Wicab Sign Finder -- Installation instruction

Setting up and compiling

Cloning the repository

The Sign Finder repository can be cloned from:
https://YOURUSERNAME@bitbucket.org/skericompvis/wicab_signfinder.git

Installing CMake

In order to compile the source code you'll need to download and install CMake (http://www.cmake.org/download/)

Dependencies

In order to compile the software, the following dependencies need to be satisfied:

- OpenCV 2.4.11
- LibSVM 3.20

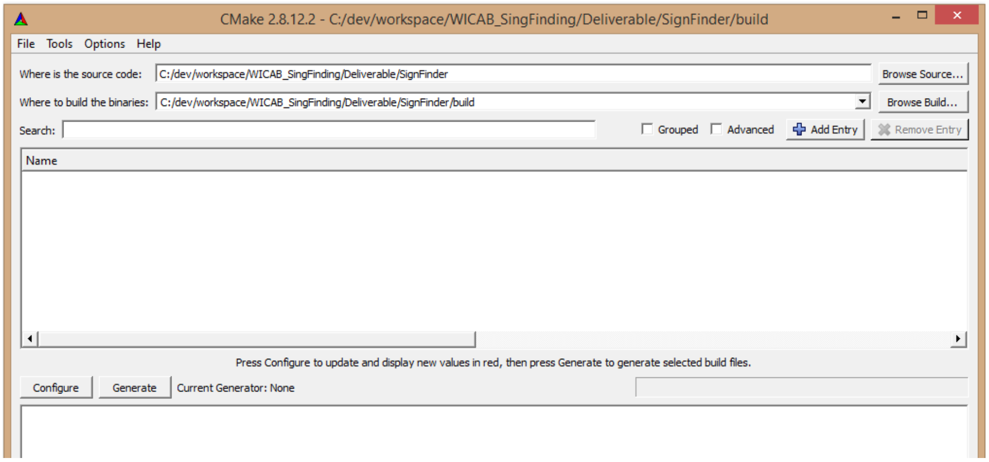
Install and compile OpenCV 2.4.11 (http://opencv.org/downloads.html). After the installation, make sure that the directory containing the OpenCV libraries is included in the system path, otherwise a "library not found" error may happen at runtime.

Here you can find instruction on how to compile and install OpenCV: http://docs.opencv.org/doc/tutorials/introduction/table_of_content_introduction/table_of_content_introduction.html.

Download LibSVM 3.20 and extract the archive in your workspace. No installation or compilation are required for this library.

Configuring the Sign Finder project

Launch CMake-gui. In the box “**Where is the source code**” put the location of your SignFinder working copy. In the box “**Where to build the binaries**” you can specify any folder. This is where the executable will appear after the compilation.



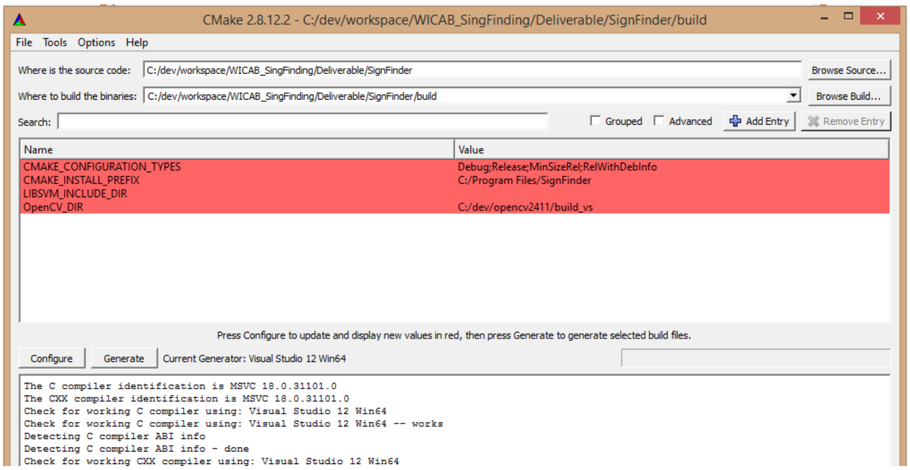
Where is the source code: C:/dev/workspace/WICAB_SingFinding/Deliverable/SignFinder
Where to build the binaries: C:/dev/workspace/WICAB_SingFinding/Deliverable/SignFinder/build
Search:

Name

Press Configure to update and display new values in red, then press Generate to generate selected build files.

Current Generator: None

After setting up the folders, press the **Configure** button and when prompted choose the compiler you will be using to compile the code. If you set up everything correctly, this should appear:



Where is the source code: C:/dev/workspace/WICAB_SingFinding/Deliverable/SignFinder
Where to build the binaries: C:/dev/workspace/WICAB_SingFinding/Deliverable/SignFinder/build
Search:

Name	Value
CMAKE_CONFIGURATION_TYPES	Debug;Release;MinSizeRel;RelWithDebInfo
CMAKE_INSTALL_PREFIX	C:/Program Files/SignFinder
LIBSVM_INCLUDE_DIR	
OpenCV_DIR	C:/dev/opencv2411/build_vs

Press Configure to update and display new values in red, then press Generate to generate selected build files.

Current Generator: Visual Studio 12 Win64

```
The C compiler identification is MSVC 18.0.31101.0
The CXX compiler identification is MSVC 18.0.31101.0
Check for working C compiler using: Visual Studio 12 Win64
Check for working C compiler using: Visual Studio 12 Win64 -- works
Detecting C compiler ABI info
Detecting C compiler ABI info - done
Check for working CXX compiler using: Visual Studio 12 Win64
```

In LIBSVM_INCLUDE_DIR field, specify the location of LibSVM source code (the folder where svm.h and svm.cpp are). In the OpenCV_DIR field, specify the folder in which you put the compiled OpenCV libraries.

Press **Configure** again to update the configuration and then **Generate** to generate the makefile.

Now you’re ready to compile SignFinder using the compiler you selected earlier.

Sign Finder - Configuration File

The SignFinder uses an external YAML file to specify the parameters of the object detector. We provided a configuration file for the detection of exit signs. The current configuration gives a good compromise between recall and computational time.

```
%YAML:1.0


# ATTENTION: DO NOT USE TAB CHARACTERS WHEN EDITING THIS FILE (you can replace tabs with
spaces in you editor settings)


# Classifiers location

CascadeFile: "res/cascade.xml"     # full path to cascade classifier CascadeFile

SVMFile: "res/exitModel.svm"        # full path to SVM model


# Cascade search window parameters. It can be changed, but it is a good idea to keep the same
aspect ratio (w/h = 1.5)

minWinSize:

    width: 18

    height: 12

CascadeScaleFactor: 1.1    # scale factor for multiscale detection


maxWinSizeFactor: 8.       #specify largest search window as a multiple of minWinSize


# HOG

HOG_winSize:               # DO NOT CHANGE: this parameter should be changed only if the SVM
is trained with a different window size

    width: 36

    height: 24

SVMThreshold: .85          # sets the SVM confidence threshold

# Preprocessing

CroppingFactors:           # specify which section of the image to process. Cropping origin is
(0,0) i.e. top left corner

    width: 1.0

    height: 0.66

ScaleFactor: 0.66          # specify rescaling factor for cropped frames (helps with detection
of small objects)

Flip: 1                    # flip input frames  (used for portrait vs landscape videos)

Transpose: 1               # transpose input frames (used for portrait vs landscape videos)


# Debug options

ShowIntermediate: 0
```

Running the SignFinder

SignFinder is a console application and it take two parameters in input: the full path to a configuration file and the full path to a video.

To launch it, open a shell interface in the folder where the binary is and use the following syntax:

SignFinder *full_path_to_config_file full_path_to_video*

While the software is running, as the video plays, a window will show the detection results over each single frame in the video sequence in the form of one or more rectangles that highlight patches of the image in which the detection of an exit sign has occurred.