

OpenALPR Compilation from sources on CentOS 7

- Written by Juan Gabriel Covas 2021

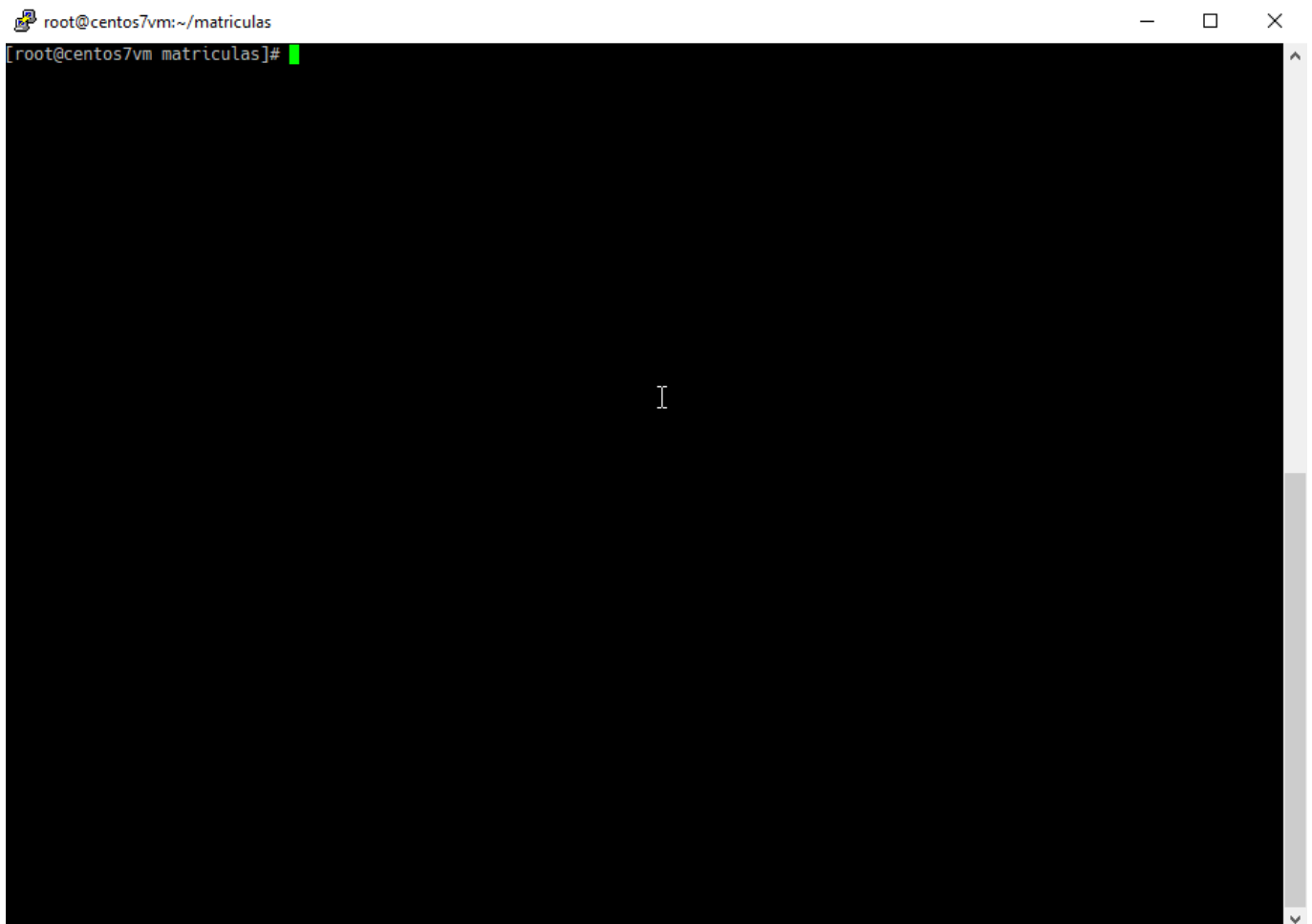
This is my guide to compile **OpenALPR** under Linux CentOS 7 / RedHat, which was a pain since I didn't find any other similar how-to: the official page about OpenALPR just talks about Ubuntu.

[OpenALPR](#) is an open source Automatic License Plate Recognition library written in C++ [...] that analyzes images and video streams to identify license plates. The output is the text of any license plate characters found.

Under Windows you can easily test OpenALPR using the [official OpenALPR windows installers](#), but we want binaries for Linux CentOS...

The "rabbit hole" reference:

- OpenALPR github wiki about [OpenALPR compilation instructions for Ubuntu Linux - The Easy Way](#)



Too Long, Won't read

If you're feeling lucky, you could put all of this on a bash shell script and execute it to see if you end up having the alpr command available

```
# some "easy" pre-requisites to compile openalpr
sudo yum -y install epel-release
sudo yum -y install git wget
sudo yum -y groupinstall "Development tools"
sudo yum -y install cmake cmake3 curl curl-devel log4cplus-devel

# get OpenALPR sources
mkdir ~/src; cd ~/src
git clone https://github.com/openalpr/openalpr.git
cd ~/src/openalpr/src && mkdir build; cd build

# compile Tesseract (and Leptonica), required by OpenALPR
sudo yum -y install autoconf-archive libjpeg-turbo-devel libpng-devel
libtiff-devel zlib-devel
cd ~/src
wget http://www.leptonica.org/source/leptonica-1.81.1.tar.gz
tar -zxvf leptonica-1.81.1.tar.gz
cd leptonica-1.81.1
./configure
make
sudo make install

cd ~/src
wget
https://github.com/tesseract-ocr/tesseract/archive/refs/tags/4.1.1.tar.gz -O
tesseract-4.1.1.tar.gz
tar -zxvf tesseract-4.1.1.tar.gz
cd tesseract-4.1.1
./autogen.sh
export PKG_CONFIG_PATH=/usr/local/lib/pkgconfig && export
LIBLEPT_HEADERSDIR=/usr/local/include
./configure --with-extra-includes=/usr/local/include --with-extra-
libraries=/usr/local/lib LDFLAGS="-L/usr/local/lib" CFLAGS="-
I/usr/local/include"
make -j$(nproc)
sudo make install
sudo ldconfig
cd ~/src/openalpr/runtime_data/ocr/tessdata
wget https://github.com/tesseract-ocr/tessdata/raw/master/spa.traineddata

# compile OpenCV, required by OpenALPR
sudo yum -y install epel-release git gcc gcc-c++ cmake3 qt5-qtbase-devel
python python-devel python-pip cmake python-devel python34-numpy gtk2-devel
```

```
libpng-devel jasper-devel openexr-devel libwebp-devel libjpeg-turbo-devel
libtiff-devel libdc1394-devel tbb-devel numpy eigen3-devel gstreamer-
plugins-base-devel freeglut-devel mesa-libGL mesa-libGL-devel boost boost-
thread boost-devel libv4l-devel

mkdir ~/src/opencv_build; cd ~/src/opencv_build
git clone https://github.com/opencv/opencv.git
git clone https://github.com/opencv/opencv_contrib.git
cd ~/src/opencv_build/opencv && mkdir build; cd build
cmake3 -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=/usr/local -D
INSTALL_C_EXAMPLES=ON -D INSTALL_PYTHON_EXAMPLES=ON -D
OPENCV_GENERATE_PKGCONFIG=ON -D
OPENCV_EXTRA_MODULES_PATH=~/src/opencv_build/opencv_contrib/modules -D
BUILD_EXAMPLES=ON ..
# this is going to take some minutes...
make -j$(nproc)
sudo make install
sudo ln -s /usr/local/lib64/pkgconfig/opencv4.pc /usr/share/pkgconfig/
sudo ldconfig
pkg-config --modversion opencv4

# finally, compile OpenALPR
cd ~/src/openalpr/src/build
cmake3 -DCMAKE_CXX_FLAGS="-std=c++11" -DCMAKE_INSTALL_PREFIX:PATH=/usr/local
-DCMAKE_INSTALL_SYSCONFDIR:PATH=/etc ..
make
sudo make install
```

Finally as **root** user:

```
# if [ ! -f /etc/ld.so.conf.d/usrlocal.conf ] ;then echo "/usr/local/lib" >
/etc/ld.so.conf.d/usrlocal.conf; echo "/usr/local/lib64"
>>/etc/ld.so.conf.d/usrlocal.conf; fi
# sudo ldconfig -v
```

Test it:

```
# alpr --version

alpr version: 2.3.0
```

```
# alpr --help
[...]
```

LONG VERSION AHEAD:

Step 1. Prepare after a fresh MINIMAL CentOS 7 installation

This guide assumes:

1. A sweet, bare-bones, really Minimal CentOS 7 installation, because I love it that way.
2. Commands will be issued by a non-root user that can sudo.

We'll need EPEL repo, wget, git and basic development tools like a C/C++ compiler etc.

```
$ sudo yum -y install epel-release
$ sudo yum -y install git wget
$ sudo yum groupinstall "Development tools"
```

Now we can install cmake (2) and cmake3 (3.17) which is found on EPEL repo

```
# sudo yum install cmake cmake3
```

Now the problem is cmake command is 2.8 and cmake3 the version we need. Thanks to this guide about [Installing the latest git/cmake versions on RHEL/Centos](#).

The relevant part is to execute the following two commands, assuming you have installed cmake AND cmake3:

```
$ sudo alternatives --install /usr/local/bin/git git /usr/bin/git 10 \
--slave /usr/local/git-receive-pack git-receive-pack /usr/bin/git-receive-pack \
--slave /usr/local/git-shell git-shell /usr/bin/git-shell \
--slave /usr/local/git-upload-archive git-upload-archive /usr/bin/git-upload-archive \
--slave /usr/local/git-upload-pack git-upload-pack /usr/bin/git-upload-pack \
--family git

$ sudo alternatives --install /usr/local/bin/git git /usr/local/git/bin/git 20 \
--slave /usr/local/git-cvsserver git-cvsserver /usr/local/git/bin/git-cvsserver \
--slave /usr/local/git-receive-pack git-receive-pack /usr/local/git/bin/git-receive-pack \
--slave /usr/local/git-shell git-shell /usr/local/git/bin/git-shell \
--slave /usr/local/git-upload-archive git-upload-archive /usr/local/git/bin/git-upload-archive \
--slave /usr/local/git-upload-pack git-upload-pack /usr/local/git/bin/git-upload-pack \
--slave /usr/local/gitk gitk /usr/local/git/bin/gitk \
--family git
```

Test cmake version:

```
$ cmake --version
cmake3 version 3.17.5
```

So in case we need to switch back cmake to be 2 or 3, we'll be able to change the default cmake version using:

```
$ sudo alternatives --config cmake
```

Step 2. Let's try to compile OpenALPR, that requires Tesseract

Prepare to build OpenALPR:

```
$ mkdir ~/src; cd ~/src
$ git clone https://github.com/openalpr/openalpr.git
$ cd ~/src/openalpr/src && mkdir build; cd build
```

First shot, **note** that `CMAKE_INSTALL_PREFIX` has to be `/usr/local` in CentOS distro, otherwise problems will arise regarding "Set runtime path" to a blank path for binaries and libraries when we execute the command `sudo make install` later.

```
$ cmake -DCMAKE_INSTALL_PREFIX:PATH=/usr/local -
DCMAKE_INSTALL_SYSCONFDIR:PATH=/etc ..
[...]
**ERROR: Required library Tesseract NOT FOUND.
```

If you get this other error: `CMake 3.1 or higher is required. You are running version 2.8.12.2` then you didn't properly follow the step 1 of this guide, or just use `cmake3` command instead of `cmake`.

So... **ERROR: Required library Tesseract NOT FOUND.**

Fine... **Tesseract** is a an *OCR engine* that OpenALPR needs so we need to install it too.

A) The Short path: we can install the default, Old tesseract version from our distro repos, which is 3.04 and it's aged.

```
$ sudo yum install tesseract tesseract-devel tesseract-langpack-spa
```

Or we can go...

B) The Long path: install the shiny Tesseract version 4, reference:

<https://www.hoangdung.net/2020/01/how-to-install-tesseract-4-on-centos-7.html>

First we need to compile **Leptonica** (a software for image processing and analysis), which is a dependency of Tesseract.

```
$ sudo yum -y install autoconf-archive libjpeg-turbo-devel libpng-devel
```

```
libtiff-devel zlib-devel
$ cd ~/src
$ wget http://www.leptonica.org/source/leptonica-1.81.1.tar.gz
$ tar -zxvf leptonica-1.81.1.tar.gz
$ cd leptonica-1.81.1
$ ./configure
$ make
$ sudo make install
```

Check if libpng is installed: type `whereis libpng` and expect to see a directory; a blank line is not good

Check if leptonica is installed: type `ls /usr/local/include` and expect to see "leptonica"

Now let's compile the latest Tesseract released, 4.1.1 at the time of this writing. The process was smooth too:

```
$ cd ~/src
$ wget
https://github.com/tesseract-ocr/tesseract/archive/refs/tags/4.1.1.tar.gz -O
tesseract-4.1.1.tar.gz
$ tar -zxvf tesseract-4.1.1.tar.gz
$ cd tesseract-4.1.1
$ ./autogen.sh
$ export PKG_CONFIG_PATH=/usr/local/lib/pkgconfig && export
LIBLEPT_HEADERSDIR=/usr/local/include
$ ./configure --with-extra-includes=/usr/local/include --with-extra-
libraries=/usr/local/lib LDFLAGS="-L/usr/local/lib" CFLAGS="-
I/usr/local/include"
$ make -j$(nproc)
$ sudo make install
$ sudo ldconfig
```

We have additional trained data here: <https://github.com/tesseract-ocr/tessdata/>

```
$ sudo yum install mlocate
$ sudo updatedb
$ locate traineddata
$ cd ~/src/openalpr/runtime_data/ocr/tessdata
```

Latest Spanish trained data:

```
$ wget https://github.com/tesseract-ocr/tessdata/raw/master/spa.traineddata
```

Step 3. We have Tesseract, now we need OpenCV

Let's try to configure openalpr using cmake again:

```
$ cd ~/src/openalpr/src/build
```

The next complain will be about OpenCV not installed.

```
$ cmake -DCMAKE_INSTALL_PREFIX:PATH=/usr/local -  
DCMAKE_INSTALL_SYSCONFDIR:PATH=/etc ..  
[...]  
blablah OpenCV blah blah
```

So we need **OpenCV**, a real-time optimized Computer Vision library, another dependency of *OpenALPR*.

We have a “opencv-devel” package from official CentOS 7 repos, but...

```
$ sudo yum list opencv-devel
```

OpenCV from CentOS 7 repos is version 2.4.5 which won't be enough:

```
$ cmake -DCMAKE_INSTALL_PREFIX:PATH=/usr/local -  
DCMAKE_INSTALL_SYSCONFDIR:PATH=/etc ..  
[...]  
ERROR: OpenCV version is not compatible : 2.4.5
```

Sweet... Let's cleanup if we installed it:

```
$ sudo yum remove opencv-devel opencv opencv-core
```

Why not, we're going to build OpenCV from sources to get OpenCV **4.x**, which is a bit of extra pain, but this guide helped: <https://linuxize.com/post/how-to-install-opencv-on-centos-7/>

```
$ sudo yum -y install epel-release git gcc gcc-c++ cmake3 qt5-qtbase-devel  
python python-devel python-pip cmake python-devel python34-numpy gtk2-devel  
libpng-devel jasper-devel openexr-devel libwebp-devel libjpeg-turbo-devel  
libtiff-devel libdc1394-devel tbb-devel numpy eigen3-devel gstreamer-  
plugins-base-devel freeglut-devel mesa-libGL mesa-libGL-devel boost boost-  
thread boost-devel libv4l-devel
```

```
$ mkdir ~/src/opencv_build; cd ~/src/opencv_build
```

```
$ git clone https://github.com/opencv/opencv.git  
$ git clone https://github.com/opencv/opencv_contrib.git
```

```
$ cd ~/src/opencv_build/opencv && mkdir build; cd build
```

```
$ cmake3 -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=/usr/local -D
```

```
INSTALL_C_EXAMPLES=ON -D INSTALL_PYTHON_EXAMPLES=ON -D
OPENCV_GENERATE_PKGCONFIG=ON -D
OPENCV_EXTRA_MODULES_PATH=~/.src/opencv_build/opencv_contrib/modules -D
BUILD_EXAMPLES=ON ..
```

This is going to take some minutes (nproc here is to inject number of CPUs available):

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

Everything ok? Then

```
$ sudo make install
```

```
$ sudo ln -s /usr/local/lib64/pkgconfig/opencv4.pc /usr/share/pkgconfig/
$ sudo ldconfig
```

Check the OpenCV version available now:

```
$ pkg-config --modversion opencv4
4.5.3
```

Step 4. More dependencies

Let's try again to configure OpenALPR

TLDR; `sudo yum install curl curl-devel log4cplus-devel`

```
$ cd ~/.src/openalpr/src/build
```

```
$ cmake -DCMAKE_INSTALL_PREFIX:PATH=/usr/local -
DCMAKE_INSTALL_SYSCONFDIR:PATH=/etc ..
[...]
ERROR: Could NOT find CURL (missing: CURL_LIBRARY CURL_INCLUDE_DIR)
```

Great... let's install curl-devel

```
$ sudo yum install curl curl-devel
```

Trying again...

```
$ cmake -DCMAKE_INSTALL_PREFIX:PATH=/usr/local -
DCMAKE_INSTALL_SYSCONFDIR:PATH=/etc ..
[...]
ERROR: Required library log4cplus NOT FOUND.
```


Fine, let's install log4cplus-devel

```
$ sudo yum install log4cplus-devel
```

Should be OK now!

```
$ cmake -DCMAKE_INSTALL_PREFIX:PATH=/usr/local -  
DCMAKE_INSTALL_SYSCONFDIR:PATH=/etc ..  
[...]  
-- Build files have been written to: ../src/openalpr/src/build
```

Step 5. Dependencies resolved, let's try to make

Let's finally compile this thing.

TLDR;

cmake..., make, sudo make install



```
$ cd ~/src/openalpr/src/build  
$ cmake -DCMAKE_CXX_FLAGS="-std=c++11" -  
DCMAKE_INSTALL_PREFIX:PATH=/usr/local -  
DCMAKE_INSTALL_SYSCONFDIR:PATH=/etc ..  
$ make  
$ sudo make install
```

The alpr command should be available now

Long version:

```
$ cd ~/src/openalpr/src/build
```

Let's try make:

```
$ make  
[...] error: #error "OpenCV 4.x+ requires enabled C++11 support"  
[...]  
make[2]: *** [video/CMakeFiles/video.dir/videobuffer.cpp.o] Error 1
```

It's complaining about enabling C++11 support which we can fix using `-DCMAKE_CXX_FLAGS="-std=c++11"` at cmake call (took me a bit of research to fix):

Cleanup first:

```
$ make clean
```

Redo the cmake call:

```
$ cmake -DCMAKE_CXX_FLAGS="-std=c++11" -  
DCMAKE_INSTALL_PREFIX:PATH=/usr/local -DCMAKE_INSTALL_SYSCONFDIR:PATH=/etc  
..
```

Try again make and it should go well now:

```
$ make
```

```
$ sudo make install
```

Everything ok?

Finally, alpr command available, but it works?

Let's try if the alpr command is finally available:

```
$ alpr  
alpr: error while loading shared libraries: libopenalpr.so.2: cannot open  
shared object file: No such file or directory
```

Sweet... Another clusterf*ck I had to research a bit.

Create a new file named `/etc/ld.so.conf.d/usrlocal.conf` and put these lines:

```
/usr/local/lib  
/usr/local/lib64
```

```
$ sudo ldconfig -v
```

Let's try again:

```
$ alpr --version  
alpr version: 2.3.0
```

Woohoo! alpr command actually works

Get an image of a vehicle plate, say `plate.jpg`.



```
$ alpr --help
```

Try an image (by default "us" country is used).

```
$ alpr plate.jpg
```

You can try "eu" as country and limit guess at 2 top matches:

```
$ alpr -c eu -n 2 plate.jpg
plate0: 2 results
- 2008ZGZ    confidence: 94.4568
- 20Q8ZGZ    confidence: 87.0557
```

~~DISCUSSION|Comentarios~~

From:
<https://www.juangacovas.info/> - JuangaCovas.info

Permanent link:
<https://www.juangacovas.info/doku.php/linux/howtos/centos/openalpr-compile-from-sources?rev=1629567851>

Last update: **21/08/2021 19:44**

