

目 录

1	所需要的库及其依赖关系	3
2	准备工作	3
3	各个平台配置	4
3.1	安装设置	4
3.1.1	armv7-linux 平台	4
3.1.2	aarch64-linux 平台	4
3.1.3	armv7-android 平台	4
3.1.4	aarch64-android 平台	5
3.2	zlib 安装	5
3.2.1	armv7-linux 平台	5
3.2.2	aarch64-linux 平台	5
3.2.3	armv7-android 平台	6
3.2.4	aarch64-android 平台	6
3.3	libpng 安装	6
3.3.1	armv7-linux 平台	6
3.3.2	aarch64-linux 平台	7
3.3.3	armv7-android 平台	7
3.3.4	aarch64-android 平台	7
3.4	libjpeg 安装	8
3.4.1	armv7-linux 平台	8
3.4.2	aarch64-linux 平台	8
3.4.3	armv7-android 平台	8
3.4.4	aarch64-android 平台	9
3.5	yasm 安装	9
3.5.1	armv7-linux 平台	9

3.5.2	aarch64-linux 平台	9
3.5.3	armv7-android 平台	10
3.5.4	aarch64-android 平台	10
3.6	x264 安装	10
3.6.1	armv7-linux 平台	10
3.6.2	aarch64-linux 平台	11
3.6.3	armv7-android 平台	11
3.6.4	aarch64-android 平台	12
3.7	libv4l 安装	12
3.7.1	armv7-linux 平台	12
3.8	xvid 安装	12
3.8.1	armv7-linux 平台	12
3.8.2	aarch64-linux 平台	13
3.8.3	armv7-android 平台	13
3.8.4	aarch64-android 平台	13
3.9	ffmpeg 安装	13
3.9.1	armv7-linux 平台	13
3.9.2	aarch64-linux 平台	14
3.9.3	armv7-android 平台	15
3.9.4	aarch64-android 平台	15
3.10	opencv 安装	16
3.10.1	armv7-linux 平台	16
3.10.2	aarch64-linux 平台	16
3.10.3	armv7-android 平台	17
3.10.4	aarch64-android 平台	18

1 所需要的库及其依赖关系

1. zlib
2. libpng (依赖于 zlib)
3. libjpeg
4. x264
5. yasm
6. libv4l
7. xvid
8. ffmpeg (依赖于 x264 与 xvid)
9. opencv-2.4.13 (依赖于 ffmpeg 与 zlib)

2 准备工作

1. 找到交叉编译器的路径, 如/opt/toolchains/android-toolchain-aarch64/bin/aarch64-linux-android-gcc
2. 准备好所有安装包, 如图 1所示。

```
arm-gnueabi.toolchain.cmake  libv4l.tar.bz2
cap_ffmpeg_impl.hpp         opencv-2.4.13.tar.gz
CMakeLists.txt              OpenCVFindLibsVideo.cmake
ffmpeg-1.2.12.tar.bz2      xvidcore_1.3.2.orig.tar.gz
jpegsrc.v8d.tar.gz         yasm-1.3.0.tar.gz
last_x264.tar.bz2          zlib-1.2.7.tar.gz
libpng-1.6.34.tar.gz
```

图 1: backup 文件夹内容

- 需要用 CMakeLists.txt 覆盖 Opencv 目录下原 CMakeList.txt 文件 (4 个平台均如此)。
- cap_ffmpeg_impl.hpp 以及 OpenCVFindLibsVideos.cmake 在安卓的两个平台配置时需要使用(因为编译安卓系统的库时, Opencv 默认会关闭 FFMPEG 这个依赖库的选项, 如果手动修改配置文件, 将其打开, 会出现一些报错。cap_ffmpeg_impl.hpp 是为了解决其中一个报错“找不到头文件 sys/sys***.h”而准备的, 该文件在 Opencv 目录下, 具体位置请通过 find 命令搜寻)

- OpenCVFindLibsVideos.cmake 是为了打开 FFMPEG 这个依赖库的选项而准备的。
- arm-gnueabi.toolchain.cmake 是为了配置 armv7-linux 平台的环境准备的。

3 各个平台配置

3.1 安装设置

注：以下的配置请一段一段地复制到命令行，可能过程中会出现一些小问题。复制前注意检查代码是否需要修改。

3.1.1 armv7-linux 平台

```
1 CC1='/usr/bin/arm-linux-gnueabi-gcc'
2 LD='/usr/bin/arm-linux-gnueabi-ld'
3 AR='/usr/bin/arm-linux-gnueabi-ar'
4 RANLIB='/usr/bin/arm-linux-gnueabi-ranlib'
5 STRIP='/usr/bin/arm-linux-gnueabi-strip'
6 PREFIX='/home/liuzili/arm-linux-build' # 安装目录
7 HOST='arm-linux' # 编译出的库在什么平台上运行
```

3.1.2 aarch64-linux 平台

```
1 CC1='/opt/toolchains/linux-toolchain-aarch64/bin/aarch64-linux-gnu-gcc'
2 LD='/opt/toolchains/linux-toolchain-aarch64/bin/aarch64-linux-gnu-ld'
3 AR='/opt/toolchains/linux-toolchain-aarch64/bin/aarch64-linux-gnu-ar'
4 RANLIB='/opt/toolchains/linux-toolchain-aarch64/bin/aarch64-linux-gnu-ranlib'
5 STRIP='/opt/toolchains/linux-toolchain-aarch64/bin/aarch64-linux-gnu-strip'
6 PREFIX='/home/liuzili/linux-aarch64-build'
7 HOST='aarch64-linux'
8 export LDFLAGS="-L$PREFIX/lib"
9 export CPPFLAGS="-I$PREFIX/include"
10 YOUR_BUILD_PATH='/home/liuzili/linux-aarch64'
11 TOOLCHAINS='/opt/toolchains/linux-toolchain-aarch64'
12 CROSS_PREFIX="$TOOLCHAINS/bin/aarch64-linux-gnu-"
13 ARCH="aarch64"
```

3.1.3 armv7-android 平台

```
1 CC1='/opt/toolchains/android-toolchain-armv7/bin/arm-linux-androideabi-gcc'
2 LD='/opt/toolchains/android-toolchain-armv7/bin/arm-linux-androideabi-ld'
3 AR='/opt/toolchains/android-toolchain-armv7/bin/arm-linux-androideabi-ar'
4 RANLIB='/opt/toolchains/android-toolchain-armv7/bin/arm-linux-androideabi-ranlib'
5 STRIP='/opt/toolchains/android-toolchain-armv7/bin/arm-linux-androideabi-strip'
6 PREFIX='/home/liuzili/armv7-android-build'
7 HOST='arm-linux-android'
```

```

8 TOOLCHAINS='/opt/toolchains/android-toolchain-armv7'
9 SYSROOT="$TOOLCHAINS/sysroot"
10 CROSS_PREFIX="$TOOLCHAINS/bin/arm-linux-androideabi-"
11 EXTRA_CFLAGS="-march=armv7-a -mfloat-abi=softfp -mfpu=neon -mthumb -D__ANDROID__ ...
    -D__ARM_ARCH_7__ -D__ARM_ARCH_7A__ -D__ARM_ARCH_7R__ -D__ARM_ARCH_7M__ ...
    -D__ARM_ARCH_7S__"
12 EXTRA_LDFLAGS="-nostdlib"
13 YOUR_BUILD_PATH="/home/liuzili/armv7-android"

```

3.1.4 aarch64-android 平台

```

1 CC1='/opt/toolchains/android-toolchain-aarch64/bin/aarch64-linux-android-gcc'
2 LD='/opt/toolchains/android-toolchain-aarch64/bin/aarch64-linux-android-ld'
3 AR='/opt/toolchains/android-toolchain-aarch64/bin/aarch64-linux-android-ar'
4 RANLIB='/opt/toolchains/android-toolchain-aarch64/bin/aarch64-linux-android-ranlib'
5 STRIP='/opt/toolchains/android-toolchain-aarch64/bin/aarch64-linux-android-strip'
6 PREFIX='/home/liuzili/test-aarch64/3rdpart-aarch64-install'
7 HOST='aarch64-linux-android'
8 TOOLCHAINS='/opt/toolchains/android-toolchain-aarch64'
9 SYSROOT="$TOOLCHAINS/sysroot"
10 CROSS_PREFIX="$TOOLCHAINS/bin/aarch64-linux-android-"
11 YOUR_BUILD_PATH='-~/ad-aarch64'
12 EXTRA_CFLAGS="-march=armv8-a -D__ANDROID__ -D__ARM_ARCH_8__ -D__ARM_ARCH_8A__"
13 EXTRA_LDFLAGS="-nostdlib"

```

3.2 zlib 安装

zlib 一般最先编译，因为 libpng 以及后面的 ffmpeg 的编译会依赖它。通过 prefix 指定安装目录，sed 语句将 Makefile 文件中的“CC”，“LDSHARED”等项替换为交叉编译器的对应项。

3.2.1 armv7-linux 平台

```

1 tar -zxvf zlib-1.2.7.tar.gz;
2 cd zlib-1.2.7/;
3 ./configure --prefix=$PREFIX --shared;
4 sed -i "s|CC=gcc|CC=$CC1|g" Makefile;
5 sed -i "s|LDSHARED=gcc|LDSHARED=$CC1|g" Makefile;
6 sed -i "s|CPP=gcc|CPP=$CC1|g" Makefile;
7 sed -i "s|AR=ar|AR=$AR|g" Makefile;
8 sed -i "s|RANLIB=ranlib|RANLIB=$RANLIB|g" Makefile;
9 make;
10 make install;
11 cd ..;

```

3.2.2 aarch64-linux 平台

```

1 tar -zxvf zlib-1.2.7.tar.gz;

```

```

2 cd zlib-1.2.7/;
3 ./configure --prefix=$PREFIX --shared;
4 sed -i "s|CC=gcc|CC=$CC1|g" Makefile;
5 sed -i "s|LD_SHARED=gcc|LD_SHARED=$CC1|g" Makefile;
6 sed -i "s|CPP=gcc|CPP=$CC1|g" Makefile;
7 sed -i "s|AR=ar|AR=$AR|g" Makefile;
8 sed -i "s|RANLIB=ranlib|RANLIB=$RANLIB|g" Makefile;
9 make -j;
10 make install;
11 cd ..;

```

3.2.3 armv7-android 平台

```

1 tar -zxvf zlib-1.2.7.tar.gz;
2 cd zlib-1.2.7/;
3 ./configure --prefix=$PREFIX --shared;
4 sed -i "s|CC=gcc|CC=$CC1|g" Makefile;
5 sed -i "s|LD_SHARED=gcc|LD_SHARED=$CC1|g" Makefile;
6 sed -i "s|CPP=gcc|CPP=$CC1|g" Makefile;
7 sed -i "s|AR=ar|AR=$AR|g" Makefile;
8 sed -i "s|RANLIB=ranlib|RANLIB=$RANLIB|g" Makefile;
9 make -j;
10 make install;
11 cd ..;

```

3.2.4 aarch64-android 平台

```

1 tar -zxvf zlib-1.2.7.tar.gz;
2 cd zlib-1.2.7/;
3 ./configure --prefix=$PREFIX --shared;
4 sed -i "s|CC=gcc|CC=$CC1|g" Makefile;
5 sed -i "s|LD_SHARED=gcc|LD_SHARED=$CC1|g" Makefile;
6 sed -i "s|CPP=gcc|CPP=$CC1|g" Makefile;
7 sed -i "s|AR=ar|AR=$AR|g" Makefile;
8 sed -i "s|RANLIB=ranlib|RANLIB=$RANLIB|g" Makefile;
9 make -j;
10 make install;
11 cd ..;

```

3.3 libpng 安装

编译 libpng 时需要设置环境变量指向安装 zlib 的目录，否则会提示找不到 zlib。同样需要修改 Makefile。

3.3.1 armv7-linux 平台

```

1 export LDFLAGS="-L$PREFIX/lib";
2 export CPPFLAGS="-I$PREFIX/include";
3 tar -zxvf libpng-1.6.34.tar.gz;

```

```

4 cd libpng-1.6.34/;
5 ./configure --host=$HOST --prefix=$PREFIX --enable-shared --enable-static ...
   CC=arm-linux-gnueabi-gcc;
6 sed -i "s|CC = gcc|CC = $CC1|g" Makefile;
7 sed -i "s|LD = /usr/bin/ld|LD = $LD1|g" Makefile;
8 sed -i "s|STRIP = strip|STRIP = $STRIP1|g" Makefile;
9 sed -i "s|CPP = gcc|CPP = $CC1|g" Makefile;
10 sed -i "s|AR = ar|AR = $AR1|g" Makefile;
11 sed -i "s|RANLIB = ranlib|RANLIB = $RANLIB1|g" Makefile;
12 make -j;
13 make install;
14 cd ..;

```

3.3.2 aarch64-linux 平台

```

1 tar -zxvf libpng-1.6.34.tar.gz;
2 cd libpng-1.6.34/;
3 export LDFLAGS="-L$PREFIX/lib"
4 export CPPFLAGS="-I$PREFIX/include"
5 ./configure --host=$HOST --prefix=$PREFIX --enable-shared --enable-static CC=$CC1;
6 sed -i "s|CC = gcc|CC = $CC1|g" Makefile;
7 sed -i "s|LD = /usr/bin/ld|LD = $LD1|g" Makefile;
8 sed -i "s|STRIP = strip|STRIP = $STRIP1|g" Makefile;
9 sed -i "s|CPP = gcc|CPP = $CC1|g" Makefile;
10 sed -i "s|AR = ar|AR = $AR1|g" Makefile;
11 sed -i "s|RANLIB = ranlib|RANLIB = $RANLIB1|g" Makefile;
12 make -j;
13 make install;
14 cd ..;

```

3.3.3 armv7-android 平台

```

1 tar -zxvf libpng-1.6.34.tar.gz;
2 cd libpng-1.6.34/;
3 ./configure --host=$HOST --prefix=$PREFIX --enable-shared --enable-static CC=$CC1;
4 sed -i "s|CC = gcc|CC = $CC1|g" Makefile;
5 sed -i "s|LD = /usr/bin/ld|LD = $LD1|g" Makefile;
6 sed -i "s|STRIP = strip|STRIP = $STRIP1|g" Makefile;
7 sed -i "s|CPP = gcc|CPP = $CC1|g" Makefile;
8 sed -i "s|AR = ar|AR = $AR1|g" Makefile;
9 sed -i "s|RANLIB = ranlib|RANLIB = $RANLIB1|g" Makefile;
10 make -j;
11 make install;
12 cd ..;

```

3.3.4 aarch64-android 平台

```

1 tar -zxvf libpng-1.6.34.tar.gz;
2 cd libpng-1.6.34/;
3 ./configure --host=$HOST --prefix=$PREFIX --enable-shared --enable-static CC=$CC1;
4 sed -i "s|CC = gcc|CC = $CC1|g" Makefile;

```

```

5 sed -i "s|LD = /usr/bin/ld|LD = $LD|g" Makefile;
6 sed -i "s|STRIP = strip|STRIP = $STRIP|g" Makefile;
7 sed -i "s|CPP = gcc|CPP = $CC1|g" Makefile;
8 sed -i "s|AR = ar|AR = $AR|g" Makefile;
9 sed -i "s|RANLIB = ranlib|RANLIB = $RANLIB|g" Makefile;
10 make -j;
11 make install;
12 cd ..;

```

3.4 libjpeg 安装

3.4.1 armv7-linux 平台

```

1 tar -zxvf jpegsrc.v8d.tar.gz;
2 cd jpeg-8d/;
3 ./configure --host=$HOST --prefix=$PREFIX --enable-shared --enable-static ...
   CC=arm-linux-gnueabi-hf-gcc;
4 sed -i "s|CC = gcc -std=gnu99|CC = $CC1|g" Makefile;
5 sed -i "s|STRIP = strip|STRIP = $STRIP|g" Makefile;
6 sed -i "s|LD = /usr/bin/ld|LD = $LD|g" Makefile;
7 sed -i "s|CPP = gcc|CPP = $CC1|g" Makefile;
8 sed -i "s|AR = ar|AR = $AR|g" Makefile;
9 sed -i "s|RANLIB = ranlib|RANLIB = $RANLIB|g" Makefile;
10 make -j;
11 make install;
12 cd ..;

```

3.4.2 aarch64-linux 平台

```

1 tar -zxvf jpegsrc.v8d.tar.gz;
2 cd jpeg-8d/;
3 ./configure --host=arm-linux --prefix=$PREFIX --enable-shared --enable-static CC=$CC1;
4 sed -i "s|CC = gcc -std=gnu99|CC = $CC1|g" Makefile;
5 sed -i "s|STRIP = strip|STRIP = $STRIP|g" Makefile;
6 sed -i "s|LD = /usr/bin/ld|LD = $LD|g" Makefile;
7 sed -i "s|CPP = gcc|CPP = $CC1|g" Makefile;
8 sed -i "s|AR = ar|AR = $AR|g" Makefile;
9 sed -i "s|RANLIB = ranlib|RANLIB = $RANLIB|g" Makefile;
10 make -j;
11 make install;
12 cd ..;

```

3.4.3 armv7-android 平台

```

1 tar -zxvf jpegsrc.v8d.tar.gz;
2 cd jpeg-8d/;
3 ./configure --host=arm-linux-android --prefix=$PREFIX --enable-shared ...
   --enable-static CC=$CC1;
4 sed -i "s|CC = gcc -std=gnu99|CC = $CC1|g" Makefile;
5 sed -i "s|STRIP = strip|STRIP = $STRIP|g" Makefile;

```



```

6 sed -i "s|LD = /usr/bin/ld|LD = $LD|g" Makefile;
7 sed -i "s|CPP = gcc|CPP = $CC1|g" Makefile;
8 sed -i "s|AR = ar|AR = $AR|g" Makefile;
9 sed -i "s|RANLIB = ranlib|RANLIB = $RANLIB|g" Makefile;
10 make -j;
11 make install;
12 cd ..;

```

3.4.4 aarch64-android 平台

```

1 tar -zxvf jpegsrc.v8d.tar.gz;
2 cd jpeg-8d/;
3 ./configure --host=arm-linux-android --prefix=$PREFIX --enable-shared ...
  --enable-static CC=$CC1;
4 sed -i "s|CC = gcc -std=gnu99|CC = $CC1|g" Makefile;
5 sed -i "s|STRIP = strip|STRIP = $STRIP|g" Makefile;
6 sed -i "s|LD = /usr/bin/ld|LD = $LD|g" Makefile;
7 sed -i "s|CPP = gcc|CPP = $CC1|g" Makefile;
8 sed -i "s|AR = ar|AR = $AR|g" Makefile;
9 sed -i "s|RANLIB = ranlib|RANLIB = $RANLIB|g" Makefile;
10 make -j;
11 make install;
12 cd ..;

```

3.5 yasm 安装

3.5.1 armv7-linux 平台

```

1 tar -zxvf yasm-1.3.0.tar.gz;
2 cd yasm-1.3.0/;
3 CC=arm-linux-gnueabi-gcc;
4 ./configure --host=$HOST --prefix=$PREFIX --enable-shared --enable-static;
5 sed -i "s|CC = gcc -std=gnu99|CC = $CC1|g" Makefile;
6 sed -i "s|CPP = gcc|CPP = $CC1|g" Makefile;
7 sed -i "s|STRIP = strip|STRIP = $STRIP|g" Makefile;
8 sed -i "s|AR = ar|AR = $AR|g" Makefile;
9 sed -i "s|RANLIB = ranlib|RANLIB = $RANLIB|g" Makefile;
10 make -j;
11 make install;
12 cd ..;

```

3.5.2 aarch64-linux 平台

```

1 tar -zxvf yasm-1.3.0.tar.gz;
2 cd yasm-1.3.0/;
3 ./configure --host=$HOST --prefix=$PREFIX CC=$CC1;
4 sed -i "s|CC = gcc -std=gnu99|CC = $CC1|g" Makefile;
5 sed -i "s|CPP = gcc|CPP = $CC1|g" Makefile;
6 sed -i "s|STRIP = strip|STRIP = $STRIP|g" Makefile;
7 sed -i "s|AR = ar|AR = $AR|g" Makefile;

```

```

8 sed -i "s|RANLIB = ranlib|RANLIB = $RANLIB|g" Makefile;
9 make -j;
10 make install;
11 cd ..;

```

3.5.3 armv7-android 平台

```

1 tar -zxvf yasm-1.3.0.tar.gz;
2 cd yasm-1.3.0/;
3 ./configure --host=arm-linux-android --prefix=$PREFIX CC=$CC1;
4 sed -i "s|CC = gcc -std=gnu99|CC = $CC1|g" Makefile;
5 sed -i "s|CPP = gcc|CPP = $CC1|g" Makefile;
6 sed -i "s|STRIP = strip|STRIP = $STRIP|g" Makefile;
7 sed -i "s|AR = ar|AR = $AR|g" Makefile;
8 sed -i "s|RANLIB = ranlib|RANLIB = $RANLIB|g" Makefile;
9 make -j;
10 make install;
11 cd ..;

```

3.5.4 aarch64-android 平台

```

1 tar -zxvf yasm-1.3.0.tar.gz;
2 cd yasm-1.3.0/;
3 ./configure --host=$HOST --prefix=$PREFIX --enable-shared --enable-static CC=$CC1;
4 sed -i "s|CC = gcc -std=gnu99|CC = $CC1|g" Makefile;
5 sed -i "s|CPP = gcc|CPP = $CC1|g" Makefile;
6 sed -i "s|STRIP = strip|STRIP = $STRIP|g" Makefile;
7 sed -i "s|AR = ar|AR = $AR|g" Makefile;
8 sed -i "s|RANLIB = ranlib|RANLIB = $RANLIB|g" Makefile;
9 make -j;
10 make install;
11 cd ..;

```

3.6 x264 安装

x264 的安装容易出现奇怪的问题，如果下面的命令出现问题，请谷歌上搜寻解决方案。

3.6.1 armv7-linux 平台

```

1 tar -xjf last_x264.tar.bz2;
2 cd x264-snapshot-20171201-2245/;
3 ./configure --enable-shared --host=$HOST --disable-asm --prefix=$PREFIX ...
   CC=arm-linux-gnueabihf-gcc;
4 sed -i "s|CC=gcc|CC=$CC1|g" config.mak;
5 sed -i "s|AR=gcc-ar|AR=$AR|g" config.mak;
6 sed -i "s|RANLIB=gcc-ranlib|RANLIB=$RANLIB|g" config.mak;
7 sed -i "s|LD=gcc -o|LD=$CC1 -o|g" config.mak;
8 sed -i "s|STRIP=strip|STRIP=$STRIP|g" config.mak;

```

```

9 make -j;
10 make install;
11 cd ..;

```

3.6.2 aarch64-linux 平台

```

1 YOUR_BUILD_PATH='/home/liuzili/linux-aarch64'
2 TOOLCHAINS=/opt/toolchains/linux-toolchain-aarch64
3 CROSS_PREFIX=${TOOLCHAINS}/bin/aarch64-linux-gnu-
4 ARCH=aarch64
5 tar -xjf last_x264.tar.bz2;
6 cd x264-snapshot-20171201-2245/;
7 ./configure --cross-prefix=${CROSS_PREFIX} \
8 --host=aarch64-linux \
9 --enable-pic \
10 --enable-shared \
11 --extra-ldflags="-nostdlib" \
12 --disable-cli \
13 --prefix=${PREFIX}
14 sed -i "s|CC=gcc|CC=${CC1}|g" config.mak;
15 sed -i "s|AR=gcc-ar|AR=${AR}|g" config.mak;
16 sed -i "s|RANLIB=gcc-ranlib|RANLIB=${RANLIB}|g" config.mak;
17 sed -i "s|LD=gcc -o|LD=${CC1} -o|g" config.mak;
18 make -j;
19 make install;
20 cd ..;

```

3.6.3 armv7-android 平台

```

1 tar -xjf last_x264.tar.bz2;
2 cd x264-snapshot-20171201-2245/;
3 ./configure --prefix=${PREFIX} \
4 --host=arm-linux-androideabi \
5 --sysroot=${SYSROOT} \
6 --cross-prefix=${CROSS_PREFIX} \
7 --extra-cflags="${EXTRA_CFLAGS}" \
8 --extra-ldflags="${EXTRA_LDFLAGS}" \
9 --enable-pic \
10 --enable-static \
11 --enable-shared \
12 --enable-strip \
13 --disable-cli \
14 --disable-win32thread \
15 --disable-avs \
16 --disable-swscale \
17 --disable-lavf \
18 --disable-ffms \
19 --disable-gpac \
20 --disable-lsmash
21 make clean;
22 make STRIP= -j8 install;
23 cd ..;

```

3.6.4 aarch64-android 平台

```
1 tar -xjf last_x264.tar.bz2;
2 cd x264-snapshot-20171201-2245/;
3 ./configure --prefix=$PREFIX \
4 --host=aarch64-linux-android \
5 --sysroot=$SYSROOT \
6 --cross-prefix=$CROSS_PREFIX \
7 --extra-cflags="$EXTRA_CFLAGS" \
8 --extra-ldflags="$EXTRA_LDFLAGS" \
9 --enable-pic \
10 --enable-static \
11 --enable-shared \
12 --enable-strip \
13 --disable-cli \
14 --disable-win32thread \
15 --disable-avs --disable-swscale --disable-lavf --disable-ffms --disable-gpac ...
    --disable-lsmash
16 make clean;
17 make STRIP= -j8 install;
18 cd ..;
```

3.7 libv4l 安装

这个压缩包中为库文件，库文件直接拷贝到“安装目录/lib”文件夹下即可。
armv7-linux 以外的平台不需要安装。

3.7.1 armv7-linux 平台

```
1 cp libv4l.tar.bz2 $PREFIX/lib;
2 cd $PREFIX/lib;
3 tar -xjf libv4l.tar.bz2;
4 rm -f libv4l.tar.bz2;
5 cd -;
```

3.8 xvid 安装

注意 xviud 是在 build/generic 目录下进行操作，修改 platform.inc 文件中的内容（CC, AR, RANLIB），修改方式同前面修改 Makefile 一样，这个文件会作用于 Makefile。

3.8.1 armv7-linux 平台

```
1 tar -zxvf xvidcore_1.3.2.orig.tar.gz;
2 cd xvidcore-1.3.2/build/generic;
3 ./configure --prefix=$PREFIX --host=$HOST --disable-assembly ...
    CC=arm-linux-gnueabi-gcc;
4 sed -i "s|CC=gcc|CC=$CC1|g" platform.inc;
```

```

5 sed -i "s|AR=ar|AR=$AR|g" platform.inc;
6 sed -i "s|RANLIB=ranlib|RANLIB=$RANLIB|g" platform.inc;
7 make -j;
8 make install;
9 cd ../../..;

```

3.8.2 aarch64-linux 平台

```

1 tar -zxvf xvidcore_1.3.2.orig.tar.gz;
2 cd xvidcore-1.3.2/build/generic;
3 ./configure --prefix=$PREFIX --host=arm-linux --disable-assembly CC=$CC1;
4 sed -i "s|CC=gcc|CC=$CC1|g" platform.inc;
5 sed -i "s|AR=ar|AR=$AR|g" platform.inc;
6 sed -i "s|RANLIB=ranlib|RANLIB=$RANLIB|g" platform.inc;
7 make -j;
8 make install;
9 cd ../../..;

```

3.8.3 armv7-android 平台

```

1 tar -zxvf xvidcore_1.3.2.orig.tar.gz;
2 cd xvidcore-1.3.2/build/generic;
3 ./configure --prefix=$PREFIX --host=$HOST --disable-assembly CC=$CC1;
4 sed -i "s|CC=gcc|CC=$CC1|g" platform.inc;
5 sed -i "s|AR=ar|AR=$AR|g" platform.inc;
6 sed -i "s|RANLIB=ranlib|RANLIB=$RANLIB|g" platform.inc;
7 make -j;
8 make install;
9 cd ../../..;

```

3.8.4 aarch64-android 平台

```

1 tar -zxvf xvidcore_1.3.2.orig.tar.gz;
2 cd xvidcore-1.3.2/build/generic;
3 ./configure --prefix=$PREFIX --host=arm-linux-android --disable-assembly CC=$CC1;
4 sed -i "s|CC=gcc|CC=$CC1|g" platform.inc;
5 sed -i "s|AR=ar|AR=$AR|g" platform.inc;
6 sed -i "s|RANLIB=ranlib|RANLIB=$RANLIB|g" platform.inc;
7 make -j;
8 make install;
9 cd ../../..;

```

3.9 ffmpeg 安装

3.9.1 armv7-linux 平台

ffmpeg 是用于 mpeg4 编解码的。在安卓平台一定要 enable-static，否则 Opencv 无法依赖 ffmpeg。

```

1 tar -xjf ffmpeg-1.2.12.tar.bz2;
2 cd ffmpeg-1.2.12/
3 ./configure \
4 --enable-cross-compile \
5 --target-os=linux \
6 --cc=arm-linux-gnueabi-gcc \
7 --arch=arm \
8 --enable-shared \
9 --disable-static \
10 --enable-gpl \
11 --enable-nonfree \
12 --enable-ffmpeg \
13 --disable-ffplay \
14 --enable-ffserver \
15 --enable-swscale \
16 --enable-pthreads \
17 --enable-yasm \
18 --disable-stripping \
19 --enable-libx264 \
20 --enable-libxvid \
21 --extra-cflags=-I$PREFIX/include \
22 --extra-ldflags=-L$PREFIX/lib \
23 --prefix=$PREFIX;
24 make -j;
25 make install;
26 cd ..;

```

3.9.2 aarch64-linux 平台

```

1 tar -xjf ffmpeg-1.2.12.tar.bz2;
2 cd ffmpeg-1.2.12/
3 ./configure \
4 --enable-cross-compile \
5 --target-os=linux \
6 --cc=$CC1 \
7 --arch=arm64 \
8 --enable-shared \
9 --disable-static \
10 --enable-gpl \
11 --enable-nonfree \
12 --enable-ffmpeg \
13 --disable-ffplay \
14 --enable-ffserver \
15 --enable-swscale \
16 --enable-pthreads \
17 --enable-yasm \
18 --disable-stripping \
19 --enable-libx264 \
20 --enable-libxvid \
21 --extra-cflags="-I$PREFIX/include" \
22 --extra-ldflags="-L$PREFIX/lib" \
23 --prefix=$PREFIX;
24 make -j;
25 make install;
26 cd ..;

```

3.9.3 armv7-android 平台

```
1 tar -xjf ffmpeg-1.2.12.tar.bz2;
2 cd ffmpeg-1.2.12/
3 ./configure \
4 --enable-cross-compile \
5 --target-os=linux \
6 --cc=$CC1 \
7 --arch=arm \
8 --enable-shared \
9 --enable-static \
10 --enable-gpl \
11 --enable-nonfree \
12 --enable-ffmpeg \
13 --disable-ffplay \
14 --enable-ffserver \
15 --enable-swscale \
16 --disable-symver \
17 --enable-pthreads \
18 --disable-yasm \
19 --disable-stripping \
20 --enable-libx264 \
21 --enable-libxvid \
22 --extra-cflags=-I$PREFIX/include \
23 --extra-ldflags=-L$PREFIX/lib \
24 --prefix=$PREFIX;
25 make -j;
26 make install;
27 cd ..;
```

3.9.4 aarch64-android 平台

```
1 tar -xjf ffmpeg-1.2.12.tar.bz2;
2 cd ffmpeg-1.2.12/
3 ./configure \
4 --enable-cross-compile \
5 --target-os=linux \
6 --cc=$CC1 \
7 --arch=arm64 \
8 --enable-shared \
9 --enable-static \
10 --enable-gpl \
11 --enable-nonfree \
12 --enable-ffmpeg \
13 --disable-ffplay \
14 --enable-ffserver \
15 --enable-swscale \
16 --disable-symver \
17 --enable-pthreads \
18 --disable-yasm \
19 --disable-stripping \
20 --enable-libx264 \
21 --enable-libxvid \
22 --extra-cflags=-I$PREFIX/include \
23 --extra-ldflags=-L$PREFIX/lib \
```

```

24 --prefix=$PREFIX;
25 make -j;
26 make install;
27 cd ..;

```

3.10 opencv 安装

在安卓平台请格外注意，需要手动进行一些替换操作。

3.10.1 armv7-linux 平台

```

1 tar -zxvf opencv-2.4.13.tar.gz;
2 cd opencv-2.4.13/;
3 export LD_LIBRARY_PATH=$PREFIX/lib/;
4 export PKG_CONFIG_PATH=$PREFIX/lib/pkgconfig;
5 export PKG_CONFIG_LIBDIR=$PKG_CONFIG_LIBDIR:$PREFIX/lib/;
6 mv CMakeLists.txt CMakeLists_backup.txt;
7 cp ../../CMakeLists.txt ./;
8 mkdir -p build;
9 cd build;
10 mkdir -p install;
11 mkdir -p lib;
12 cp -r $PREFIX/lib/* install/;
13 cp -r $PREFIX/lib/* lib/;
14 cp ../../arm-gnueabi.toolchain.cmake ./;
15 cmake -DCMAKE_TOOLCHAIN_FILE=arm-gnueabi.toolchain.cmake -DCMAKE_BUILD_TYPE=Release ...
    ..;
16 #sed -i "s|WITH_LIBV4L:B00L=OFF|WITH_LIBV4L:B00L=ON|g" CMakeCache.txt;
17 make -j;
18 make install;
19 cp -r lib/ ../lib

```

3.10.2 aarch64-linux 平台

```

1 tar -zxvf opencv-2.4.13.tar.gz;
2 cd opencv-2.4.13/;
3 export LD_LIBRARY_PATH=$PREFIX/lib/;
4 export PKG_CONFIG_PATH=$PREFIX/lib/pkgconfig;
5 export PKG_CONFIG_LIBDIR=$PKG_CONFIG_LIBDIR:$PREFIX/lib/;
6 mv CMakeLists.txt CMakeLists_backup.txt;
7 cp ../../CMakeLists.txt ./;
8 mkdir -p build;
9 cd build;
10 mkdir -p install;
11 mkdir -p lib;
12 cp -r $PREFIX/lib/* install/;
13 cp -r $PREFIX/lib/* lib/;
14 cp ../../arm-gnueabi.toolchain.cmake ./;
15 sed -i '$a\set(CMAKE_C_COMPILER ...
    /opt/toolchains/linux-toolchain-aarch64/bin/aarch64-linux-gnu-gcc)' ...
    arm-gnueabi.toolchain.cmake

```



```

16 sed -i '$a\set(CMAKE_CXX_COMPILER ...
    /opt/toolchains/linux-toolchain-aarch64/bin/aarch64-linux-gnu-g++)' ...
    arm-gnueabi.toolchain.cmake
17 sed -i '$a\set(CMAKE_FIND_ROOT_PATH "$YOUR_BUILD_PATH/opencv-2.4.13/build")' ...
    arm-gnueabi.toolchain.cmake
18 sed -i '$a\set(link_directories(${PREFIX}/lib))' arm-gnueabi.toolchain.cmake
19 sed -i '$a\set(CMAKE_FIND_ROOT_PATH_MODE_PROGRAM NEVER)' arm-gnueabi.toolchain.cmake
20 sed -i '$a\set(CMAKE_FIND_ROOT_PATH_MODE_LIBRARY ONLY)' arm-gnueabi.toolchain.cmake
21 sed -i '$a\set(CMAKE_FIND_ROOT_PATH_MODE_INCLUDE ONLY)' arm-gnueabi.toolchain.cmake
22 cmake -DCMAKE_TOOLCHAIN_FILE=arm-gnueabi.toolchain.cmake -DCMAKE_BUILD_TYPE=Release ...
    .;.
23 #sed -i "s|WITH_LIBV4L:B00L=OFF|WITH_LIBV4L:B00L=ON|g" CMakeCache.txt;
24 make -j;
25 make install;
26 cp -r lib/ -/lib

```

3.10.3 armv7-android 平台

```

1 tar -zxvf opencv-2.4.13.tar.gz;
2 cd opencv-2.4.13/;
3 export LD_LIBRARY_PATH=$PREFIX/lib/;
4 export PKG_CONFIG_PATH=$PREFIX/lib/pkgconfig;
5 export PKG_CONFIG_LIBDIR=$PREFIX/lib/;
6 mv CMakeLists.txt CMakeLists_backup.txt;
7 cp ../../CMakeLists.txt ./;
8 mkdir -p build;
9 cd build;
10 mkdir -p install;
11 mkdir -p lib;
12 cp -r $PREFIX/lib/* install/;
13 cp -r $PREFIX/lib/* lib/;
14 cp ../../platforms/android/android.toolchain.cmake ./;
15 sed -i '212iset(CMAKE_C_COMPILER ...
    /opt/toolchains/android-toolchain-armv7/bin/arm-linux-androideabi-gcc)' ...
    android.toolchain.cmake
16 sed -i '213iset(CMAKE_CXX_COMPILER ...
    /opt/toolchains/android-toolchain-armv7/bin/arm-linux-androideabi-g++)' ...
    android.toolchain.cmake
17 sed -i '214iset(CMAKE_FIND_ROOT_PATH "$YOUR_BUILD_PATH/opencv-2.4.13/build")' ...
    android.toolchain.cmake
18 sed -i '215ilink_directories(${PREFIX}/lib)' android.toolchain.cmake
19 sed -i '216iset(CMAKE_FIND_ROOT_PATH_MODE_PROGRAM NEVER)' android.toolchain.cmake
20 sed -i '217iset(CMAKE_FIND_ROOT_PATH_MODE_LIBRARY ONLY)' android.toolchain.cmake
21 sed -i '218iset(CMAKE_FIND_ROOT_PATH_MODE_INCLUDE ONLY)' android.toolchain.cmake
22
23 export ANDROID_STANDALONE_TOOLCHAIN="/opt/toolchains/android-toolchain-armv7"
24 export PKG_CONFIG_PATH=$PREFIX/lib:$PKG_CONFIG_PATH
25
26 sed -i 's|OCV_OPTION(WITH_FFMPEG          "Include FFMPEG support" ...
    ON IF (NOT ANDROID AND NOT IOS))|OCV_OPTION(WITH_FFMPEG ...
    "Include FFMPEG support"          ON IF (ANDROID AND NOT ...
    IOS))|g' ../CMakeLists.txt;
27 mv ../cmake/OpenCVFindLibsVideo.cmake ../cmake/OpenCVFindLibsVideo_backup.cmake
28 cp ../../OpenCVFindLibsVideo.cmake ../cmake/OpenCVFindLibsVideo.cmake
29 sed -i "s|include_directories( |include_directories("${PREFIX}/include" |g" ...
    ../cmake/OpenCVFindLibsVideo.cmake
30 sed -i "s|REPLACEME|${PREFIX}|g" ../cmake/OpenCVFindLibsVideo.cmake

```

```

31
32 mv ../modules/highgui/src/cap_ffmpeg_impl.hpp ...
   ../modules/highgui/src/cap_ffmpeg_impl_backup.hpp
33 cp ../../cap_ffmpeg_impl.hpp ../modules/highgui/src/cap_ffmpeg_impl.hpp
34 sed -i "s|include_directories( |include_directories("${PREFIX}/include" |g" ...
   android.toolchain.cmake
35
36 cmake -DANDROIDID=1 -DCMAKE_TOOLCHAIN_FILE=android.toolchain.cmake ...
   -DBUILD_SHARED_LIBS=ON -DANDROID_ABI="armeabi-v7a" -DCMAKE_BUILD_TYPE=Release ..
37
38 #sed -i "s|WITH_LIBV4L:B00L=OFF|WITH_LIBV4L:B00L=ON|g" CMakeCache.txt;
39 make -j;
40 make install;
41 cp -r ./install/sdk/native/libs/armeabi-v7a -/lib

```

3.10.4 aarch64-android 平台

```

1 tar -zxvf opencv-2.4.13.tar.gz;
2 cd opencv-2.4.13/;
3 export LD_LIBRARY_PATH=$PREFIX/lib/;
4 export PKG_CONFIG_PATH=$PREFIX/lib/pkgconfig;
5 export PKG_CONFIG_LIBDIR=$PKG_CONFIG_LIBDIR:$PREFIX/lib/;
6 mv CMakeLists.txt CMakeLists_backup.txt;
7 cp ../../CMakeLists.txt ./;
8 mkdir -p build;
9 cd build;
10 mkdir -p install;
11 mkdir -p lib;
12 cp -r $PREFIX/lib/* install/;
13 cp -r $PREFIX/lib/* lib/;
14 cp ../../platforms/android/android.toolchain.cmake ./;
15 sed -i '$a\set(CMAKE_C_COMPILER ...
   /opt/toolchains/android-toolchain-aarch64/bin/aarch64-linux-android-gcc)' ...
   android.toolchain.cmake
16 sed -i '$a\set(CMAKE_CXX_COMPILER ...
   /opt/toolchains/android-toolchain-aarch64/bin/aarch64-linux-android-g++)' ...
   android.toolchain.cmake
17 sed -i '$a\set(CMAKE_FIND_ROOT_PATH "$YOUR_BUILD_PATH/opencv-2.4.13/build")' ...
   android.toolchain.cmake
18 sed -i '$a\link_directories(${PREFIX}/lib)' android.toolchain.cmake
19 sed -i '$a\set(CMAKE_FIND_ROOT_PATH_MODE_PROGRAM NEVER)' android.toolchain.cmake
20 sed -i '$a\set(CMAKE_FIND_ROOT_PATH_MODE_LIBRARY ONLY)' android.toolchain.cmake
21 sed -i '$a\set(CMAKE_FIND_ROOT_PATH_MODE_INCLUDE ONLY)' android.toolchain.cmake
22
23 export ANDROID_STANDALONE_TOOLCHAIN="/opt/toolchains/android-toolchain-aarch64"
24 sed -i 's|OCV_OPTION(WITH_FFMPEG          "Include FFMPEG support" ...
   ON IF (NOT ANDROID AND NOT IOS))|OCV_OPTION(WITH_FFMPEG ...
   "Include FFMPEG support"          ON IF (ANDROID AND NOT ...
   IOS))|g' ../../CMakeLists.txt;
25 mv ../cmake/OpenCVFindLibsVideo.cmake ../cmake/OpenCVFindLibsVideo_backup.cmake
26 cp ../../OpenCVFindLibsVideo.cmake ../cmake/OpenCVFindLibsVideo.cmake
27 sed -i "s|include_directories( |include_directories("${PREFIX}/include" |g" ...
   ../cmake/OpenCVFindLibsVideo.cmake
28 sed -i "s|REPLACEME|${PREFIX}|g" ../cmake/OpenCVFindLibsVideo.cmake
29
30 mv ../modules/highgui/src/cap_ffmpeg_impl.hpp ...
   ../modules/highgui/src/cap_ffmpeg_impl_backup.hpp

```

```
31 cp ../../cap_ffmpeg_impl.hpp ../modules/highgui/src/cap_ffmpeg_impl.hpp
32 sed -i "s|include_directories( |include_directories("${PREFIX}/include" |g" ...
    android.toolchain.cmake
33
34 cmake -DANDROID=1 -DCMAKE_TOOLCHAIN_FILE=android.toolchain.cmake ...
    -DBUILD_SHARED_LIBS=ON -DBUILD_ZLIB=ON -DANDROID_ABI="arm64-v8a" ...
    -DCMAKE_BUILD_TYPE=Release ..
35 make -j;
36 make install;
37 cp -r ./install/sdk/native/libs/arm64-v8a -/lib
```