**APPENDIX A**

Expanded Block Diagram of the Face Detection System



**APPENDIX B**

**SOCKIT DEVELOPMENT KITS & TOOLS DETAIL**

|  |  |
| --- | --- |
| Description | The SoCKit Development Kit presents a robust hardware design platform built around the Altera System-on-Chip (SoC) FPGA, which combines the latest dual-core Cortex-A9 embedded cores with industry-leading programmable logic for ultimate design flexibility. Users can now leverage the power of tremendous re-configurability paired with a high-performance, low-power processor system. Altera’s SoC integrates an ARM-based hard processor system (HPS) consisting of processor, peripherals and memory interfaces tied seamlessly with the FPGA fabric using a high-bandwidth interconnect backbone. The SoCKit development board includes hardware such as high-speed DDR3 memory, video and audio capabilities, Ethernet networking, and much more. In addition, an on-board HSMC connector with high-speed transceivers allows for an even greater array of hardware setups. By leveraging all of these capabilities, the SoCKit is the perfect solution for showcasing, evaluating, and prototyping the true potential of the Altera SoC. |
| Features | FPGA Device • Cyclone V SoC 5CSXFC6D6F31 Device • Dual-core ARM Cortex-A9 (HPS) • 110K Programmable Logic Elements • 5,140 Kbits embedded memory • 6 Fractional PLLs • 2 Hard Memory Controllers • 3.125G Transceivers  Configuration and Debug • Quad Serial Configuration device – EPCQ256 on FPGA • On-Board USB Blaster II (micro USB type B connector)  Memory Device • 1GB (2x256MBx16) DDR3 SDRAM on FPGA • 1GB (2x256MBx16) DDR3 SDRAM on HPS • 128MB QSPI Flash on HPS • Micro SD Card Socket on HPS  Communication • USB 2.0 OTG (ULPI interface with micro USB type AB connector) • USB to UART (micro USB type B connector) • 10/100/1000 Ethernet  Connectors • One HSMC (8-channel Transceivers, Configurable I/O standards 1.5/1.8/2.5/3.3V) • One LTC connector (One Serial Peripheral Interface (SPI) Master ,one I2C and one GPIO interface )  Display • 24-bit VGA DAC • 128x64 dots LCD Module with Backlight  Audio • 24-bit CODEC, Line-in, line-out, and microphone-in jacks  Switches, Buttons and LEDs • 8 User Keys (FPGA x4 ; HPS x 4) • 8 User Switches (FPGA x4 ; HPS x 4) • 8 User LEDs (FPGA x4 ; HPS x 4) • 2 HPS Reset Buttons (HPS\_RSET\_n and HPS\_WARM\_RST\_n)  Sensors • G-Sensor on HPS • Temperature Sensor on FPGA  Power • 12V DC input |
| Kit Includes | • The SoCKit development board • USB Cable for FPGA programming and control • Ethernet Cable • 12V DC power adapter |

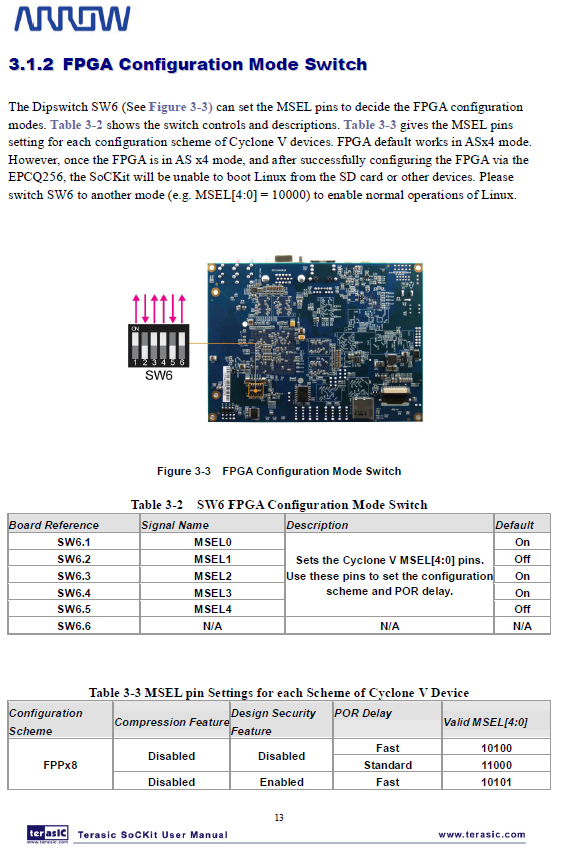
**APPENDIX C**

ARM® Cortex™-A9 MPCore™ Specification

|  |
| --- |
| * 800 MHz dual-core processor supporting symmetric and asymmetric multiprocessing * Each processor includes the following:   + High-efficiency, dual-issue superscalar pipeline (2.5 MIPS\* per MHz)   + NEONTM media processing engine for media and signal processing acceleration   + Single- and double-precision floating-point unit   + 32 KB instruction and 32 KB data caches   + Cache coherence for enhanced inter-processor communication   + Memory Management Unit with TrustZone® security technology   + Thumb®-2 technology for enhanced code density, performance, and power efficiency   + Jazelle® architecture extensions for accelerating Java Virtual Machine   + Program Trace Macrocell for full visibility of processor instruction flow * Shared 512 KB, 8-way associative L2 cache, lockable by way, line, or master * Acceleration coherency port that extends coherent memory access beyond the CPUs * Generic interrupt controller * 32 bit general purpose timer * Watchdog timer * Available in Altera® Arria® V SoCs and Cyclone® V SoCs |

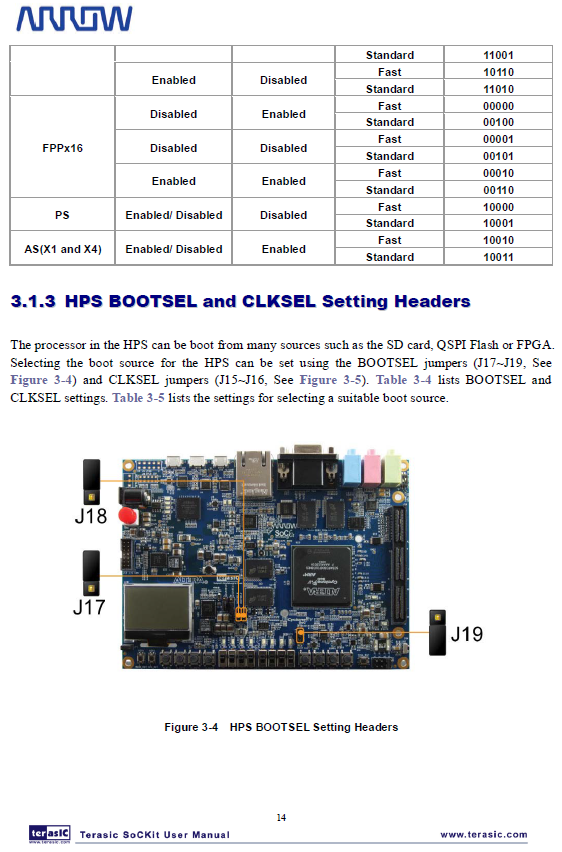
**APPENDIX D**

**FPGA Configuration Mode (Taken from SoCKit User Manual)**



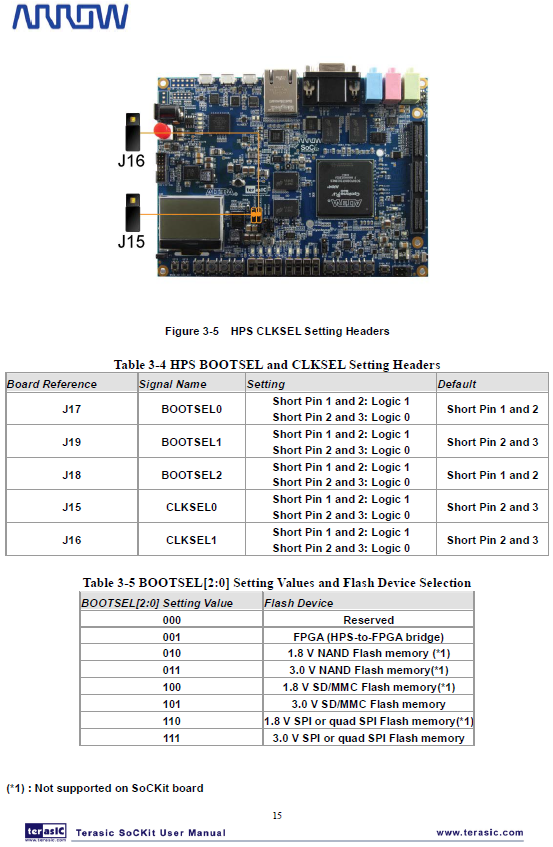
**APPENDIX D (CONTINUED)**

**FPGA Configuration Mode (Taken from SoCKit User Manual)**



**APPENDIX D (CONTINUED)**

**FPGA Configuration Mode (Taken from SoCKit User Manual)**



**APPENDIX E**

**Configuring U-Boot to the Proper Locations of Kernel Program**

bootcmd=run mmcload; run fpgaload; run mmcboot

fpga=0

fpgadata=0x2000000

fpgadatasize=2126D5

fpgaload=fatload mmc 0:1 0x2000000 soc\_system.rbf; fpga load 0 ${fpgadata} ${fi}

**Configuring the Serial Console to Port 57600 for Debugging**

setenv mmcboot 'setenv bootargs console=tty0 console=ttyS0,57600 root=${mmcroot} rw rootwait;bootm ${loadaddr} - ${fdtaddr}'

**APPENDIX F**

**From RocketBoards Org On How to Partition a uSD Card.**

Partition the SD Card

1. Determine the device associated with the SD card on the host by running the following command before and after inserting the card in the reader:

$ cat /proc/partitions

Let's assume it is /dev/sdx

1. Repartition the card, with the sudo fdisk/dev/sdx command

$ Use p to print current partitioning

$ Use d to delete all partitions one by one

Create partition using these options, pressing enter after each one: n p 2 4096 +4496384 t 83  
Create partition using these options, pressing enter after each one: n p 1 9000000 +20480K t 1 b  
Create partition using these options, pressing enter after each one: n p 3 2048 +1024K t 3 a2

1. Press w to write partitions to card, then exit

Verify the card was partitioned correctly, it should look like this (use fdisk and p to see it):   
Device Boot Start End Blocks Id System   
/dev/sdx1 9000000 9020480 20480 b W95 FAT32   
/dev/sdx2 4096 8996863 4496384 83 Linux   
/dev/sdx3 2048 4095 1024 a2 Unknown

1. Make partition one a DOS partition

$ sudo mkdosfs /dev/sdx1

**APPENDIX G**

**Specs of the Intel CISC based system**

|  |
| --- |
| Host Name: ANDROMEDA  OS Name: Microsoft Windows 7 Ultimate  OS Version: 6.1.7601 Service Pack 1 Build 7601  OS Manufacturer: Microsoft Corporation  OS Configuration: Standalone Workstation  OS Build Type: Multiprocessor Free  Registered Owner: Bangonkali  Registered Organization:  Product ID: 00426-OEM-8992662-00006  Original Install Date: 11/30/2013, 1:30:10 AM  System Boot Time: 3/13/2014, 9:30:26 PM  System Manufacturer: Acer  System Model: Aspire 4755  System Type: x64-based PC  Processor(s): 1 Processor(s) Installed.  [01]: Intel64 Family 6 Model 42 Stepping 7 GenuineInt  el ~792 Mhz  BIOS Version: Phoenix Technologies Ltd. V2.13, 10/21/2011  Windows Directory: C:\Windows  System Directory: C:\Windows\system32  Boot Device: \Device\HarddiskVolume1  System Locale: en-us;English (United States)  Input Locale: en-us;English (United States)  Time Zone: (UTC+08:00) Taipei  Total Physical Memory: 8,003 MB  Available Physical Memory: 4,280 MB  Virtual Memory: Max Size: 16,005 MB  Virtual Memory: Available: 12,066 MB  Virtual Memory: In Use: 3,939 MB  Page File Location(s): C:\pagefile.sys  Domain: WORKGROUP  Logon Server: \\ANDROMEDA |

**APPENDIX H**

**Kernel Panic Logs on Continue Long Term Operation of Face Detection System**

|  |
| --- |
| Unable to handle kernel NULL pointer dereference at virtual address 0000003c  pgd = 80004000  [0000003c] \*pgd=00000000  Internal error: Oops: 17 [#1] SMP ARM  Modules linked in: ipv6 uvcvideo videobuf2\_core videodev videobuf2\_vmalloc video buf2\_memops gpio\_dw  CPU: 0 Not tainted (3.8.0-00111-g85cc90f #1)  PC is at usb\_hcd\_unmap\_urb\_setup\_for\_dma+0x8/0xb0  LR is at usb\_hcd\_unmap\_urb\_for\_dma+0x14/0x134  pc : [<8029b110>] lr : [<8029b1cc>] psr: a0000193  sp : bed8bc88 ip : 00000000 fp : 00000001  r10: be11b0d0 r9 : 80559a34 r8 : 00000000  r7 : 00000018 r6 : 000000a0 r5 : be11b000 r4 : 00000000  r3 : bc194b08 r2 : 800400c7 r1 : 00000000 r0 : be11b000  Flags: NzCv IRQs off FIQs on Mode SVC\_32 ISA ARM Segment user  Control: 10c5387d Table: 3c01c04a DAC: 00000015  Process facedetect (pid: 1646, stack limit = 0xbed8a240)  Stack: (0xbed8bc88 to 0xbed8c000)  bc80: 00000000 be11b000 000000a0 8029b1cc be11b0fc be11b104  bca0: be010a10 bebf2280 0000000c 000000a0 00000018 802b2738 bed8bd14 00001000  bcc0: 00000081 80546800 00000001 00001000 805467e8 8024db6c 0000004c 00000033  bce0: 20b48bfc bebf2280 c0d805e0 802afd08 8001c960 00000001 000003fc 00000040  bd00: 00000000 bebf2280 00000000 00000000 0000b071 60000193 be010a10 000000a0  bd20: 00000000 00000000 80559a34 80559a20 00000000 802af9f8 802af9ec 8029ab1c  bd40: bebf1100 80075bf0 00000001 bebf2ec0 00000002 be0109c0 bed8bdb4 be0109c0  bd60: be010a10 00000000 fee00100 be11b0d0 00000000 fffffffe a0000013 80075e24  bd80: be0109c0 be010a10 00000000 80078bac 80078b30 000000a0 000000a0 800755d4  bda0: 80525030 8000f014 fee0010c 8052e418 bed8bdd0 80008530 802b070c 803b6a88  bdc0: 60000013 ffffffff bed8be04 8000dd40 be11b0d0 a0000013 00000000 8a258a25  bde0: bedb4e00 be5f7000 bc194b00 be11b000 be11b0d0 00000000 fffffffe a0000013  be00: 0000003d bed8be18 802b070c 803b6a88 60000013 ffffffff 802b0644 fffffffe  be20: be5f7000 be11b000 fffffffe be419000 be31b788 00000001 bed8a000 8029b3f8  be40: be1c0400 fffffffe be5f7000 60000013 00000000 be419000 00000001 8029c708  be60: be5f7000 be5f700c 00000001 8029d590 bed8beac be41900c be5f6000 00000002  be80: 00000000 be419000 be31b788 be419010 be5f7000 7f041a68 00000001 be419000  bea0: 00000000 00000000 be419000 00000008 00000000 7f043e54 00000001 be43bd80  bec0: be31b780 bd11ea18 be419000 7f03ff78 7f03fee0 be23fc00 be664e88 bd11ea18  bee0: be05b310 7f014390 7f014358 be31b780 be664e88 800db0d8 00000000 00000000  bf00: be2d283c beba66c0 805618c4 be2d2580 8000e2a8 bed8a000 00000000 8003e488  bf20: be1e68c0 fd87fd87 be2d2580 be1e68c0 be2d284c be2d2580 be1e68fc 800265c8  bf40: be1e68c0 00000000 bed8a000 00000001 00382000 800c4960 fd87fd87 800da1ac  bf60: be1e6900 be57f680 00000000 bed8a000 000000f8 8000e2a8 bed8a000 00000000  bf80: 00000000 80026d88 00000000 000703c2 7698e760 7698e760 000000f8 80026e2c  bfa0: 00000000 8000e100 000703c2 7698e760 00000000 000703ae 74c214c0 00000000  bfc0: 000703c2 7698e760 7698e760 000000f8 00000000 00000000 76fa1000 00000000  bfe0: 000000f8 7eab82bc 7691fce3 768c71e6 600f0030 00000000 00000000 00000000  [<8029b110>] (usb\_hcd\_unmap\_urb\_setup\_for\_dma+0x8/0xb0) from [<be11b104>] (0xbe1 1b104)  Code: 8055994b 80497e64 e92d4070 e1a04001 (e591303c)  ---[ end trace 7240ef6d486fb03d ]--- |

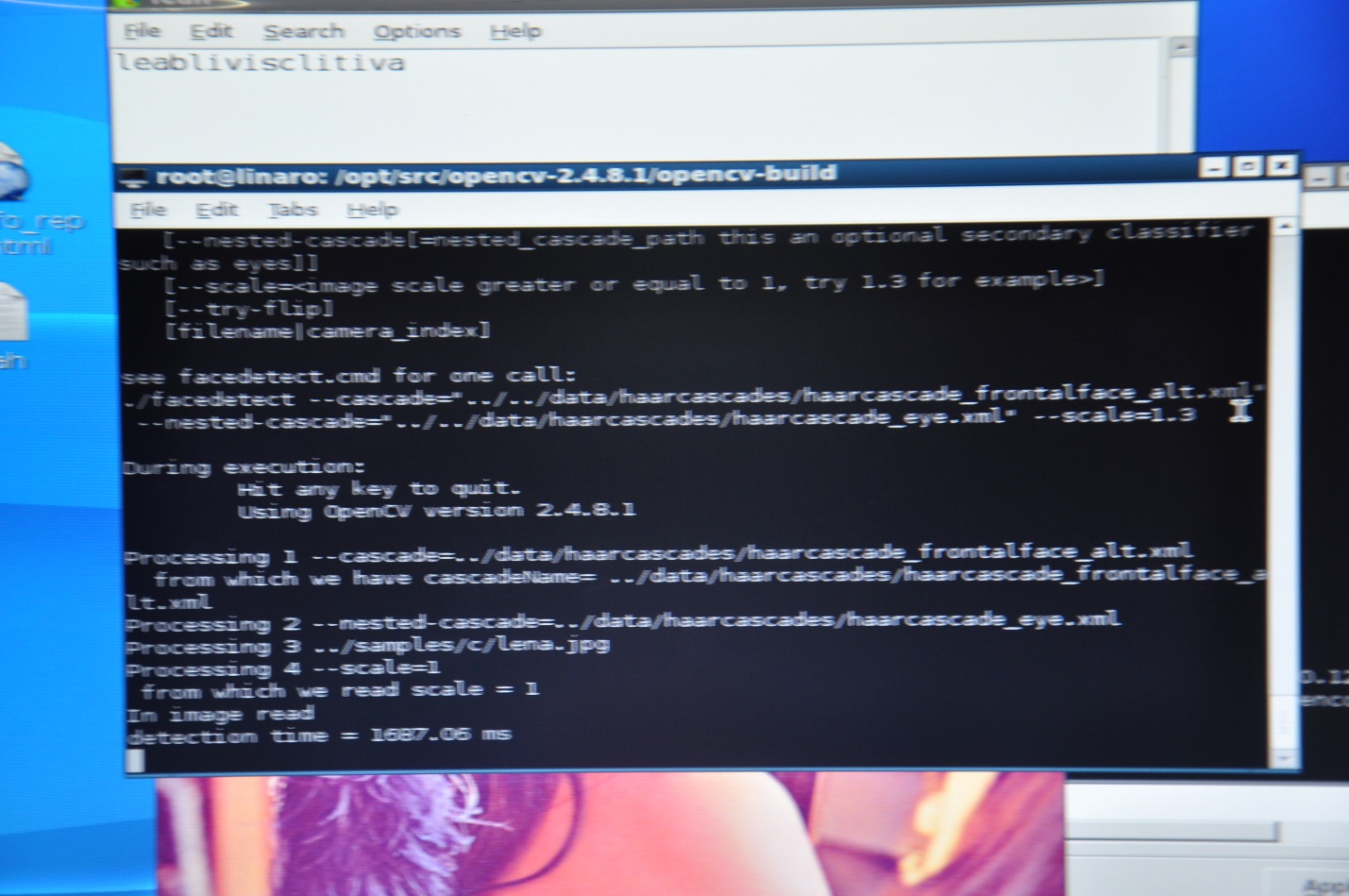
**APPENDIX I**

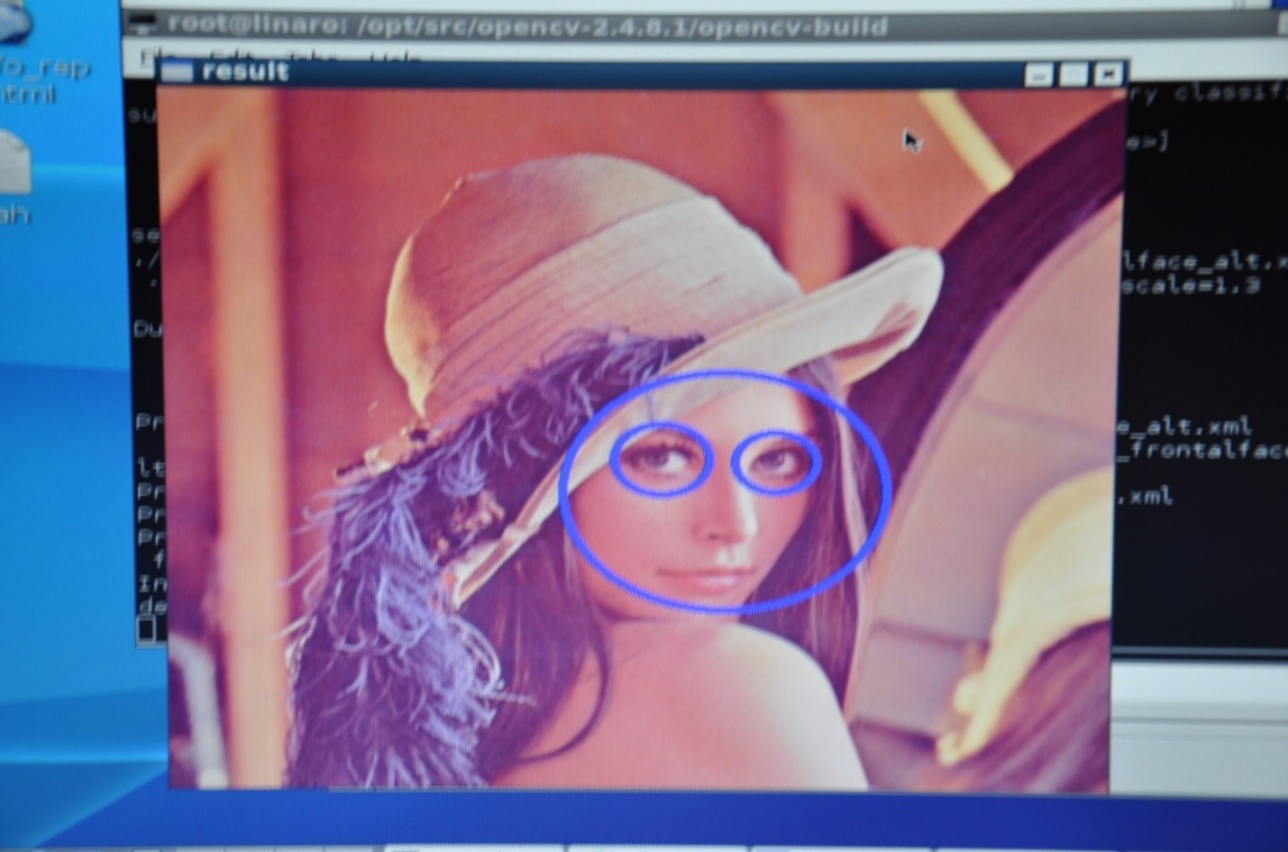
Researcher’s Notes on SoCKit Usage

|  |
| --- |
| Sockit usage notes:  As root:  sudo chmod u+s /usr/bin/sudo  su linaro  sudo apt-get update && \  sudo apt-get upgrade && \  sudo ntpdate time.nist.gov && \  sudo apt-get install wget build-essential checkinstall libncurses5-dev libncurses5 libevent-1.4 libevent-dev libncurses5-dev pkg-config libpng12-0 libpng12-dev libpng++-dev libpng3 libpnglite-dev zlib1g-dbg zlib1g zlib1g-dev pngtools libtiff4-dev libtiff4 libtiffxx0c2 libtiff-tools ffmpeg v4l-conf v4l-utils v4l2ucp libjpeg-dev libpng-dev libtiff-dev libjasper-dev libavcodec-dev libavformat-dev libswscale-dev usbutils gparted vim less psmisc lynx gcc g++ xserver-xorg xterm firefox mplayer lxde xinit x11-apps xinput-calibrator mesa-utils prboom  Check webcam  v4l2-ctl --list-formats  v4l2-ctl --list-formats-ext  v4l2-ctl --set-fmt-video=width=1280,height=720,pixelformat=1  v4l2-ctl --set-fmt-video=width=640,height=480,pixelformat=1  v4l2-ctl --set-parm=30  lsusb  1. Find & install CMAKE  download cmake  http://www.cmake.org/cmake/resources/software.html  tar xf cmake\*  cd cmake\*  ./bootstrap && make && sudo make install  Installing Synergy  sudo apt-get install xorg-dev libqt4-dev python  libcurl4-openssl-dev [cant be detected]  python-dev make gcc  http://synergy-foss.org/wiki/Compiling  ./hm.sh conf -g1 --make-gui  ./hm.sh build --make-gui  HTOP  svn co https://htop.svn.sourceforge.net/svnroot/htop/trunk htop  --disable-unicode  Installing OpenSSH  http://www.unixwiz.net/techtips/openssh.html#tarballs  ./configure --with-ipv4-default --with-md5-passwords --with-pam && make && make install  ssdh privilege escalation  groupadd sshd  useradd -M -g sshd -c 'sshd privsep' -d /var/empty -s /sbin/nologin sshd  passwd -l sshd  useradd -g root linaro  building cmake  http://www.cmake.org/cmake/help/install.html  http://docs.opencv.org/doc/tutorials/introduction/linux\_install/linux\_install.html  sudo apt-get install ffmpeg && \  sudo apt-get install v4l-conf v4l-utils v4l2ucp && \  sudo apt-get install libjpeg-dev libpng-dev libtiff-dev libjasper-dev && \  sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev  time  sudo apt-get install ntp  Probably the ntp service is running, that's why ntpdate can't open the socket (port 123 UDP) and connect to ntp server.  Try from command line:  sudo service ntp stop  sudo ntpdate -s time.nist.gov  sudo service ntp start  If you want to put this in /etc/rc.local use the following:  ( /etc/init.d/ntp stop  until ping -nq -c3 8.8.8.8; do  echo "Waiting for network..."  done  ntpdate -s time.nist.gov  /etc/init.d/ntp start )&  http://askubuntu.com/questions/254826/how-to-force-a-clock-update-using-ntp  <https://help.ubuntu.com/10.04/serverguide/NTP.html>  chmod -R 777 /tmp  chmod -R 777 /usr/local/src  chown -R root:root /usr/bin/sudo  chmod 4755 /usr/bin/sudo  apt-get update  apt-get dist-upgrade  apt-get install wget  #!/bin/bash  cd /usr/local/src  //zlib  wget http://zlib.net/zlib-1.2.8.tar.gz  wget http://www.openssl.org/source/openssl-1.0.1f.tar.gz  wget ftp://ftp3.usa.openbsd.org/pub/OpenBSD/OpenSSH/portable/openssh-6.4p1.tar.gz  tar -xf openssh-6.4p1.tar.gz  tar -xf openssl-1.0.1f.tar.gz  tar -xf zlib-1.2.8.tar.gz  cd /usr/local/src  cd zlib-1.2.8  ./configure && make && sudo make install  cd ../openssl-1.0.1f  ./config && make && make test && sudo make install  cd ../openssh-6.4p1  ./configure --with-ipv4-default --with-md5-passwords && make  sudo groupadd sshd  sudo useradd -M -g sshd -c 'sshd privsep' -d /var/empty -s /sbin/nologin sshd  sudo passwd -l sshd  sudo make install  // set time  http://askubuntu.com/questions/254826/how-to-force-a-clock-update-using-ntp  apt-get install ntp  service ntp start  sudo service ntp stop  sudo ntpdate -s time.nist.gov  sudo service ntp start  //ssh  http://www.unixwiz.net/techtips/openssh.html  sshd re-exec requires execution with an absolute path  // install x11  // install arm compiler  sudo apt-get install libc6-dev-i386  download source  compile source  http://askubuntu.com/questions/251978/cannot-find-crti-o-no-such-file-or-directory  apt-get install libc-dev  export LIBRARY\_PATH=/usr/lib32:/usr/lib/x86\_64-linux-gnu  export LIBRARY\_PATH=/usr/lib/x86\_64-linux-gnu/  http://askubuntu.com/questions/98416/error-kernel-headers-not-found-but-they-are-in-place  sudo apt-get install linux-headers-$(uname-r)  http://stackoverflow.com/questions/11471722/libstdc-so-6-cannot-open-shared-object-file-no-such-file-or-directory  sudo apt-get install lib32z1  sudo apt-get install lib32z1 lib32z1-dev  sudo apt-get install lib32stdc++6  sudo apt-get install ia32-libs  sudo apt-get install uboot-mkimage  CHANGING IMAGE on Boot  http://sumanprasanna.wordpress.com/2012/12/11/how-to-change-your-boot-logo-in-linux-kernel/ |

**APPENDIX J**

Face Detection System SoCKit Test Results





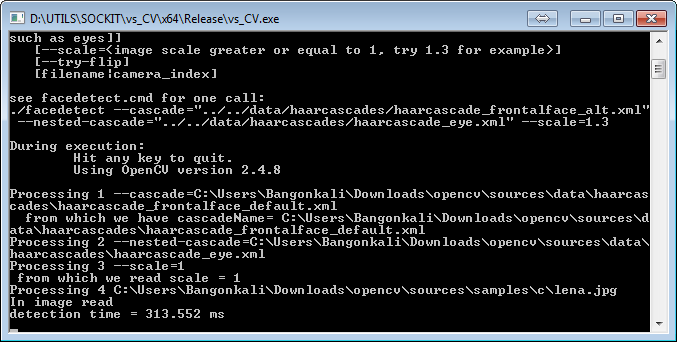
**APPENDIX K**

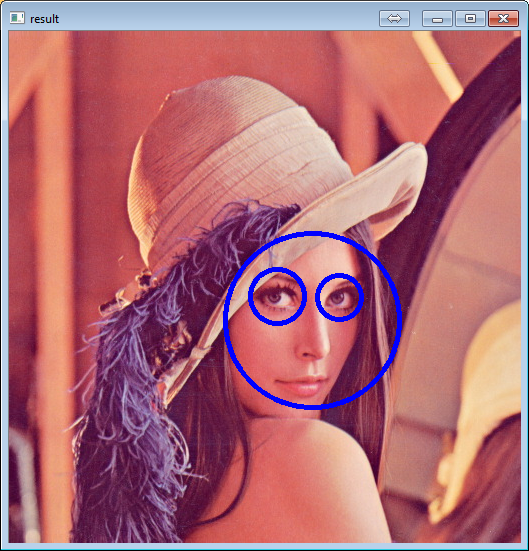
Standard 512x512 pixels test image.



**APPENDIX L**

Intel CISC Based Core i7 Test Results





**APPENDIX M**

Source code used for performance comparison facedetect.cpp

|  |
| --- |
| #include "stdafx.h"  #include "opencv2/objdetect/objdetect.hpp"  #include "opencv2/highgui/highgui.hpp"  #include "opencv2/imgproc/imgproc.hpp"  #include <cctype>  #include <iostream>  #include <iterator>  #include <stdio.h>  **using** **namespace** std**;**  **using** **namespace** cv**;**  static void help**()**  **{**  cout **<<** "\nThis program demonstrates the cascade recognizer. Now you can use Haar or LBP features.\n"  "This classifier can recognize many kinds of rigid objects, once the appropriate classifier is trained.\n"  "It's most known use is for faces.\n"  "Usage:\n"  "./facedetect [--cascade=<cascade\_path> this is the primary trained classifier such as frontal face]\n"  " [--nested-cascade[=nested\_cascade\_path this an optional secondary classifier such as eyes]]\n"  " [--scale=<image scale greater or equal to 1, try 1.3 for example>]\n"  " [--try-flip]\n"  " [filename|camera\_index]\n\n"  "see facedetect.cmd for one call:\n"  "./facedetect --cascade=\"../../data/haarcascades/haarcascade\_frontalface\_alt.xml\" --nested-cascade=\"../../data/haarcascades/haarcascade\_eye.xml\" --scale=1.3\n\n"  "During execution:\n\tHit any key to quit.\n"  "\tUsing OpenCV version " **<<** CV\_VERSION **<<** "\n" **<<** endl**;**  **}**  void detectAndDraw**(** Mat**&** img**,** CascadeClassifier**&** cascade**,**  CascadeClassifier**&** nestedCascade**,**  double scale**,** bool tryflip **);**  string cascadeName **=** "../../data/haarcascades/haarcascade\_frontalface\_alt.xml"**;**  string nestedCascadeName **=** "../../data/haarcascades/haarcascade\_eye\_tree\_eyeglasses.xml"**;**  int main**(** int argc**,** const char**\*\*** argv **)**  **{**  CvCapture**\*** capture **=** 0**;**  Mat frame**,** frameCopy**,** image**;**  const string scaleOpt **=** "--scale="**;**  size\_t scaleOptLen **=** scaleOpt**.**length**();**  const string cascadeOpt **=** "--cascade="**;**  size\_t cascadeOptLen **=** cascadeOpt**.**length**();**  const string nestedCascadeOpt **=** "--nested-cascade"**;**  size\_t nestedCascadeOptLen **=** nestedCascadeOpt**.**length**();**  const string tryFlipOpt **=** "--try-flip"**;**  size\_t tryFlipOptLen **=** tryFlipOpt**.**length**();**  string inputName**;**  bool tryflip **=** **false;**  help**();**  CascadeClassifier cascade**,** nestedCascade**;**  double scale **=** 1**;**  **for(** int i **=** 1**;** i **<** argc**;** i**++** **)**  **{**  cout **<<** "Processing " **<<** i **<<** " " **<<** argv**[**i**]** **<<** endl**;**  **if(** cascadeOpt**.**compare**(** 0**,** cascadeOptLen**,** argv**[**i**],** cascadeOptLen **)** **==** 0 **)**  **{**  cascadeName**.**assign**(** argv**[**i**]** **+** cascadeOptLen **);**  cout **<<** " from which we have cascadeName= " **<<** cascadeName **<<** endl**;**  **}**  **else** **if(** nestedCascadeOpt**.**compare**(** 0**,** nestedCascadeOptLen**,** argv**[**i**],** nestedCascadeOptLen **)** **==** 0 **)**  **{**  **if(** argv**[**i**][**nestedCascadeOpt**.**length**()]** **==** '=' **)**  nestedCascadeName**.**assign**(** argv**[**i**]** **+** nestedCascadeOpt**.**length**()** **+** 1 **);**  **if(** **!**nestedCascade**.**load**(** nestedCascadeName **)** **)**  cerr **<<** "WARNING: Could not load classifier cascade for nested objects" **<<** endl**;**  **}**  **else** **if(** scaleOpt**.**compare**(** 0**,** scaleOptLen**,** argv**[**i**],** scaleOptLen **)** **==** 0 **)**  **{**  **if(** **!**sscanf**(** argv**[**i**]** **+** scaleOpt**.**length**(),** "%lf"**,** **&**scale **)** **||** scale **<** 1 **)**  scale **=** 1**;**  cout **<<** " from which we read scale = " **<<** scale **<<** endl**;**  **}**  **else** **if(** tryFlipOpt**.**compare**(** 0**,** tryFlipOptLen**,** argv**[**i**],** tryFlipOptLen **)** **==** 0 **)**  **{**  tryflip **=** **true;**  cout **<<** " will try to flip image horizontally to detect assymetric objects\n"**;**  **}**  **else** **if(** argv**[**i**][**0**]** **==** '-' **)**  **{**  cerr **<<** "WARNING: Unknown option %s" **<<** argv**[**i**]** **<<** endl**;**  **}**  **else**  inputName**.**assign**(** argv**[**i**]** **);**  **}**  **if(** **!**cascade**.**load**(** cascadeName **)** **)**  **{**  cerr **<<** "ERROR: Could not load classifier cascade" **<<** endl**;**  help**();**  **return** **-**1**;**  **}**  **if(** inputName**.**empty**()** **||** **(**isdigit**(**inputName**.**c\_str**()[**0**])** **&&** inputName**.**c\_str**()[**1**]** **==** '\0'**)** **)**  **{**  capture **=** cvCaptureFromCAM**(** inputName**.**empty**()** **?** 0 **:** inputName**.**c\_str**()[**0**]** **-** '0' **);**  int c **=** inputName**.**empty**()** **?** 0 **:** inputName**.**c\_str**()[**0**]** **-** '0' **;**  **if(!**capture**)** cout **<<** "Capture from CAM " **<<** c **<<** " didn't work" **<<** endl**;**  **}**  **else** **if(** inputName**.**size**()** **)**  **{**  image **=** imread**(** inputName**,** 1 **);**  **if(** image**.**empty**()** **)**  **{**  capture **=** cvCaptureFromAVI**(** inputName**.**c\_str**()** **);**  **if(!**capture**)** cout **<<** "Capture from AVI didn't work" **<<** endl**;**  **}**  **}**  **else**  **{**  image **=** imread**(** "lena.jpg"**,** 1 **);**  **if(**image**.**empty**())** cout **<<** "Couldn't read lena.jpg" **<<** endl**;**  **}**  cvNamedWindow**(** "result"**,** 1 **);**  **if(** capture **)**  **{**  cout **<<** "In capture ..." **<<** endl**;**  **for(;;)**  **{**  IplImage**\*** iplImg **=** cvQueryFrame**(** capture **);**  frame **=** iplImg**;**  **if(** frame**.**empty**()** **)**  **break;**  **if(** iplImg**->**origin **==** IPL\_ORIGIN\_TL **)**  frame**.**copyTo**(** frameCopy **);**  **else**  flip**(** frame**,** frameCopy**,** 0 **);**  detectAndDraw**(** frameCopy**,** cascade**,** nestedCascade**,** scale**,** tryflip **);**  **if(** waitKey**(** 10 **)** **>=** 0 **)**  **goto** \_cleanup\_**;**  **}**  waitKey**(**0**);**  \_cleanup\_**:**  cvReleaseCapture**(** **&**capture **);**  **}**  **else**  **{**  cout **<<** "In image read" **<<** endl**;**  **if(** **!**image**.**empty**()** **)**  **{**  detectAndDraw**(** image**,** cascade**,** nestedCascade**,** scale**,** tryflip **);**  waitKey**(**0**);**  **}**  **else** **if(** **!**inputName**.**empty**()** **)**  **{**  /\* assume it is a text file containing the  list of the image filenames to be processed - one per line \*/  FILE**\*** f **=** fopen**(** inputName**.**c\_str**(),** "rt" **);**  **if(** f **)**  **{**  char buf**[**1000**+**1**];**  **while(** fgets**(** buf**,** 1000**,** f **)** **)**  **{**  int len **=** **(**int**)**strlen**(**buf**),** c**;**  **while(** len **>** 0 **&&** isspace**(**buf**[**len**-**1**])** **)**  len**--;**  buf**[**len**]** **=** '\0'**;**  cout **<<** "file " **<<** buf **<<** endl**;**  image **=** imread**(** buf**,** 1 **);**  **if(** **!**image**.**empty**()** **)**  **{**  detectAndDraw**(** image**,** cascade**,** nestedCascade**,** scale**,** tryflip **);**  c **=** waitKey**(**0**);**  **if(** c **==** 27 **||** c **==** 'q' **||** c **==** 'Q' **)**  **break;**  **}**  **else**  **{**  cerr **<<** "Aw snap, couldn't read image " **<<** buf **<<** endl**;**  **}**  **}**  fclose**(**f**);**  **}**  **}**  **}**  cvDestroyWindow**(**"result"**);**  **return** 0**;**  **}**  void detectAndDraw**(** Mat**&** img**,** CascadeClassifier**&** cascade**,**  CascadeClassifier**&** nestedCascade**,**  double scale**,** bool tryflip **)**  **{**  int i **=** 0**;**  double t **=** 0**;**  vector**<**Rect**>** faces**,** faces2**;**  const static Scalar colors**[]** **=** **{** CV\_RGB**(**0**,**0**,**255**),**  CV\_RGB**(**0**,**128**,**255**),**  CV\_RGB**(**0**,**255**,**255**),**  CV\_RGB**(**0**,**255**,**0**),**  CV\_RGB**(**255**,**128**,**0**),**  CV\_RGB**(**255**,**255**,**0**),**  CV\_RGB**(**255**,**0**,**0**),**  CV\_RGB**(**255**,**0**,**255**)}** **;**  Mat gray**,** smallImg**(** cvRound **(**img**.**rows**/**scale**),** cvRound**(**img**.**cols**/**scale**),** CV\_8UC1 **);**  cvtColor**(** img**,** gray**,** CV\_BGR2GRAY **);**  resize**(** gray**,** smallImg**,** smallImg**.**size**(),** 0**,** 0**,** INTER\_LINEAR **);**  equalizeHist**(** smallImg**,** smallImg **);**  t **=** **(**double**)**cvGetTickCount**();**  cascade**.**detectMultiScale**(** smallImg**,** faces**,**  1.1**,** 2**,** 0  //|CV\_HAAR\_FIND\_BIGGEST\_OBJECT  //|CV\_HAAR\_DO\_ROUGH\_SEARCH  **|**CV\_HAAR\_SCALE\_IMAGE  **,**  Size**(**30**,** 30**)** **);**  **if(** tryflip **)**  **{**  flip**(**smallImg**,** smallImg**,** 1**);**  cascade**.**detectMultiScale**(** smallImg**,** faces2**,**  1.1**,** 2**,** 0  //|CV\_HAAR\_FIND\_BIGGEST\_OBJECT  //|CV\_HAAR\_DO\_ROUGH\_SEARCH  **|**CV\_HAAR\_SCALE\_IMAGE  **,**  Size**(**30**,** 30**)** **);**  **for(** vector**<**Rect**>::**const\_iterator r **=** faces2**.**begin**();** r **!=** faces2**.**end**();** r**++** **)**  **{**  faces**.**push\_back**(**Rect**(**smallImg**.**cols **-** r**->**x **-** r**->**width**,** r**->**y**,** r**->**width**,** r**->**height**));**  **}**  **}**  t **=** **(**double**)**cvGetTickCount**()** **-** t**;**  printf**(** "detection time = %g ms\n"**,** t**/((**double**)**cvGetTickFrequency**()\***1000.**)** **);**  **for(** vector**<**Rect**>::**const\_iterator r **=** faces**.**begin**();** r **!=** faces**.**end**();** r**++,** i**++** **)**  **{**  Mat smallImgROI**;**  vector**<**Rect**>** nestedObjects**;**  Point center**;**  Scalar color **=** colors**[**i**%**8**];**  int radius**;**  double aspect\_ratio **=** **(**double**)**r**->**width**/**r**->**height**;**  **if(** 0.75 **<** aspect\_ratio **&&** aspect\_ratio **<** 1.3 **)**  **{**  center**.**x **=** cvRound**((**r**->**x **+** r**->**width**\***0.5**)\***scale**);**  center**.**y **=** cvRound**((**r**->**y **+** r**->**height**\***0.5**)\***scale**);**  radius **=** cvRound**((**r**->**width **+** r**->**height**)\***0.25**\***scale**);**  circle**(** img**,** center**,** radius**,** color**,** 3**,** 8**,** 0 **);**  **}**  **else**  rectangle**(** img**,** cvPoint**(**cvRound**(**r**->**x**\***scale**),** cvRound**(**r**->**y**\***scale**)),**  cvPoint**(**cvRound**((**r**->**x **+** r**->**width**-**1**)\***scale**),** cvRound**((**r**->**y **+** r**->**height**-**1**)\***scale**)),**  color**,** 3**,** 8**,** 0**);**  **if(** nestedCascade**.**empty**()** **)**  **continue;**  smallImgROI **=** smallImg**(\***r**);**  nestedCascade**.**detectMultiScale**(** smallImgROI**,** nestedObjects**,**  1.1**,** 2**,** 0  //|CV\_HAAR\_FIND\_BIGGEST\_OBJECT  //|CV\_HAAR\_DO\_ROUGH\_SEARCH  //|CV\_HAAR\_DO\_CANNY\_PRUNING  **|**CV\_HAAR\_SCALE\_IMAGE  **,**  Size**(**30**,** 30**)** **);**  **for(** vector**<**Rect**>::**const\_iterator nr **=** nestedObjects**.**begin**();** nr **!=** nestedObjects**.**end**();** nr**++** **)**  **{**  center**.**x **=** cvRound**((**r**->**x **+** nr**->**x **+** nr**->**width**\***0.5**)\***scale**);**  center**.**y **=** cvRound**((**r**->**y **+** nr**->**y **+** nr**->**height**\***0.5**)\***scale**);**  radius **=** cvRound**((**nr**->**width **+** nr**->**height**)\***0.25**\***scale**);**  circle**(** img**,** center**,** radius**,** color**,** 3**,** 8**,** 0 **);**  **}**  **}**  cv**::**imshow**(** "result"**,** img **);**  **}** |