Technical Guide: Installing Python 3.10 and OpenCV 4.10.0 with CUDA Support

This guide provides the updated procedure to compile and install **Python 3.10.19** on Ubuntu 24.04, create a virtual environment, and build **OpenCV 4.10.0** with CUDA and cuDNN support inside that environment. The 4.10.0 version is explicitly fixed to ensure reproducibility across NVIDIA Jetson and discrete GPU environments.

1. Compiling and Installing Python 3.10.19

1. Update and upgrade system packages: sudo apt update && sudo apt upgrade -y
2. Install build dependencies: 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. Download Python source: cd /usr/src && sudo wget https://www.python.org/ftp/python/3.10.19/Python-3.10.19.tgz
4. Extract and compile: sudo tar -xvf Python-3.10.19.tgz && cd Python-3.10.19
5. Configure the build: ./configure --enable-optimizations --with-ensurepip=install
6. Compile and install: sudo make -j$(nproc) && sudo make altinstall
7. Verify installation: python3.10 --version

2. Creating a Python 3.10 Virtual Environment

1. Create a directory for environments: mkdir -p ~/environments
2. Create the virtual environment: python3.10 -m venv ~/environments/opencv_env
3. Activate the virtual environment: source ~/environments/opencv_env/bin/activate
4. Upgrade pip and base tools: pip install --upgrade pip setuptools wheel

3. Installing OpenCV 4.10.0 with CUDA Support

1. Install system dependencies:
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. Clone the OpenCV repositories:
4. cd ~/ && git clone https://github.com/opencv/opencv.git && git clone https://github.com/opencv/opencv_contrib.git
5. Fix the version to 4.10.0 in both repositories:
6. cd ~/opencv && git checkout 4.10.0
7. cd ~/opencv_contrib && git checkout 4.10.0
8. Create the build directory: cd ~/opencv && mkdir build && cd build
9. Run CMake with CUDA/cuDNN enabled:

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. Compile and install: make -j$(nproc) && make install
12. Verify version: python -c 'import cv2; print(cv2.__version__)' (should display 4.10.0)

4. Environment Validation

1. Check active versions:
2. which python && python --version
3. pip list | grep opencv
4. Test CUDA support from Python:
5. python - <<'EOF' import cv2; print('CUDA Enabled:', cv2.cuda.getCudaEnabledDeviceCount() > 0) EOF