

Herramientas para Laboratorio: Visión Artificial

José Carlos Rangel

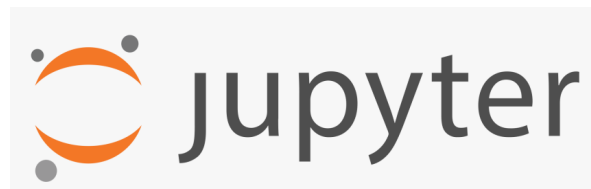
2 de septiembre de 2022

1. Herramientas de Desarrollo

1.1. Python



1.2. Jupyter



Jupyter Notebook es un **ambiente interactivo de desarrollo** basado en web para *notebooks*, código y datos. Su interfaz flexible permite a los usuarios configurar y organizar *workflows* en **ciencia de los datos, computación científica, periodismo computacional y aprendizaje automático**. Su diseño modular permite a las extensiones expandir y enriquecer su funcionalidad. Diferentes kernels permiten ejecutar lenguajes como **Python**, R, C++, Julia, Kotlin, Ruby, entre otros.

1.3. Google Colab



Colaboratory, o “Colab” para abreviar, es un producto de **Google Research**. Permite a cualquier usuario escribir y ejecutar código arbitrario de **Python** en el navegador. Es especialmente adecuado para tareas de **aprendizaje automático, análisis de datos y educación**. Desde un punto de vista más técnico, Colab es un servicio alojado de **Jupyter Notebook** que no requiere configuración y que ofrece acceso gratuito a recursos informáticos, como **GPUs**.

1.4. Conexión con Google Drive

- Permite la conexión con Google Drive para utilizar este servicio como almacenamiento de datos.
- Para esto se debe agregar a nuestro notebook la siguiente línea de código, la cual estable el enlace con nuestro *drive*.

```
from google.colab import drive
drive.mount('/content/drive', force_remount=True)
```

- Una vez montado y ejecutada la celda, la ruta a nuestras carpetas y localización de nuestro archivos generalmente tiene la siguiente estructura. Se recomienda no utilizar espacios en el nombre de la carpeta.

```
'/content/drive/MyDrive/CARPETA/ARCHIVO'
```

Utilizando este fragmento de código se puede configurar la librería `OS` para que cuando se deba leer algún archivo, y se necesite la ruta del sistema, esta se pueda obtener utilizando el comando `os.getcwd()`.

```
# Define functions to connect to Google and change directories
def connectDrive():
    from google.colab import drive
    drive.mount('/content/drive', force_remount=True)

def changeDirectory(path):
    import os
    original_path = os.getcwd()
    os.chdir(path)
    new_path = os.getcwd()
    print("Original path: ",original_path)
    print("New path: ",new_path)

connectDrive() # Connect to Google Drive

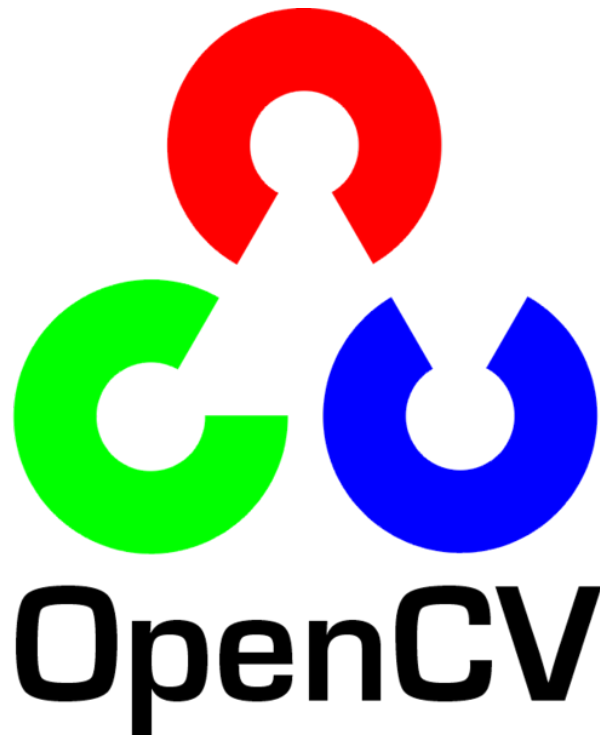
# Change path
```

```
changeDirectory("/content/drive/MyDrive/LaboratoriosVisionArtificial")
```

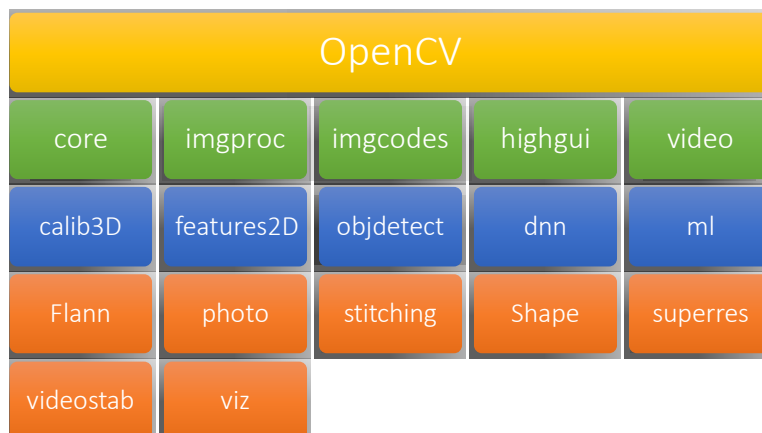
Para utilizar las funciones recién creadas, debemos modificar la manera en la cual hacemos referencia a nuestra imagen en el código usando el siguiente fragmento donde indicamos la ruta de nuestra imagen en función de la ubicación en Google Drive:

```
current_path = os.getcwd()  
path = '/img/tower.jpg'  
im = cv2.imread(current_path + path)
```

1.5. OpenCV



- Es una librería *open source* de visión por computador, análisis de imagen y con algunas herramientas de aprendizaje automático.
- Para ello dispone de infinidad de algoritmos que permiten, con sólo escribir unas pocas líneas de código, identificar rostros, reconocer objetos, clasificarlos, detectar movimientos de manos, etc.
- Multiplataforma
- Puede programarse con C, C++, Python, Java, Matlab, entre otros.
- Se ha convertido en el librería por defecto para el desarrollo de aplicaciones con visión artificial.



Incluye los siguientes módulos

- Procesamiento de Imágenes
- Análisis de Vídeo
- Calibración de Cámaras
- Reconstrucción 3D
- Características 2D
- Detección de Objetos
- Interfaces de alto nivel
- Vídeo I/O

Dentro de sus aplicaciones se puede encontrar

- Interacción Humano Computador - HCI
- Identificación de Objetos
- Reconocimiento de Objetos
- Reconocimiento de Rostros
- Reconocimiento de Gestos
- Rastreo de Movimiento
- Procesamiento de Imágenes
- Robótica Móvil
- Realidad Aumentada
- Registro de imágenes de vista de calle
- Vigilancia Automatizada
- Análisis de Imágenes Médicas
- *Egomotion*
- ... entre otras

Características

- Puede procesar al rededor de 30 frames por segundo
- Se puede integrar con diferentes entornos de desarrollo
- Optimizada para la ejecución
- Gran comunidad de desarrolladores que también dan soporte
- Soporte para el *Machine Learning*
- Cuenta con más de 2500 algoritmos optimizados y del estado del arte en visión computacional

1.5.1. CV2 *imshow* en Google Colab

Debido al funcionamiento de colab esta función de OpenCV es incompatible con el entorno, por lo cual Google utiliza una modificación que nos permitirá emular el funcionamiento de esta ventana dentro del entorno de Colab. Se debe importar primero y luego su uso es de una manera muy sencilla imitando el `cv2.imshow('Imagen en Escala de Grises', gray)`.

```
# Importar
from google.colab.patches import cv2_imshow

# Forma de Uso
cv2_imshow(gray)
```
