



Procesamiento de Imágenes con Python

Carlos Bustillo

¡Hola!



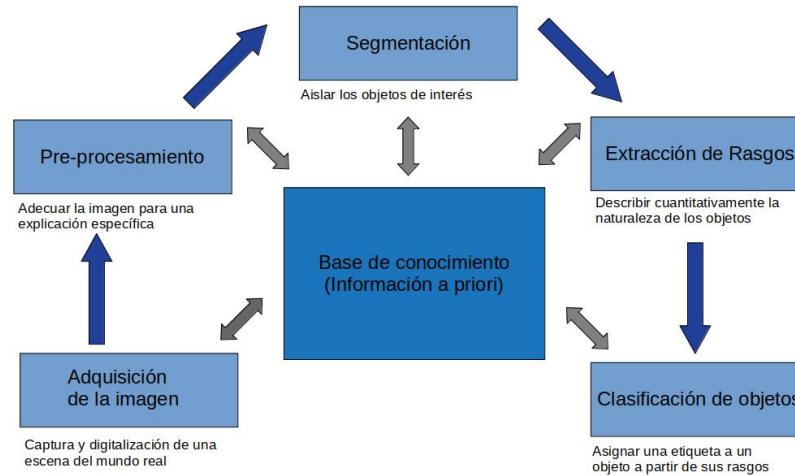
Soy Carlos Bustillo

- Estudiante avanzado de Ingeniería Mecatrónica- UNCUYO.
- Apasionado por el Procesamiento de Imágenes e Inteligencia Artificial.
- Actualmente participo de varias comunidades de Python.
- Open Source Contributor.



Agenda

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



¿Qué es esto?



¿Perro? ¿Ratón?



¿Muffin? ¿Quequito?



¿Qué es esto?



¿Perro? ¿Ratón? ¿Muffin? ¿Quequito?



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

Carlos Bustillo

1

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>

2

Transformación

Inicialmente cada imagen es de 100x100 píxeles.



Acá se puede recortar el elemento de la imagen, variación en la escala, etc.

Preprocesamiento

Conversión a escala de grises cada imagen.



$$E(x,y) = WR \cdot R(x,y) + WG \cdot G(x,y) + WB \cdot B(x,y)$$

WR=0.299 WG=0.589 WB=0.114

Método utilizado para señales a color en TV

WR=0.2125 WG=0.7154 WB=0.072

Método ITU-BT.709 para codificación digital a color

WR= $\frac{1}{3}$ WG= $\frac{1}{3}$ WB= $\frac{1}{3}$

Se obtiene una imagen más oscura

WR=0.309 WG=0.609 WB=0.082

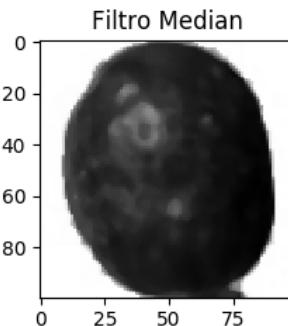
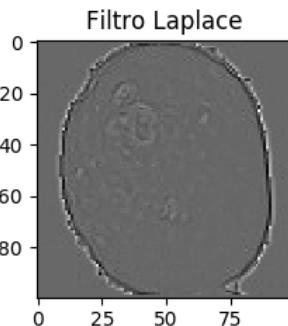
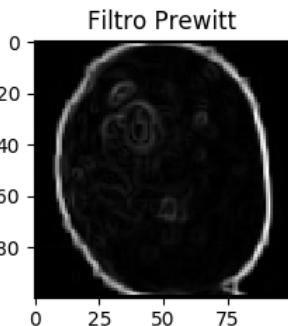
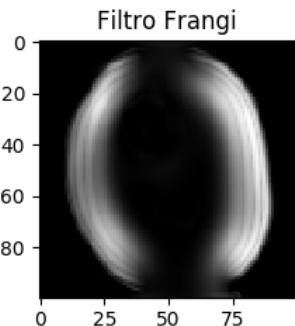
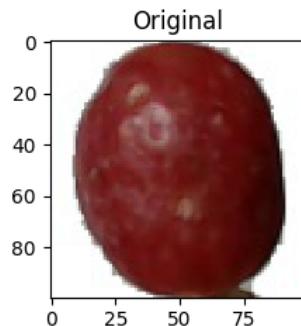
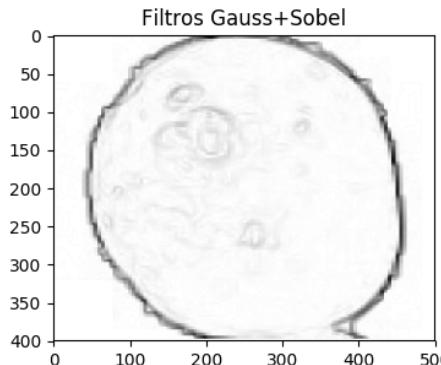
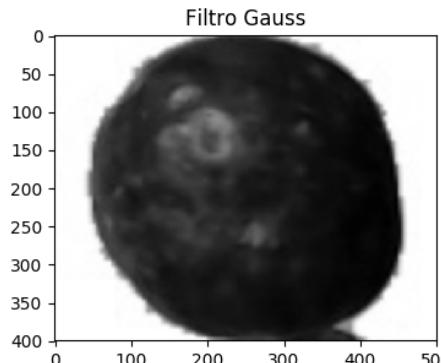
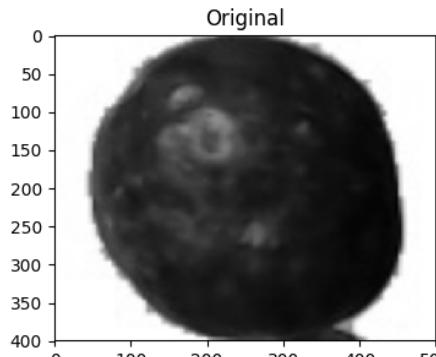
Corrección distorsión Gamma en señales de TV

Filtración

Aplicar un filtro para suavizar la imagen, eliminar ruido, realzar la imagen y detectar bordes.



Filtración



Extracción de rasgos

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

Definir los momentos de Hu 1,2 y 4.

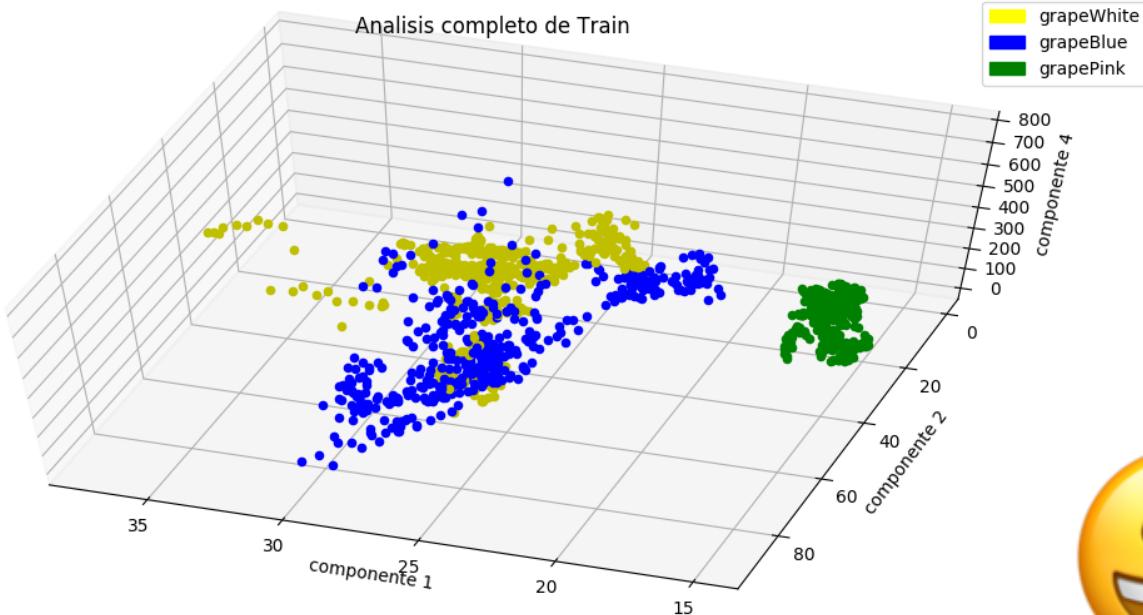
Blue Grape : [7.266208318798105, 0.024048648552466356, 0.6853083786356352]

Pink Grape : [4.7386009969045215, 0.13936792354833638, 0.6428959835480056]

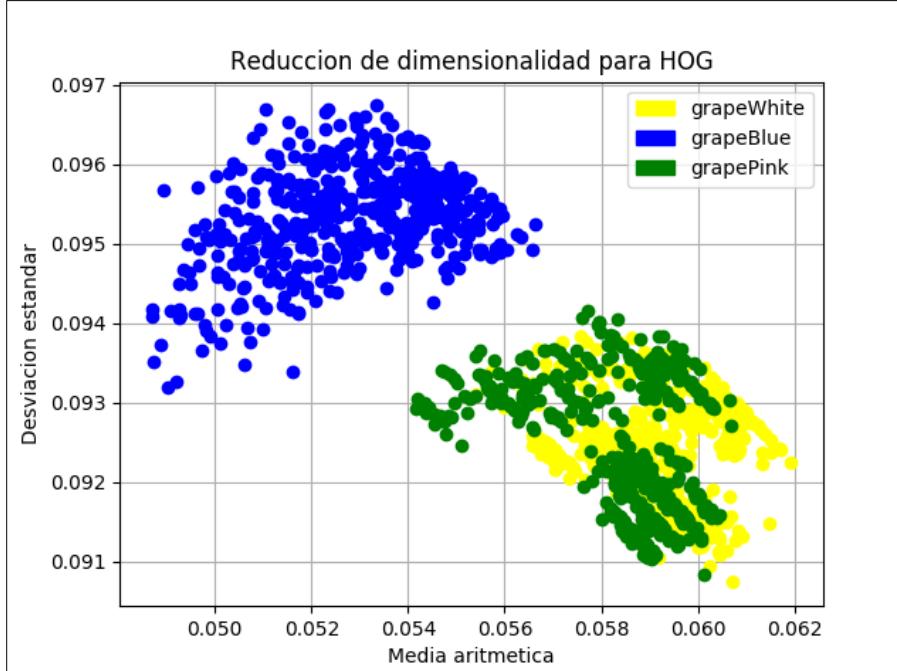
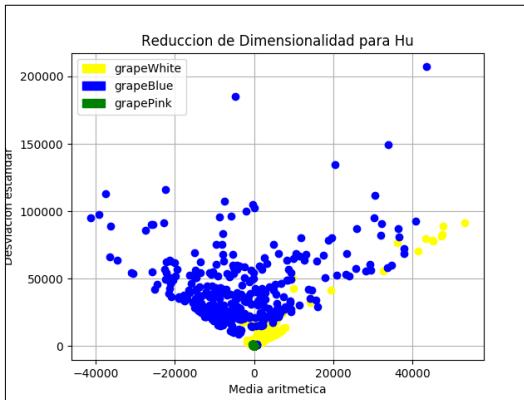
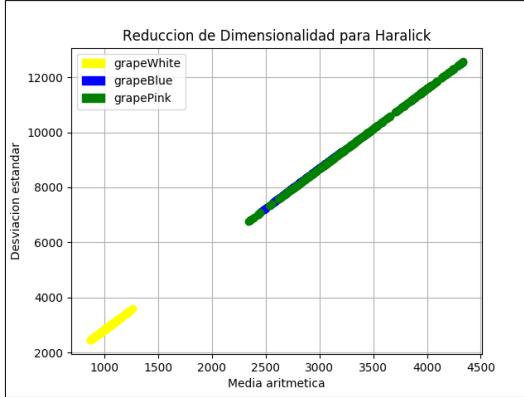
White Grape: [8.903935254086356, 0.005172179956379898, 0.046441568479963]

Gráfico de Momentos de Hu

Para 1470 imágenes
¿eso es bueno?



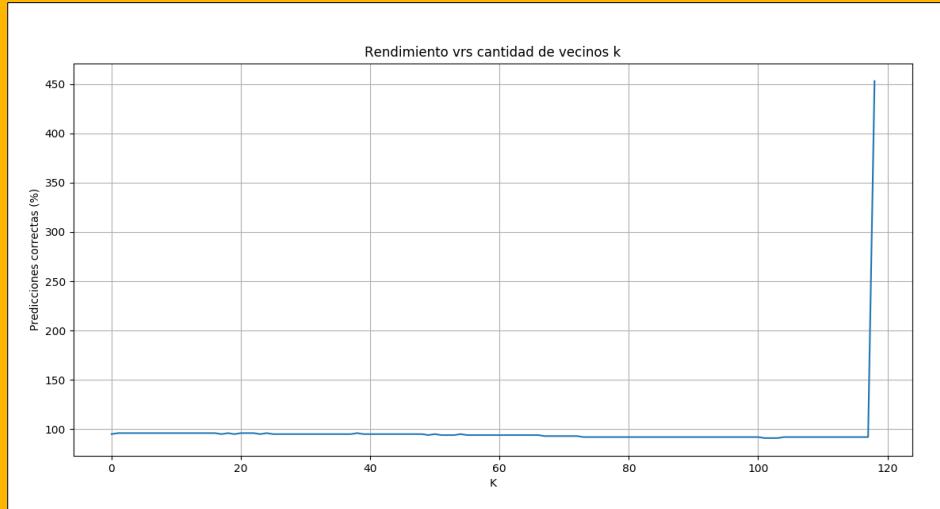
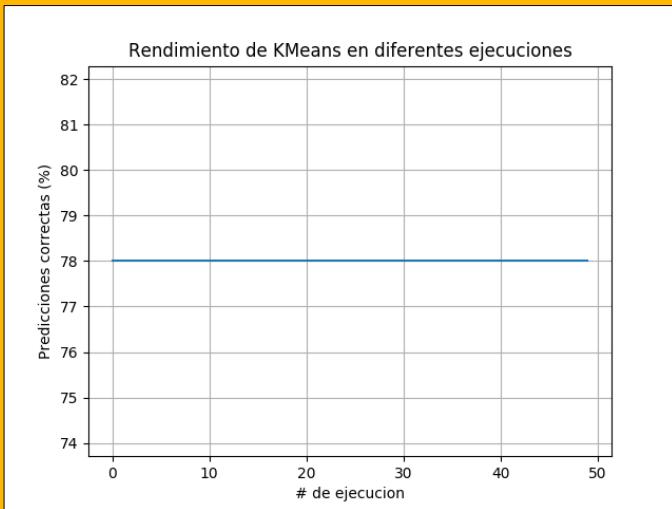
Otro extractores de características



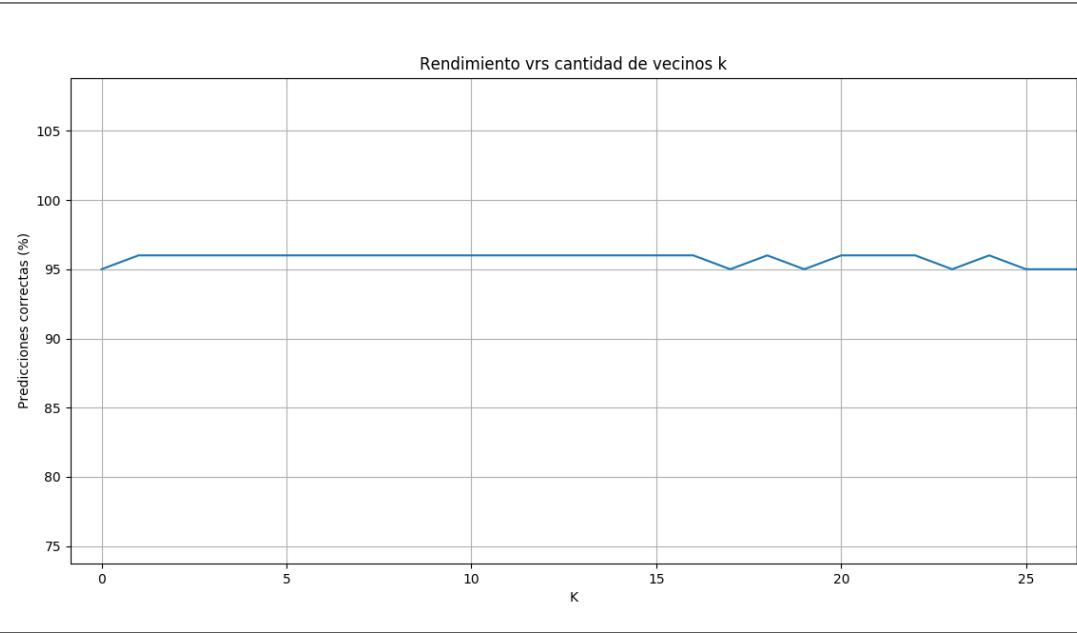
6

Clasificación

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



KNN



Implementación

<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$
```



```
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$
```



¡Gracias!



Alguna duda, pregunta o sugerencia

Github: cabustillo13

Mail: cabustillo13@hotmail.com

Linkedin: Carlos Bustillo

Telegram: @Cracken13

