

Instalación de OpenCV 4.1.1 en Raspberry pi 4

Paso 1: expanda el sistema de archivos

```
sudo raspi-config
```

Luego selecciona el siguiente

- Opciones avanzadas> A1 Expandir sistema de archivos> Presione "Enter"
Mostrará un mensaje "La partición raíz ha sido redimensionada".

Entonces necesitas reiniciar tu pi usando el siguiente comando.

```
sudo shutdown -r now
```

Paso 2: libera espacio (opcional)

```
sudo apt-get purge wolfram-engine  
sudo apt-get purge libreoffice*  
sudo apt-get clean  
sudo apt-get autoremove
```

Paso 3: Instalar dependencias

- El primer paso es actualizar y actualizar cualquier paquete existente:

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

- Luego reinicia tu pi.

```
sudo shutdown -r now
```

Después de que arranque, vuelva a iniciar la Terminal. Haz lo siguiente.

- Instalar paquetes de desarrollador CMAKE

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

- Instalar paquetes de E / S de imagen

```
sudo apt-get install libjpeg-dev libtiff5-dev libjasper-dev libpng-dev
```

- Instalar paquetes de E / S de video

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

- Instale la biblioteca de desarrollo GTK para ventanas básicas de GUI

```
sudo apt-get install libfontconfig1-dev libcairo2-dev  
sudo apt-get install libgdk-pixbuf2.0-dev libpangol.0-dev  
sudo apt-get install libgtk2.0-dev libgtk-3-dev
```

Instalar paquetes de optimización (operaciones matriciales mejoradas para OpenCV)

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

Los siguientes requisitos previos son para conjuntos de datos HDF5 y GUI Qt:

```
sudo apt-get install libhdf5-dev libhdf5-serial-dev libhdf5-103
sudo apt-get install libqtgui4 libqtwebkit4 libqt4-test python3-pyqt5
```

Por último, instalemos archivos de encabezado de Python 3 para que podamos compilar OpenCV con enlaces de Python:

```
sudo apt-get install python3-dev
```

Paso 4: Instala Python 3, setuptools, dev y Numpy

- Instalar Python 3 y numpy

```
wget https://bootstrap.pypa.io/get-pip.py
sudo python get-pip.py
sudo python3 get-pip.py
sudo rm -rf ~/.cache/pip
```

Paso 5: Descargue el OpenCV 3.4 y contrib módulos adicionales

```
cd ~
wget -O opencv.zip https://github.com/opencv/opencv/archive/4.1.1.zip
wget -O opencv_contrib.zip https://github.com/opencv/opencv_contrib/archive/4.1.1.zip
unzip opencv.zip
unzip opencv_contrib.zip
mv opencv-4.1.1 opencv
mv opencv_contrib-4.1.1 opencv_contrib
```

Paso 7: Cambie el tamaño del espacio antes de compilar para agregar más memoria virtual

Permitirá que OpenCV **compile con los cuatro núcleos** del Raspberry PI sin problemas de memoria.

Abra su **/etc/dphys-swapfile** y luego edite la variable **CONF_SWAPSIZE**

```
sudo nano /etc/dphys-swapfile
```

Se abrirá el editor nano para editar CONF_SWAPSIZE. Cámbielo como a continuación:

```
# set size to absolute value, leaving empty (default) then uses computed value
# you most likely don't want this, unless you have an special disk situation
# CONF_SWAPSIZE=100
CONF_SWAPSIZE=2048
```

A continuación, guarde los cambios que ha realizado, pulse **Ctrl + O** . Para salir nano, escriba **Ctrl + X** . Si le pide a nano que salga de un archivo modificado, le preguntará si desea guardarlo. Simplemente presione **N**

caso de que no lo haga, o **Y** en caso de que lo haga. Luego le pedirá un nombre de archivo. Simplemente escríbalo y presiona **Enter**.

Luego escribe las siguientes líneas para llevarlo a cabo

```
sudo /etc/init.d/dphys-swapfile stop
sudo /etc/init.d/dphys-swapfile start
```

Paso : Instalar numpy

```
pip install numpy
```

Paso 6: compila e instala OpenCV 3.4.0 para Python 3

```
cd ~/opencv
mkdir build
cd build
cmake -D CMAKE_BUILD_TYPE=RELEASE \
-D CMAKE_INSTALL_PREFIX=/usr/local \
-D OPENCV_EXTRA_MODULES_PATH=~/opencv_contrib/modules \
-D ENABLE_NEON=ON \
-D ENABLE_VFPV3=ON \
-D BUILD_TESTS=OFF \
-D INSTALL_PYTHON_EXAMPLES=OFF \
-D OPENCV_ENABLE_NONFREE=ON \
-D CMAKE_SHARED_LINKER_FLAGS=-latomic \
-D BUILD_EXAMPLES=OFF ..
```

Paso 7: finalmente listo para ser compilado

Escriba el siguiente comando para compilarlo usando 4 núcleos de pi

```
make -j4
```

Paso 8: instala la compilación en raspberry pi

```
sudo make install
```

```
sudo ldconfig
```

Paso 10: probando la instalación de OpenCV 4.1.1

```
pi@raspberrypi:~ $ python3
Python 3.5.3 (default, Jan 19 2017, 14:11:04)
[GCC 6.3.0 20170124] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> cv2.__version__
'4.1.1'
```

Paso 11: ¡no olvide cambiar el tamaño de su intercambio!

Abra su **/etc/dphys-swapfile** y luego edite la variable **CONF_SWAPSIZE**

```
sudo nano /etc/dphys-swapfile
```

Se abrirá el editor nano para editar **CONF_SWAPSIZE** . Cámbielo como a continuación:

```
# set size to absolute value, leaving empty (default) then uses computed value
# you most likely don't want this, unless you have an special disk situation
CONF_SWAPSIZE=100
# CONF_SWAPSIZE=2048
```

A continuación, guarde los cambios que ha realizado, pulse **Ctrl + O** . Para salir nano, escriba **Ctrl + X** . Si le pide a nano que salga de un archivo modificado, le preguntará si desea guardarlo. Simplemente presione **N**

caso de que no lo haga, o **Y** en caso de que lo haga. Luego le pedirá un nombre de archivo. Simplemente escríbalo y presiona **Enter**.

Luego escribe las siguientes líneas para llevarlo a cabo

```
sudo /etc/init.d/dphys-swapfile stop
sudo /etc/init.d/dphys-swapfile start
```
