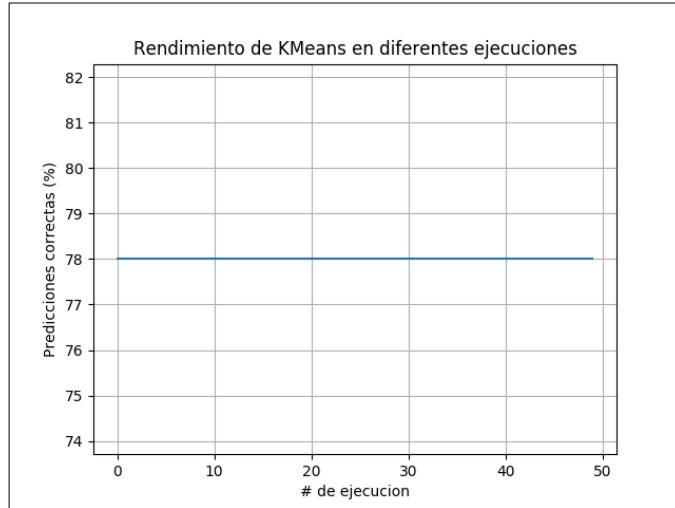
<Contacto>

Mail: [cabustillo13@hotmail.com](mailto:cabustillo13@hotmail.com) Github: [cabustillo13](#)

</Contacto>

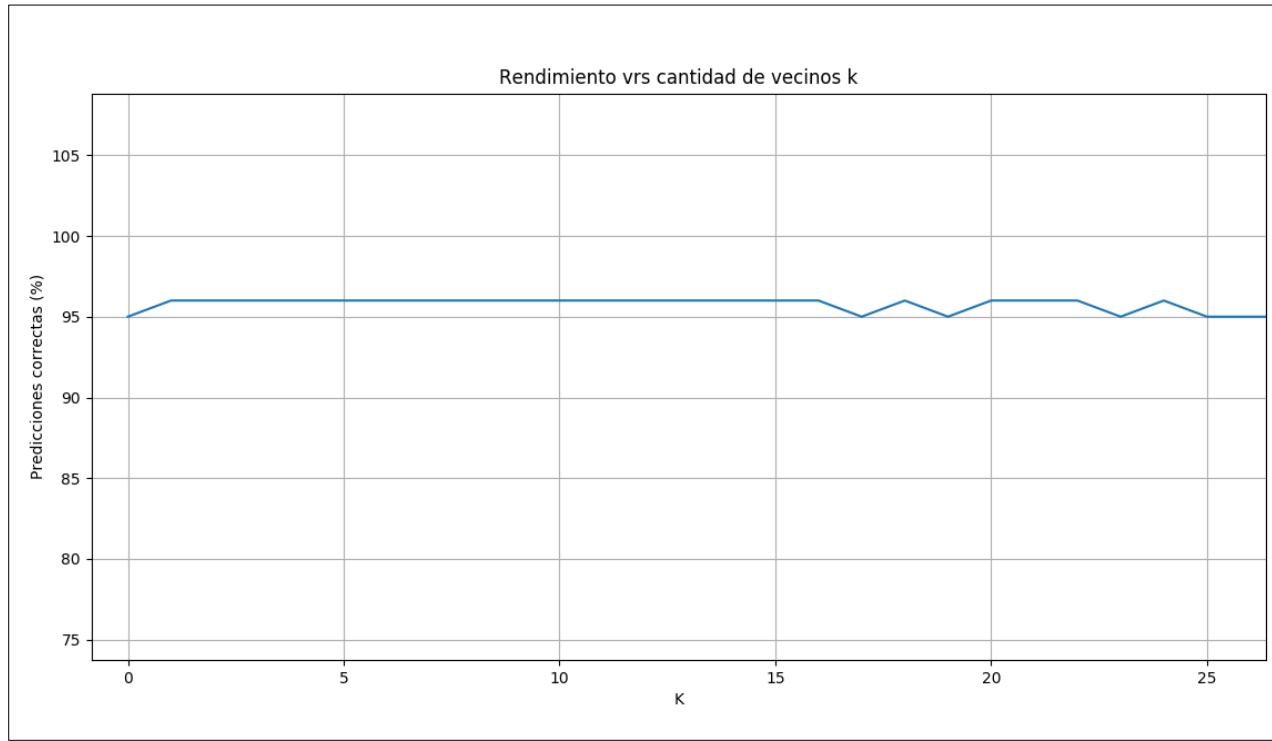
## Paso 6/6 : Clasificación

Utilizamos el algoritmo KNN (vecinos más cercanos) debido que se obtuvo un buen rendimiento.



## Paso 6/6 : Clasificación

Al hacerle zoom a la imagen.



<Contacto>

Mail: [cabustillo13@hotmail.com](mailto:cabustillo13@hotmail.com) Github: cabustillo13

</Contacto>

## Implementación

Ver en mi repositorio:

<https://github.com/cabustillo13/Charla-procesamiento-de-imagenes>

```
carlos@cabustillo13:~/Documentos/Proyectos/Charla-procesamiento-de-imagenes$ python main.py
grapeWhite OK
grapeBlues OK
grapePink OK
Analisis completo de la base de datos de Train
Cantidad de imagenes analizadas:
1470
Introduce numero de la foto: 0

Inicializacion KNN

Predicciones para KNN con K=2:
grapeBlue
grapeBlue
carlos@cabustillo13:~/Documentos/Proyectos/Charla-procesamiento-de-imagenes$
```



photo0.jpg

```
carlos@cabustillo13:~/Documentos/Proyectos/Charla-procesamiento-de-imagenes$ python main.py
grapeWhite OK
grapeBlues OK
grapePink OK
Analisis completo de la base de datos de Train
Cantidad de imagenes analizadas:
1470
Introduce numero de la foto: 13

Inicializacion KNN

Predicciones para KNN con K=2:
grapeWhite
grapeWhite
carlos@cabustillo13:~/Documentos/Proyectos/Charla-procesamiento-de-imagenes$
```



photo13.jpg

<Contacto>

Mail: [cabustillo13@hotmail.com](mailto:cabustillo13@hotmail.com) Github: cabustillo13

</Contacto>

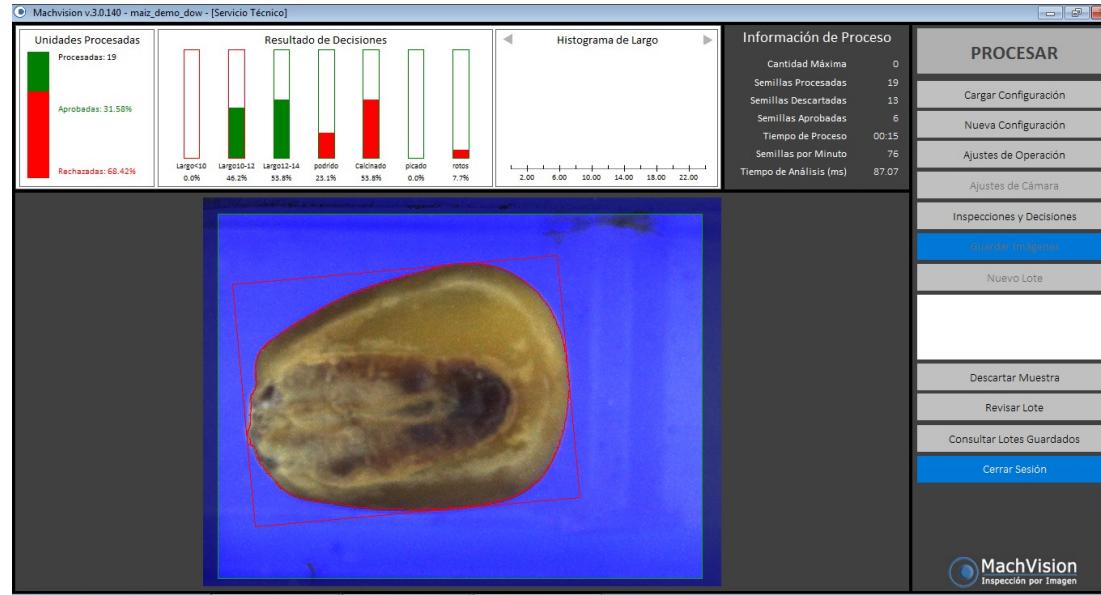
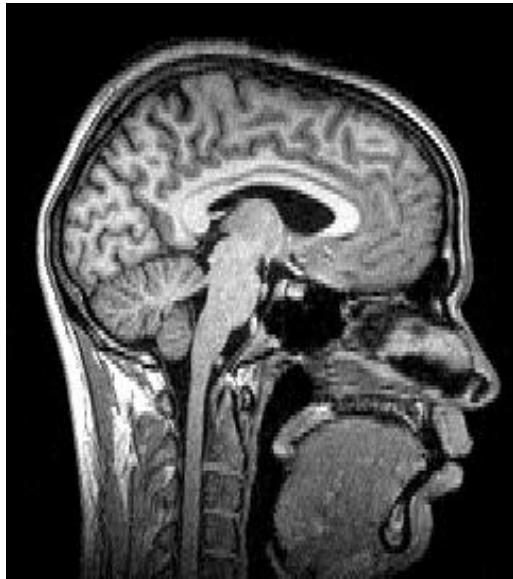
# Otros campos de aplicación

<Contacto>

Mail: [cabustillo13@hotmail.com](mailto:cabustillo13@hotmail.com) Github: cabustillo13

</Contacto>

## Otros ejemplos



<Contacto>

Mail: [cabustillo13@hotmail.com](mailto:cabustillo13@hotmail.com) Github: [cabustillo13](#)

</Contacto>

## Otros ejemplos



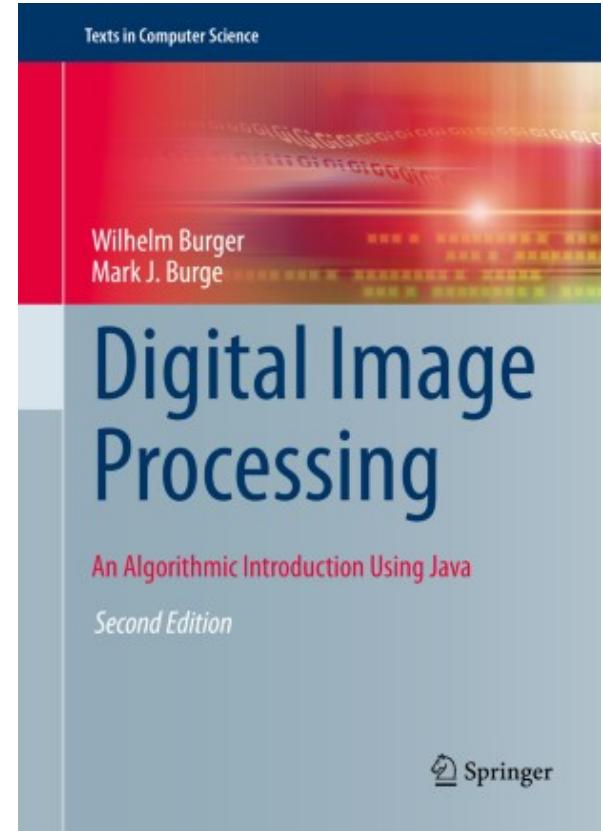
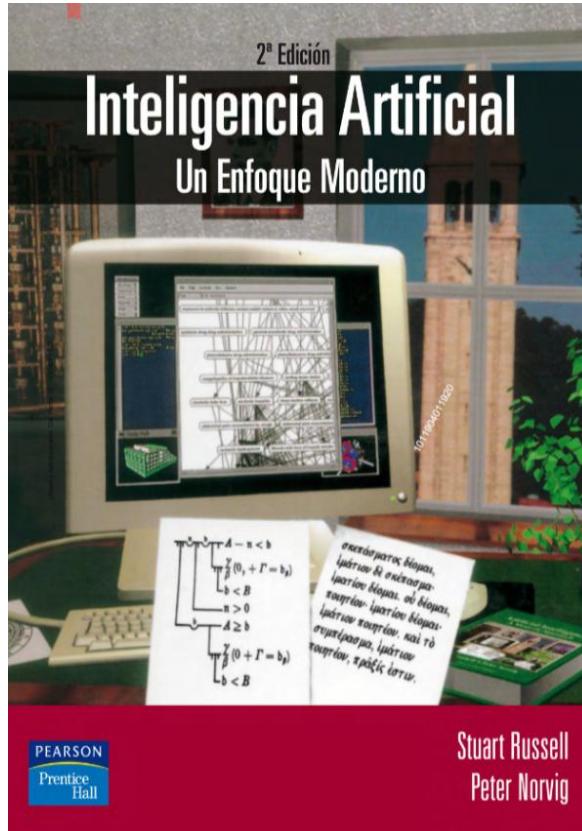
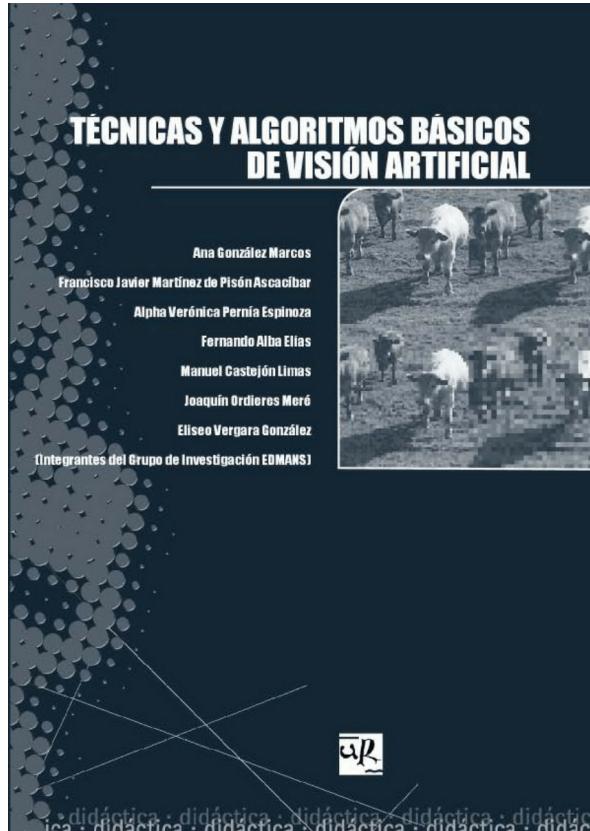
<Contacto>

Mail: [cabustillo13@hotmail.com](mailto:cabustillo13@hotmail.com) Github: [cabustillo13](#)

</Contacto>

## Bibliografía

Aparte de la Documentación de OpenCV



<Contacto>

Mail: [cabustillo13@hotmail.com](mailto:cabustillo13@hotmail.com) Github: [cabustillo13](https://github.com/cabustillo13)

</Contacto>

## Preguntas, dudas o sugerencias



**Repositorio:** <https://github.com/cabustillo13/Charla-procesamiento-de-imagenes>

**Telegram:** [@Cracken13](https://t.me/Cracken13)

<Contacto>

Mail: [cabustillo13@hotmail.com](mailto:cabustillo13@hotmail.com) Github: [cabustillo13](https://github.com/cabustillo13)

</Contacto>