

**You said:**

install opencv 4.2 and contrib in ubuntu linux

ChatGPT said:

To install **OpenCV 4.2** with the **contrib modules** on **Ubuntu Linux**, follow these steps. This process compiles OpenCV from source, which gives you full flexibility and access to all contrib modules.

---

## ✓ Step-by-Step Installation of OpenCV 4.2 + Contrib on Ubuntu

### 1. Install Dependencies

```
bash
CopyEdit
sudo apt update
sudo apt install -y build-essential cmake git pkg-config libgtk-3-dev \
    libavcodec-dev libavformat-dev libswscale-dev libv4l-dev \
    libxvidcore-dev libx264-dev libjpeg-dev libpng-dev libtiff-dev \
    gfortran openexr libatlas-base-dev python3-dev python3-numpy \
    libtbb2 libtbb-dev libdc1394-22-dev
```

### 2. Clone OpenCV and OpenCV-Contrib (v4.2.0)

```
bash
CopyEdit
cd ~
git clone -b 4.2.0 https://github.com/opencv/opencv.git
git clone -b 4.2.0 https://github.com/opencv/opencv_contrib.git
```

### 3. Create Build Directory

```
bash
CopyEdit
cd ~/opencv
mkdir build
cd build
```

### 4. Configure with CMake

```
bash
CopyEdit
cmake -D CMAKE_BUILD_TYPE=Release \
    -D CMAKE_INSTALL_PREFIX=/usr/local \
    -D OPENCV_EXTRA_MODULES_PATH=~/opencv_contrib/modules \
    -D BUILD_EXAMPLES=ON ..
```

### 5. Compile (This may take 20–90 minutes)

Use `-j` with number of CPU cores, e.g., `-j$(nproc)`

```
bash
CopyEdit
make -j$(nproc)
```

## 6. Install

```
bash
CopyEdit
sudo make install
sudo ldconfig
```

## 7. Verify Installation

Check version with Python:

```
bash
CopyEdit
python3 -c "import cv2; print(cv2.__version__)"
```

It should show 4.2.0.

---

Would you like help setting up a Python virtual environment with OpenCV 4.2 too?

40

**You said:**

```
cmake -D CMAKE_BUILD_TYPE=Release \ -D CMAKE_INSTALL_PREFIX=/usr/local \ -D
OPENCV_EXTRA_MODULES_PATH=~/opencv_contrib/modules \ -D
BUILD_EXAMPLES=ON .. complie wit gpu
```

ChatGPT said:

To compile **OpenCV 4.2** with **GPU (CUDA) support**, you need to make sure your system has:

- An NVIDIA GPU
- CUDA and cuDNN installed
- Correct version compatibility between OpenCV 4.2, CUDA, and your NVIDIA driver

---

## ✓ Update CMake Command for GPU (CUDA) Support

Here's the modified cmake command:

```
bash
CopyEdit
cmake -D CMAKE_BUILD_TYPE=Release \
-D CMAKE_INSTALL_PREFIX=/usr/local \
-D OPENCV_EXTRA_MODULES_PATH=~/opencv_contrib/modules \
-D WITH_CUDA=ON \
-D ENABLE_FAST_MATH=1 \
-D CUDA_FAST_MATH=1 \
-D WITH_CUBLAS=1 \
-D OPENCV_DNN_CUDA=ON \
-D BUILD_EXAMPLES=ON ..
```

---

## Additional Setup Steps

### 1. Install CUDA and cuDNN

If you haven't already:

```
bash
CopyEdit
sudo apt install nvidia-cuda-toolkit
```

⚠ However, it's better to install CUDA and cuDNN directly from NVIDIA's website to match your GPU.

### 2. Check CUDA Availability

After installing, verify:

```
bash
CopyEdit
nvcc --version
```

Make sure it's detected correctly.

### 3. Optional: Set Compute Capability

To optimize for your GPU, add:

```
bash
CopyEdit
-D CUDA_ARCH_BIN="7.5" \
```

Replace 7.5 with your GPU's compute capability ([check here](#)).

Example:

```
bash
CopyEdit
-D CUDA_ARCH_BIN="6.1" # For GTX 1050 Ti
```

---

### ✓ After CMake:

```
bash
CopyEdit
make -j$(nproc)
sudo make install
sudo ldconfig
```

## Test CUDA in OpenCV

```
python
CopyEdit
import cv2
print(cv2.getBuildInformation())
```

Look for:

- CUDA = YES

- `cuDNN = YES` (if used)
- `NVIDIA GPU arch = present`

Would you like help verifying if your GPU is compatible with CUDA and OpenCV 4.2?