

在LOCAL SRC FILES中,直接把dlib的源文件作为源文件之一,向之前在dlib移植到arm上说的一样,简单 的不能再简单。

注意头文件路径。

这里有个问题:如果我在上面打开了:

include \$(CLEAR VARS)

#### 会报以下的错误:

- 1 [zzz@localhost jni]\$ ndk-build NDK\_APPLICATION\_MK=Application.mk 2 make: \*\*\* No rule to make target `/home/zzz/software/android-ndk-r9d/build/core/../../face\_landmark\_detection\_test1\_6 3 [zzz@localhost jni]\$
- 2.3 执行
  - 1 zzz@localhost jni]\$ ndk-build NDK\_APPLICATION\_MK=Application.mk 2 [armeabi-v7a] Compile++ thumb: face\_landmark\_detection\_test1\_ex <= face\_landmark\_detection\_test1\_ex.cpp 3 [armeabi-v7a] Compile++ thumb: face\_landmark\_detection\_test1\_ex <= source.cpp 4 [armeabi-v7a] Executable : face landmark detection test1 ex 5 [armeabi-v7a] Install : face\_landmark\_detection\_test1\_ex => libs/armeabi-v7a/face\_landmark\_detection\_test1\_ex 6 [zzz@localhost jni]\$ ndk-build

ß

 $\odot$ 

问题记录:

1,

jni/: error: iostream: No such file or directory

或者其他头文件找不到。

解决办法:

在Application.mk中加入:

APP\_STL:=gnustl\_static 或 gnustl\_shared, c++\_static, c++\_shared

但是用system,stlport\_static stlport\_shared会有头文件找不到。

参考:

http://stackoverflow.com/questions/9130429/android-ndk-build-iostream-no-such-file-or-directory (http://stackoverflow.com/questions/9130429/android-ndk-build-iostream-no-such-file-or-directory)

因为我这个是有jin目录的,所以,不需要再额外指定NDK\_APPLICATION\_MK也可以。



#### 在线课程



http://www.baidu.com/cb.php?c=IgF\_pyfqnHmknjmsnjD0IZ0qnfK9ujYzP1mznWR10Aw-

的的企构工程模型5HR1rjfkn100T1YkPhPWnvDkPyPhuW79Phfv0AwY5HDdnHc3PWn3PjR0IgF\_5y9YIZ0IQzq-

(http://www.baidu.com/cb uzR8mLPbUB48ugiEIAqspynETZ-作品2、事時

<u>ϔϷϯϙϼϷͺϙϏϙϯϥϳϒϻϘϼϜϒϳ;;ͰͰ;ϒͿ;</u>τςαη0ΚdpyfqnHRLPjnvnfKEpyfqnHc4rj6kP0KWpyfqP1cvrHnz0AqLUWYs0ZK45HcsP6KWThnqPjmvnWf)

(http://www.baidu.com/cb.php?c=lgF\_pyfqnHmknjmsnjc0lZ0qnfK9ujYzP1mznWR10Aw-

T0TAq15H

(http://www.baidu.com/cb.php?c=lgF\_pyfqnHmknjmsnjn0lZ0qnfK9ujYzP1mznWR10Aw-YTO TAG15H

uZR8mLPbUB48ugfEIAq

&AMAGUMAN 中報場網達过程记录 (http://bl V5H00T.Regroud HRYTHIJ Article/details/497 RLPjnynfKEpyfqnHc4rj6k 63515 POKWpyfqP1cvrHnz0AqL

WYS0ZK45HcsP6KWT

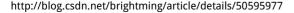
hngn1ckni6) 移植dlib到android (http://blog.csdn.net/bri ghtming/article/details/50595977)

centos7 安装wireshark (http://blog.csdn.n et/brightming/article/details/49174663)

**4394** 

opencv3.0.0编程遇到的问题 (http://blog.c sdn.net/brightming/article/details/504023

**4232** 



ß

ಹ

```
APP STL的取值可以为:
system stlport_static stlport_shared gnustl_static gnustl_shared gabi++_static gabi++_shared c++_static c++_shared
none
2、
/build-binary.mk:393: * target pattern contains no `%'
如果我把Android.mk修改为如下(先编译出dlib的静态库,再与测试程序链接),就会报如下错误:
     1 [zzz@localhost jni]$ ndk-build
     2 [armeabi-v7a] Compile++ thumb: dlib <= source.cpp
     3 [armeabi-v7a] StaticLibrary : dlib.a
     4 make: *** No rule to make target `/home/zzz/software/android-ndk-r9d/build/core/../../face_landmark_detection_test1_€
Android.mk的内容为:
     1 [zzz@localhost jni]$ cat Android.mk
     4 LOCAL_PATH:=$(call my-dir)
     5 LOCAL_MODULE:=dlib
     6 LOCAL_MODULE_FILENAME:=dlib
     7 LOCAL_SRC_FILES:=/home/zzz/software/dlib-18.18/dlib/all/source.cpp
     8 LOCAL_C_INCLUDES:=/home/zzz/software/dlib-18.18/
    11 #LOCAL_EXPORT_C_INCLUDES:=/home/zzz/software/dlib-18.18/
    12 #include $(BUILD_EXECUTABLE)
    13 include $(BUILD_STATIC_LIBRARY)
    14
    16 #include $(CLEAR_VARS)
    17 LOCAL_PATH:=$(call my-dir)
    18 LOCAL_MODULE:=face_landmark_detection_test1_ex
    19 LOCAL_MODULE_FILENAME:=face_landmark_detection_test1_ex
    20 LOCAL_SRC_FILES:=../../face_landmark_detection_test1_ex.cpp # /home/zzz/software/dlib-18.18/dlib/all/source.cpp
    21 LOCAL_C_INCLUDES:=/home/zzz/software/dlib-18.18/
   22
    23
   24 LOCAL_SHARED_LIBRARIES:=dlib
    26 #LOCAL_EXPORT_C_INCLUDES:=/home/gumh/software/dlib-18.18/
    27 #include $(BUILD_EXECUTABLE)
    28 include $(BUILD_STATIC_LIBRARY)
```

#### 原因还不明白。。。

北本 ( つの1 に 1 つの ) ・

ß

 $\Box$ 

 $\odot$ 

ಹ

ቸቦ*ፓ*ር ( 2010-1-30 *)* .

由于上面编译的时候,没有将jpeg,png的库编译进去,因此,在真正运行程序的时候,会报错。重新将相关的代码组织进mk文件进行编译,文件结构没变,各文件的情况如下:

## 1、Application.mk



#### 2, Android.mk

```
1
2 all_cpp_files_recursively = \
3 ${eval src_files = $(wildcard $1/*.cpp)) \
4 ${eval src_files = $(src_files:$(LOCAL_PATH)/%=%))$(src_files) \
```

```
5 | $(eval item_all = $(wildcard $1/*)) \
                              6 $(foreach item, $(item_all) $(),\
                              7 $(eval item := $(item:%.cpp=%)) \
ß
                              8 $(call all_cpp_files_recursively, $(item))\
                             10
\Box
                             12 #3 自定义了一个all_c_files_recursively 函数,递归遍历返回给定目录下所有C源文件。
\odot
                             13 all_c_files_recursively = \
                             14 $(eval src_files = $(wildcard $1/*.c)) \
ಹ
                             15 $(eval src_files = $(src_files:$(LOCAL_PATH)/%=%))$(src_files) \
                             16  $(eval item_all = $(wildcard $1/*)) \
                             17 $(foreach item, $(item_all) $(),\
                                 $(eval item := $(item:%.c=%)) \
                             19 $(call all_c_files_recursively, $(item))\
                            20
                            21
                            22
                             23 LOCAL_PATH:=$(call my-dir)
                            24 include $(CLEAR_VARS)
                            26 LOCAL_SRC_FILES := $(call all_cpp_files_recursively,$(LOCAL_PATH)/../.../allib/external/libjpeg/) #将jpeg相关代码加入
                            27 LOCAL_SRC_FILES += $(LOCAL_PATH)/../../dlib/all/source.cpp
                             28 LOCAL_C_INCLUDES:=/home/gumh/software/dlib-18.18/ $(LOCAL_PATH)/../../dlib/entropy_decoder/
                             29 LOCAL_CPP_INCLUDES:=${LOCAL_C_INCLUDES}
                             30 LOCAL_CXX_INCLUDES:=${LOCAL_C_INCLUDES}
                            31
                            32 $(warning "-----the value of LOCAL_SRC_FILES is $(LOCAL_SRC_FILES)------
                            34 ##neon start
                             35 TARGET_ARCH_ABI :=armeabi-v7a
                             36 LOCAL_ARM_MODE := arm
                            37 LOCAL_ARM_NEON := true
                             38
                             39 ifeq ($(TARGET_ARCH_ABI),armeabi-v7a)
                                  LOCAL_CFLAGS := -DHAVE_NEON=1
                            41
                                  LOCAL_CXXFLAGS := -DHAVE_NEON=1
                             42 endif
                             43 LOCAL_CFLAGS += -mfloat-abi=softfp -mfpu=vfpv4
                            44 ###neon end
                             45
                             46
                            47 ##放在一起编译
                            48 LOCAL_MODULE:=face_landmark_detection_test1_ex
ß
                            49 LOCAL_MODULE_FILENAME:=face_landmark_detection_test1_ex
                            50 LOCAL_SRC_FILES+=../../face_landmark_detection_test1_ex.cpp
                            51
\Box
                            52 include $(BUILD_EXECUTABLE)
\odot
                        从上面的文件可以看到,其实就是配合source.cpp定义的宏,把相关的jpeg的源代码加入一起编译。
```

是否可以先编译出dlib的静态库,给后面的应用程序编译使用呢?

ಹ

新加一个Android-prebuild.mk:

```
2
                             3 all_cpp_files_recursively = \
                             4 $(eval src_files = $(wildcard $1/*.cpp)) \
                             5 $(eval src_files = $(src_files:$(LOCAL_PATH)/%=%))$(src_files) \
                             6 $(eval item_all = $(wildcard $1/*)) \
                             7 $(foreach item, $(item_all) $(),\
ß
                             8 $(eval item := $(item:%.cpp=%)) \
                             9 $(call all_cpp_files_recursively, $(item))\
                            10 )
\Box
                            11
                            12
\overline{\odot}
                            13 #3 自定义了一个all_c_files_recursively 函数,递归遍历返回给定目录下所有C源文件。
                            14 all_c_files_recursively = \
                            16 $(eval src_files = $(src_files:$(LOCAL_PATH)/%=%))$(src_files) \
                            17  $(eval item_all = $(wildcard $1/*)) \
                            18 $(foreach item, $(item_all) $(),\
                            19 $(eval item := $(item:%.c=%)) \
                            20 $(call all_c_files_recursively, $(item))\
                            21 )
                            22
                            23
                            24
                            25
                            26 LOCAL_PATH:=$(call my-dir)
                            27
```

```
1 □ □ □
```

```
$(warning "-----")
31
      include $(CLEAR VARS)
     LOCAL_MODULE:=dlib
     LOCAL_SRC_FILES := $(call all_cpp_files_recursively,$(LOCAL_PATH)/../../dlib/external/libjpeg/)
33
34
      LOCAL_SRC_FILES += $(LOCAL_PATH)/../../dlib/all/source.cpp
     LOCAL_C_INCLUDES:=/home/gumh/software/dlib-18.18/ $(LOCAL_PATH)/../../dlib/entropy_decoder/
35
      LOCAL_CPP_INCLUDES:=${LOCAL_C_INCLUDES}
36
     LOCAL_CXX_INCLUDES:=${LOCAL_C_INCLUDES}
37
38
39
      $(warning "------the value of LOCAL_SRC_FILES is $(LOCAL_SRC_FILES)-----")
40
41
      ##neon start
42
      TARGET_ARCH_ABI :=armeabi-v7a
43
      LOCAL_ARM_MODE := arm
44
      LOCAL_ARM_NEON := true
45
46
      ifeq ($(TARGET_ARCH_ABI),armeabi-v7a)
47
       LOCAL_CFLAGS := -DHAVE_NEON=1
48
       LOCAL_CXXFLAGS := -DHAVE_NEON=1
49
50
51
      LOCAL_CFLAGS += -mfloat-abi=softfp -mfpu=vfpv4
52
53
      #######海思的设置: set(CMAKE_CXX_FLAGS "-mthumb -mcpu=cortex-a7 -mfloat-abi=softfp -mfpu=neon-vfpv=
54
55
      ###neon end
56
57
      include $(BUILD_STATIC_LIBRARY)
58
59
    #######2####################
60 #######build app############
61
    $(warning "------build app-----")
62
63 #use prebuild dlib
64 LOCAL_MODULE:=mydlib
65 LOCAL_SRC_FILES:=libdlib.a
66 LOCAL_EXPORT_C_INCLUDES:=/home/gumh/software/dlib-18.18/ $(LOCAL_PATH)/../../dlib/entropy_decoder/
67 include $(PREBUILT_STATIC_LIBRARY)
68
69 ###app######
70 include $(CLEAR_VARS)
71 LOCAL_MODULE:=face_landmark_detection_test1_ex
72 LOCAL_SRC_FILES:=../../face_landmark_detection_test1_ex.cpp
73 LOCAL_STATIC_LIBRARIES:=mydlib
74 include $(BUILD_EXECUTABLE)
75
76 #############
```

在把Application.mk里的APP\_BUILD\_SCRIPT改为:Android-prebuilt.mk

APP\_BUILD\_SCRIPT:=Android-prebuilt.mk



# 相关文章推荐

# dlib-android编译 (http://blog.csdn.net/forest\_world/article/details/78147484)

ß

https://github.com/tzutalin/dlib-android androidstudio-examples Refactor folder structions 1...

	forest_world (http://blog.csdn.net/forest_world) 2017年10月01日 14:57 □343
<b>⊕</b>	算法移植优化(三)android dlib 人脸检测使用 (http://blog.csdn.net/hjimce/article/details/6
æ	1、在cmakelist.txt中加入:include(src/main/cpp/dlib/cmake)然后再链接dlib库就可以了:target_link_libraries( # Specifie
	1 hjimce (http://blog.csdn.net/hjimce) 2017年03月20日 16:38 日3452
	AI 专业人才缺口上百万,年薪 80 万远超同行?? 就目前来看,国内 AI 人才缺乏且经验不足,为争抢优秀人才,企业背后的暗战早已打响。作为正在谋 求一份好工作我,又该如何抉择
	(http://www.baidu.com/cb.php?c=lgF_pyfqnHmknjnvPjn0lZ0qnfK9ujYzP1ndPWb10Aw-5Hc3rHnYnHb0TAq15HfLPWRznjb0T1YzPWuBnWlBmyc3uAm4PAfk0AwY5HDdnHc3PWn3Pjf0lgF_5y9YlZ0lQzq-uZR8mLPbUB48ugfElAqspynEmybz5LNYUNq1ULNzmvRqmhkEu1Ds0ZFb5HD0mhYqn0KsTWYs0ZNGujYkPHTYn1mk0AqGujYknWb3rjDY0APGujYLnWm4n1c0ULl85H00TZbqnW0v0APzm1YznWDknf)
	dlib 编译安装(包含android版本) (http://blog.csdn.net/haima1998/article/details/77938313)
	参考 http://dlib.net/compile.html 1.下载源码: http://dlib.net/files/dlib-19.6.zip 2. 编译examples cd
	A haima1998 (http://blog.csdn.net/haima1998) 2017年09月11日 21:44 □ □ 270
	Delphi7高级应用开发随书源码 (http://download.csdn.net/download/chenx  /http://download / 2003年04月30日 00:00 676KB 下载
	Android studio 2.3.3+Dlib (http://blog.csdn.net/qi_w_ip/article/details/76286216)
	最近因为需要,尝试着将Dlib移植到android上。android小白一枚,做之前没接触过android,所以就用了别人说比较好的andr oid studio 最新稳定版(windows下)。在an
	(r. ★ qi_w_ip (http://blog.csdn.net/qi_w_ip) 2017年07月29日 08:57 □ 760
	AI 工程师职业指南
<u>6</u>	我们请来商汤、杜邦、声智、希为、58同城、爱因互动、中科视拓、鲁朗软件等公司 AI 技术一线的专家,请他们从实践的角度来解析 AI 领域各技术岗位的合格工程师都是怎样炼成的。
	(http://www.baidu.com/cb.php?c=IgF_pyfqnHmknjfzrj00IZ0qnfK9ujYzP1f4Pjnd0Aw-
$\odot$	5Hc4nj6vPjm0TAq15Hf4rjn1n1b0T1YYujI- rynYryPhrAuWmynd0AwY5HDdnHc3PWn3Pjf0lgF_5y9YIZ0lQzqMpgwBUvqoQhP8QvIGIAPCmgfEmvq_Iyd8Q1R4uhF- rA7Wuj0YmhP9PARvujmYmH0vm1qdIAdxTvqdThP-
ଝ	5HDknWF9mhkEusKzujYk0AFV5H00TZcqn0KdpyfqnHRLPjnvnfKEpyfqnHnsnj0YnsKWpyfqP1cvrHnz0AqLUWYs0ZK45HcsP6KWThnqnWnknf)
	【机器学习】Dlib——机器学习库学习系列 (http://blog.csdn.net/Taily_Duan/article/details/5

http://blog.csdn.net/brightming/article/details/50595977

测)最近在做...

Dlib机器学习库学习系列1----编译安装 Dlib机器学习库系列2----人脸检测 Dlib机器学习库学习系列三----人脸对齐(特征点检



## dlib交叉编译到arm (http://blog.csdn.net/brightming/article/details/50582222)

dlib交叉编译到arm使用dlib是很方便的,开发者可以将其视为一个工具来使用,看其说明:High Quality Portable Code Good unit test coverage. Th...

brightming (http://blog.csdn.net/brightming) 2016年01月25日 17:54 

□2588

# 移植opencv2.4.9到android过程记录 (http://blog.csdn.net/brightming/article/details/50606...

在移植到arm开发板的时候已经说过,opencv已经为各平台准备了一套cmake交叉编译脚本。为android移植的脚本在opencv 目录/platforms/android/android.tool...

#### dlib的编译 (http://blog.csdn.net/liukang325/article/details/55211175)

github地址: https://github.com/davisking/dlib 也可以在此下载: http://dlib.net/我解压至此: D:/OpenCV/dlib-19.2 cmake...

liukang325 (http://blog.csdn.net/liukang325) 2017年02月15日 16:14 単1489



#### Delphi7高级应用开发随书源码 (http://download.csdn.net/download/chenx...

/http://download: 2003年04月30日 00:00 676KB 下载

## dlib编译出错以及解决方法 (http://blog.csdn.net/wangguanghui0607/article/details/523463...

想在C++代码里用机器学习的方法,就找到dlib这个C++的机器学习的library。按照下边官网的方法编译例子出现了关于X11的 error。 cd examples mkdir build cd b...

🌍 wangguanghui0607 (http://blog.csdn.net/wangguanghui0607) 2016年08月28日 16:20 🕮 1036

## Android实战技巧之十二: Android Studio导入第三方类库、jar包和so库 (http://blog.csdn.ne...

第三方类库源码将一网友的XMPP代码从ADT转到AS时,发现其使用了第三方类库,源码放在了lib下,直接在AS中Import pr oject,第三方类库并没有自动导入进来,看来需要自己动手了。项目的目...