

Guía técnica: Instalación de Python 3.10 y OpenCV 4.10.0 con soporte CUDA

Esta guía describe el procedimiento actualizado para compilar e instalar **Python 3.10.19** en Ubuntu 24.04, crear un entorno virtual y compilar **OpenCV 4.10.0** con soporte CUDA y cuDNN dentro de dicho entorno. La versión 4.10.0 se fija explícitamente para garantizar reproducibilidad en entornos NVIDIA Jetson o GPU discreta.

1. Instalación compilada de Python 3.10.19

- 1 Actualizar el sistema: `sudo apt update && sudo apt upgrade -y`
- 2 Instalar dependencias de compilación: `sudo apt install -y build-essential zlib1g-dev libncurses5-dev libgdbm-dev libnss3-dev libssl-dev libreadline-dev libffi-dev libsqlite3-dev wget libbz2-dev`
- 3 Descargar el código fuente: `cd /usr/src && sudo wget https://www.python.org/ftp/python/3.10.19/Python-3.10.19.tgz`
- 4 Extraer y compilar: `sudo tar -xvf Python-3.10.19.tgz && cd Python-3.10.19`
- 5 Configurar compilación: `./configure --enable-optimizations --with-ensurepip=install`
- 6 Compilar e instalar: `sudo make -j$(nproc) && sudo make altinstall`
- 7 Verificar instalación: `python3.10 --version`

2. Creación de entorno virtual con Python 3.10

- 1 Crear carpeta para entornos: `mkdir -p ~/environments`
- 2 Crear entorno virtual: `python3.10 -m venv ~/environments/opencv_env`
- 3 Activar entorno: `source ~/environments/opencv_env/bin/activate`
- 4 Actualizar pip y herramientas base: `pip install --upgrade pip setuptools wheel`

3. Instalación de OpenCV 4.10.0 con soporte CUDA

- 1 Instalar dependencias del sistema:
- 2 `sudo apt install -y cmake gfortran libgtk-3-dev libjpeg-dev libpng-dev libtiff-dev libavcodec-dev libavformat-dev libswscale-dev libv4l-dev libxvidcore-dev libx264-dev libopenblas-dev libatlas-base-dev liblapack-dev libhdf5-dev gstreamer1.0-tools libqt5gui5 libqt5webkit5 libqt5test5 libqt5core5a`
- 3 Clonar repositorios de OpenCV y contrib:
- 4 `cd ~/ && git clone https://github.com/opencv/opencv.git && git clone https://github.com/opencv/opencv_contrib.git`
- 5 Fijar versión 4.10.0 en ambos repositorios:
- 6 `cd ~/opencv && git checkout 4.10.0`
- 7 `cd ~/opencv_contrib && git checkout 4.10.0`
- 8 Crear carpeta de compilación: `cd ~/opencv && mkdir build && cd build`

- 9 Ejecutar CMake con soporte CUDA/cuDNN:
- 10 `cmake -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=$(python3 -c 'import sys; print(sys.prefix)') -D PYTHON_EXECUTABLE=$(which python) -D OPENCV_EXTRA_MODULES_PATH=~/.opencv_contrib/modules -D WITH_CUDA=ON -D WITH_CUDNN=ON -D ENABLE_FAST_MATH=1 -D CUDA_FAST_MATH=1 -D WITH_CUBLAS=1 ..`
- 11 Compilar e instalar: `make -j$(nproc) && make install`
- 12 Verificar versión: `python -c 'import cv2; print(cv2.__version__)'` (debería mostrar 4.10.0)

4. Validación del entorno

- 1 Verificar versiones activas:
- 2 `which python && python --version`
- 3 `pip list | grep opencv`
- 4 Probar soporte CUDA desde Python:
- 5 `python - <<'EOF' import cv2; print('CUDA Enabled:', cv2.cuda.getCudaEnabledDeviceCount() > 0) EOF`