

# INFORME DE INSTALACION E IMPLEMENTACION DE OPENCV CON PYTHON PARA EJECUTAR UN CODIGO AL ARRANCAR EL PC

## Introduccion:

El siguiente documento muestra el proceso realizado para instalar python y la librería opencv en ubuntu 16.04 aplicando el uso de ambientes virtuales con el fin de proteger los archivos del python real. Una vez realizada la instalación se muestran los pasos para hacer que el código creado se ejecute al iniciar el computador.

Todo se realizo siguiendo los tutoriales del doctor Adrian Rosebrock presentes en los siguientes enlaces:

<https://www.pyimagesearch.com/2016/10/24/ubuntu-16-04-how-to-install-opencv/>

<https://www.pyimagesearch.com/2016/05/16/running-a-python-opencv-script-on-reboot/>

Después que se esta ejecutando el código debidamente se realizan unas modificaciones para que las fotos sean copiadas a una USB extraible inmediatamente después de que se han tomado.

## Instalacion de python y Opencv

Lo primero es actualizar y mejorar el equipo:

```
$ sudo apt-get update  
$ sudo apt-get upgrade
```

Luego, se instalan algunas herramientas de desarrollador

```
$ sudo apt-get install build-essential cmake pkg-config
```

Se instalan los complementos necesarios para cargar imágenes de diferentes formatos

```
sudo apt-get install libjpeg8-dev libtiff5-dev libjasper-dev libpng12-dev
```

Tambien se instalan paquetes que permitan acceder a las cámaras de video incluyendo sus frames

```
$ sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev  
$ sudo apt-get install libxvidcore-dev libx264-dev
```

El nombre del modulo que mantiene OpenCV GUI en operación depende de la librería GTK por lo que se instala

```
sudo apt-get install libgtk-3-dev
```

Ahora se instalan las librerias que son usadas para optimizar funcionalidades internas de OpenCV

```
sudo apt-get install libatlas-base-dev gfortran
```

También se instalan los encabezados tanto de python 2.7 como de 3.5

```
$ sudo apt-get install python2.7-dev python3.5-dev
```

Ahora se deben descargar los archivos .zip de opencv y sus contribuciones

```
$ cd ~  
$ wget -O opencv.zip https://github.com/Itseez/opencv/archive/3.1.0.zip  
$ unzip opencv.zip  
$ wget -O opencv_contrib.zip https://github.com/Itseez/opencv_contrib/archive/3.1.0.zip  
$ unzip opencv_contrib.zip
```

Antes de poder comenzar a configurar python es necesario instalar el paquete pip

```
$ cd ~  
$ wget https://bootstrap.pypa.io/get-pip.py  
$ sudo python get-pip.py
```

Por medio del pip se instalan los paquetes de ambientes virtuales

```
$ sudo pip install virtualenv virtualenvwrapper  
$ sudo rm -rf ~/get-pip.py ~/.cache/pip
```

para poder usar las opciones de ambientes virtuales se debe configurar el archivo .bashrc ya que es un script carcasa que se ejecuta cada vez que se accede a un terminal nuevo, se puede editar de la siguiente forma

```
$ echo -e "\n# virtualenv and virtualenvwrapper" >> ~/.bashrc  
$ echo "export WORKON_HOME=$HOME/.virtualenvs" >> ~/.bashrc  
$ echo "source /usr/local/bin/virtualenvwrapper.sh" >> ~/.bashrc
```

se cargan los cambios

```
$ source ~/.bashrc
```

**NOTA:** Las líneas que se adicionan al archivo .bashrc también deben adicionarse en el archivo .profile ubicado en /home/julian/.profile.

ahora ya se pueden crear ambientes virtuales, en este caso se creó uno con python 3

```
$ mkvirtualenv cv -p python3
```

se entiende que se está en el ambiente virtual por que el nombre antecede el nombre del usuario en el terminal, para entrar en el ambiente virtual se puede ejecutar el comando *workon cv* se instala la librería numpy

```
$ pip install numpy
```

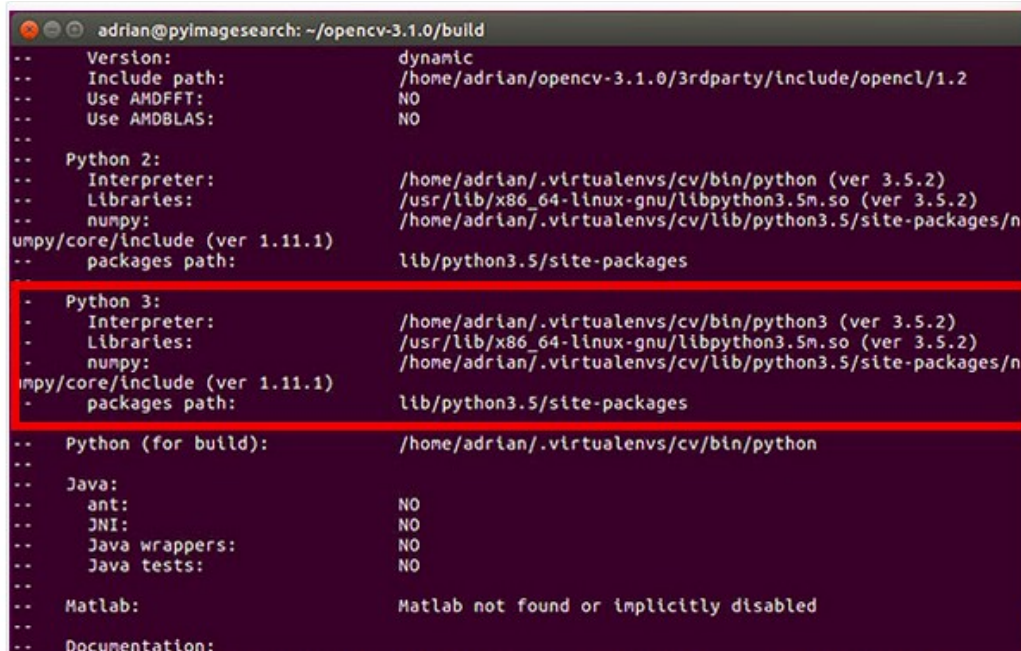
Ahora se instala Opencv

```

$ cd ~/opencv-3.1.0/
$ mkdir build
$ cd build
$ cmake -D CMAKE_BUILD_TYPE=RELEASE \
-D CMAKE_INSTALL_PREFIX=/usr/local \
-D INSTALL_PYTHON_EXAMPLES=ON \
-D INSTALL_C_EXAMPLES=OFF \
-D OPENCV_EXTRA_MODULES_PATH=~/opencv_contrib-3.1.0/modules \
-D PYTHON_EXECUTABLE=~/.virtualenvs/cv/bin/python \
-D BUILD_EXAMPLES=ON ..

```

Una vez termine el procedimiento de instalación del opencv se va a obtener en el terminal lo siguiente



```

adrian@pyimagesearch: ~/opencv-3.1.0/build
-- Version: dynamic
-- Include path: /home/adrian/opencv-3.1.0/3rdparty/include/opencv/1.2
-- Use AMDFFT: NO
-- Use AMDBLAS: NO
--
-- Python 2:
-- Interpreter: /home/adrian/.virtualenvs/cv/bin/python (ver 3.5.2)
-- Libraries: /usr/lib/x86_64-linux-gnu/libpython3.5m.so (ver 3.5.2)
-- numpy: /home/adrian/.virtualenvs/cv/lib/python3.5/site-packages/numpy/core/include (ver 1.11.1)
-- packages path: lib/python3.5/site-packages
--
-- Python 3:
-- Interpreter: /home/adrian/.virtualenvs/cv/bin/python3 (ver 3.5.2)
-- Libraries: /usr/lib/x86_64-linux-gnu/libpython3.5m.so (ver 3.5.2)
-- numpy: /home/adrian/.virtualenvs/cv/lib/python3.5/site-packages/numpy/core/include (ver 1.11.1)
-- packages path: lib/python3.5/site-packages
--
-- Python (for build): /home/adrian/.virtualenvs/cv/bin/python
--
-- Java:
-- ant: NO
-- JNI: NO
-- Java wrappers: NO
-- Java tests: NO
--
-- Matlab: Matlab not found or implicitly disabled
--
-- Documentation:

```

Para terminar la instalación es necesario decirle al sistema cuantos procesadores de la maquina pueden estar destinados a trabajar, en el caso del pc del ejemplo tiene cuatro procesadores

```
$ make -j4
```

seguido a esto ejecutar los comando de compilación

```

$ make clean
$ make

```

y los comandos de instalacion

```

$ sudo make install
$ sudo ldconfig

```

Para verificar la ubicación del archivo en /usr/local/lib/python3.5/site-packages/

```
$ ls -l /usr/local/lib/python3.5/site-packages/
```

Del que se obtiene una respuesta similar a

```
total 1972
```

```
-rw-r--r-- 1 root staff 2016816 Sep 13 17:24 cv2.cpython-35m-x86_64-linux-gnu.so
```

Continuando, para arreglar un problema del nombre con el que se hace el llamado de opencv

```
cd /usr/local/lib/python3.5/site-packages/  
sudo mv cv2.cpython-35m-x86_64-linux-gnu.so cv2.so
```

por ultimo se hace el enlace de opencv con el ambiente virtual

```
cd ~/.virtualenvs/cv/lib/python3.5/site-packages/  
ln -s /usr/local/lib/python3.5/site-packages/cv2.so cv2.so
```

Por ultimo se prueba la instalación y su comportamiento con los siguientes comandos

```
$ cd ~  
$ workon cv  
$ python  
import cv2  
cv2.__version_  
'3.1.0'
```

Se pueden borrar los archivos descargados

```
cd ~  
rm -rf opencv-3.1.0 opencv_contrib-3.1.0 opencv.zip opencv_contrib.zip
```

## Ejecutar script al arrancar el pc

Lo primero que se debe hacer es crear el archivo ejecutable que va iniciar el ambiente virtual y va a llamar el código e iniciarlo, en este caso el ejecutable debe terminar con el indicativo .sh

```
on_reboot.sh
```

```
#!/bin/bash
```

```
source /home/julian/.profile  
workon cv  
cd /home/julian/julian-reboot  
python rafaga.py
```

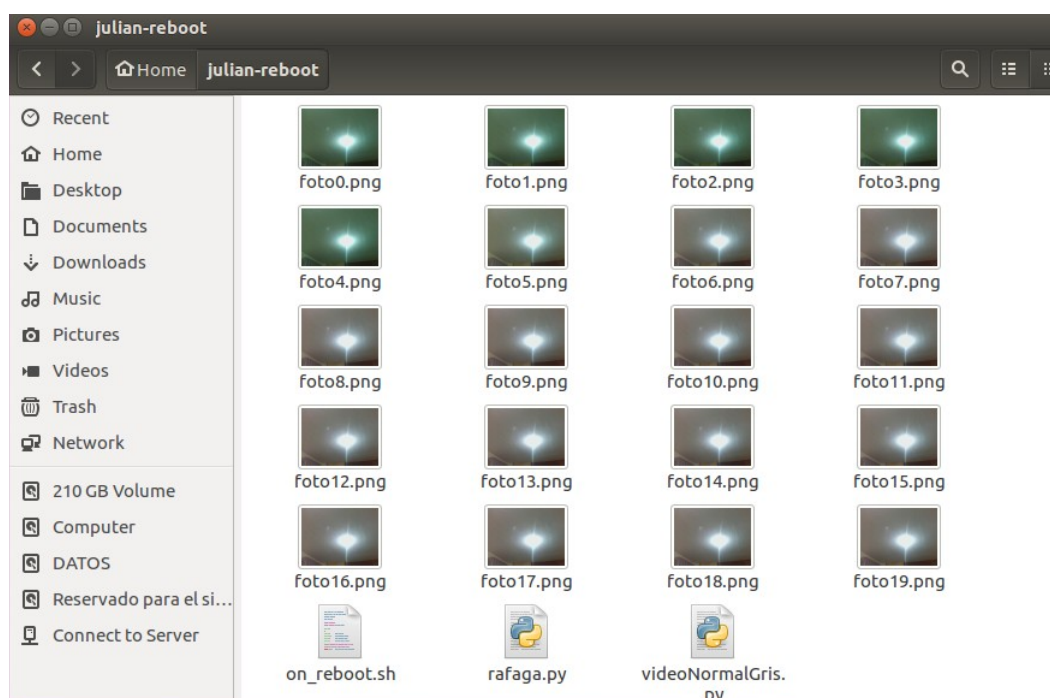
La finalidad de acceder al .profile es entablar el enlace con el ambiente virtual para luego poder llamarlo con el comando workon cv, luego viaja a la carpeta donde esta el código y finalmente ejecuta el código .py. Seguidamente se hace el archivo ejecutable

```
$chmod +x on_reboot.sh
```

finalmente se modifica el archivo cron que se encarga de llamar el archivo ejecutable, para eso hay que escribir los siguientes comandos

```
$sudo crontab -e  
@reboot /home/julian/julian-reboot/on_reboot.sh
```

Al reiniciar el computador se accede a la carpeta para asegurarse que el código se ha ejecutado correctamente y que los resultados son los esperados.



## copiando las fotos a una USB

Para que los archivos sean copiados se adicionan las siguientes líneas al final del archivo ejecutable *on\_reboot.sh*

```
cd ~  
cd /media/julian/JA_MTTTO/fotos-code
```

```
sudo cp /home/julian/julian-reboot/foto* .
```

Lo que se hace básicamente es viajar a la ubicación de la carpeta destino en la USB, en este caso es una carpeta llamada *fotos-code* de la USB *JA\_MTTO* y seguidamente se realiza el copiado de las fotos con el comando *cp* que llama cada imagen en la carpeta donde se corrió el código, como cada imagen tiene el nombre *foto* y *un numero* se pueden llamar todas las fotos usando el símbolo (\*) el punto siguiente separado por un espacio significa la dirección actual que en este caso es la carpeta en la USB.