

Procesamiento de imágenes con Python

¿Qué es esto?



¿Perro? ¿Ratón?



¿Muffin? ¿Quequito?

¿Qué es esto?



¿Perro? ¿Ratón? ¿Muffin? ¿Quequito?

<Contacto>

Mail: 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 variedades, 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/comenzo-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.



Modelo

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.

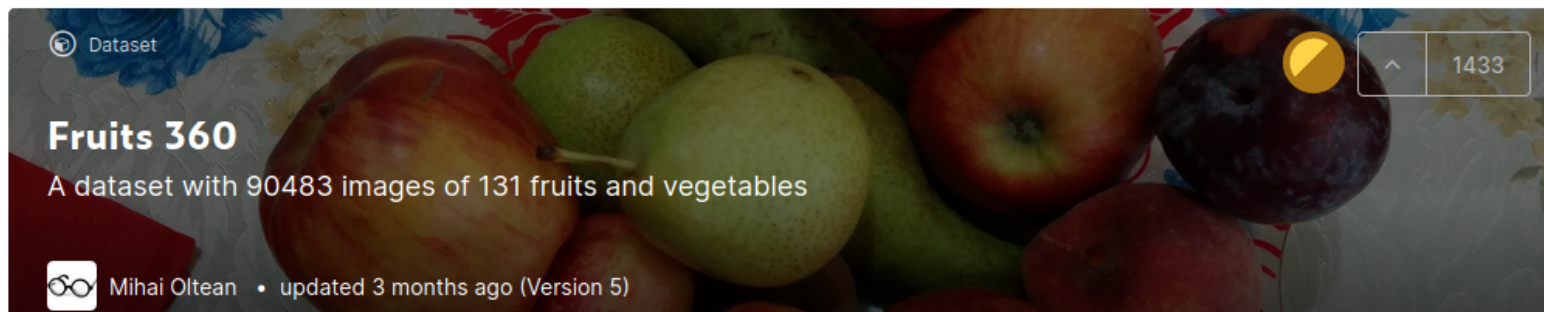
Sensores: Cámara.

Procesamiento de imágenes con Python

Paso 1/6 : Adquisición de imágenes

Vamos a utilizar un dataset de Kaggle: Fruits 360.

Y solo utilizaremos las clases grapeBlue, grapeWhite y grapePink.



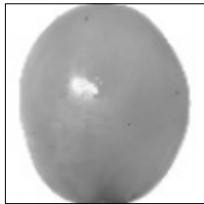
Paso 2/6 : Transformación

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.



Paso 5/6 : Extracción de rasgos

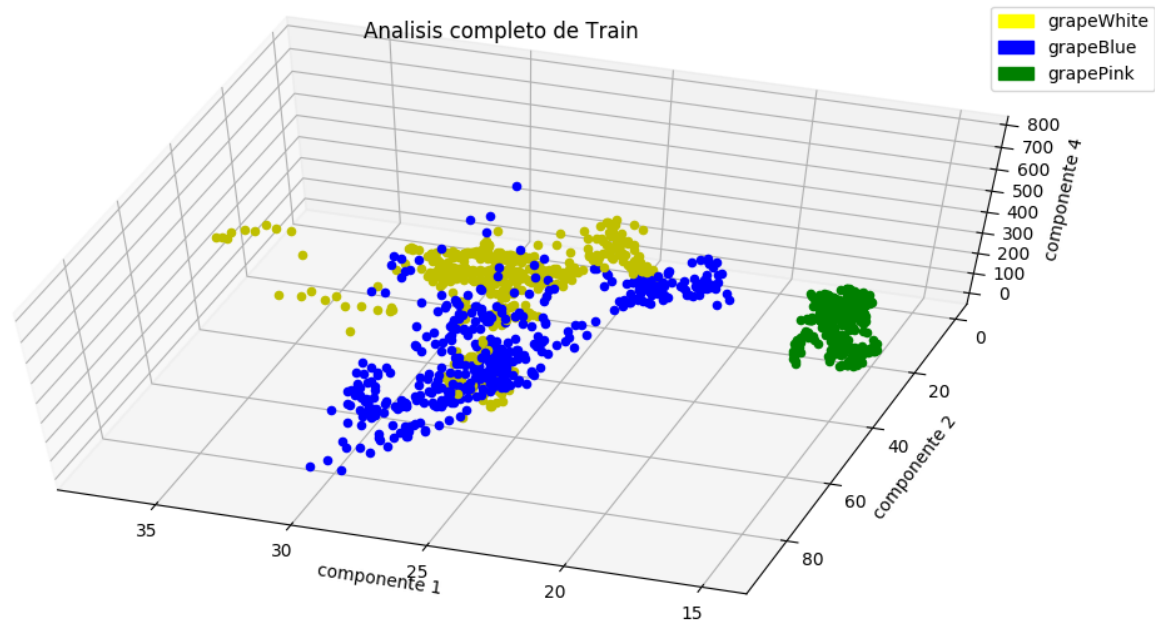
Definir los momentos de Hu 1,2 y 4.

Blue Grape : [7.266208318798105, 0.024048648552466356, 0.6853083786356352]

Pink Grape : [4.7386009969045215, 0.13936792354833638, 0.6428959835480056]

White Pink: [8.903935254086356, 0.005172179956379898, 0.04644185684799634]

Gráfico Momentos de Hu



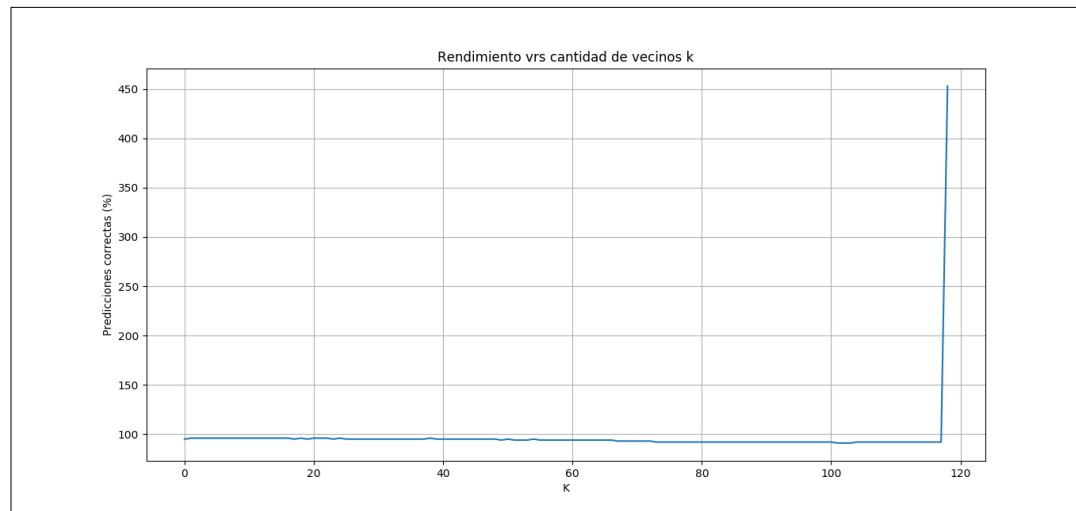
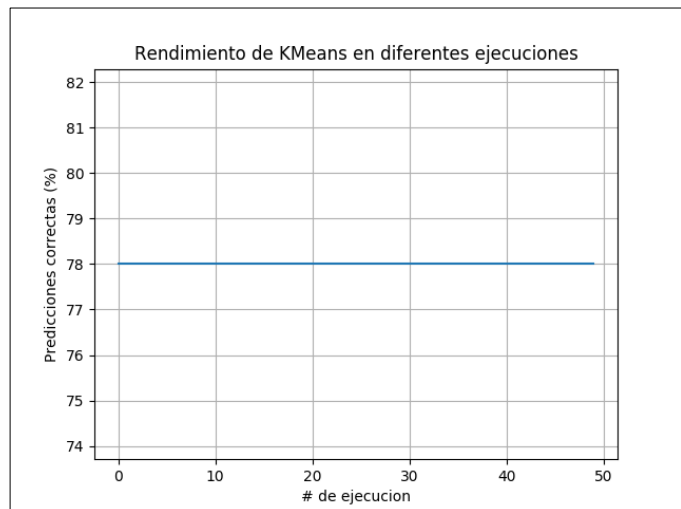
<Contacto>

Mail: cabustillo13@hotmail.com Github: cabustillo13

</Contacto>

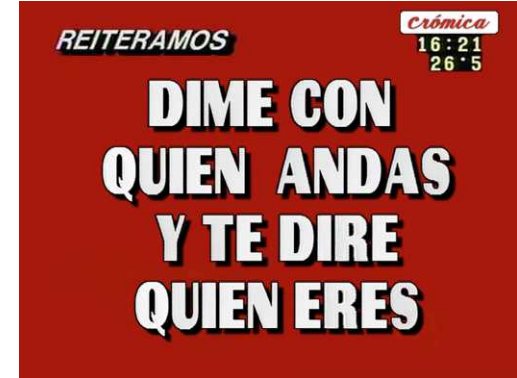
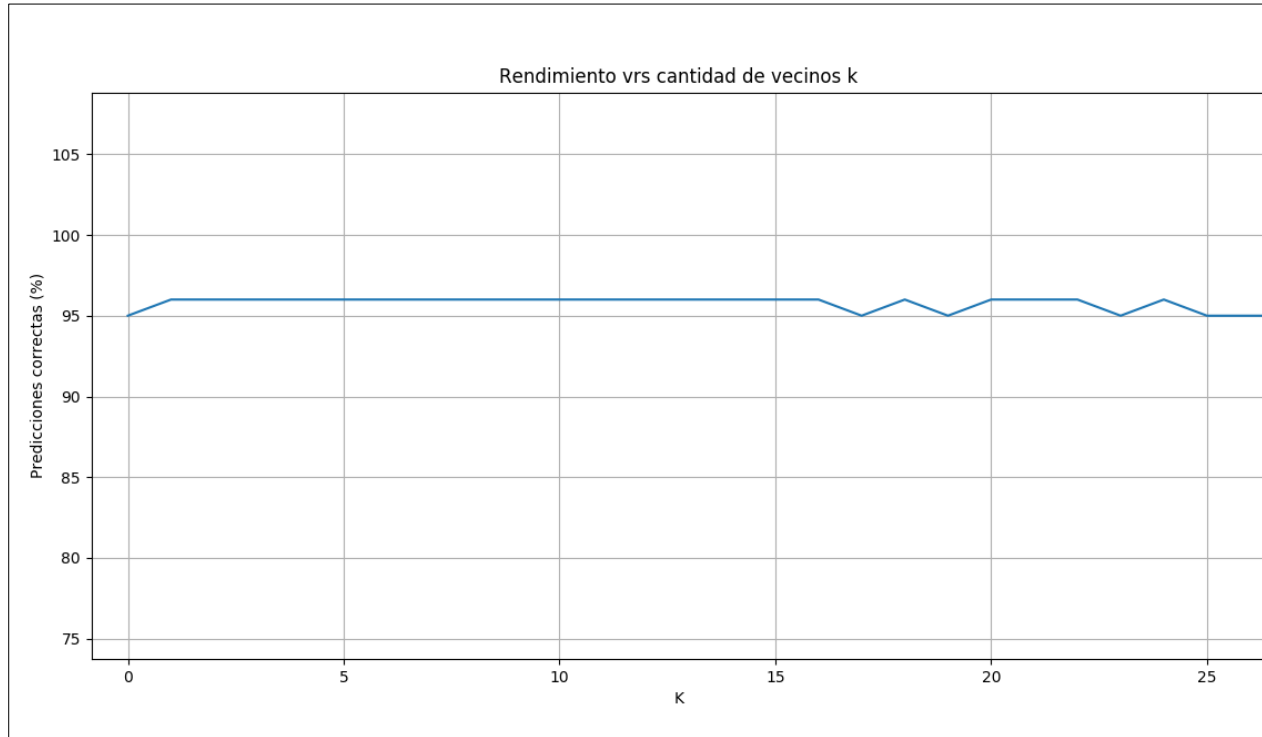
Paso 6/6 : Clasificación

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 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
Análisis completo de la base de datos de Train
Cantidad de imágenes analizadas:
1470
Introduce número de la foto: 0

Inicialización 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
Análisis completo de la base de datos de Train
Cantidad de imágenes analizadas:
1470
Introduce número de la foto: 13

Inicialización KNN

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



photo13.jpg

<Contacto>

Mail: cabustillo13@hotmail.com Github: cabustillo13

</Contacto>

Preguntas,
dudas o
sugerencias