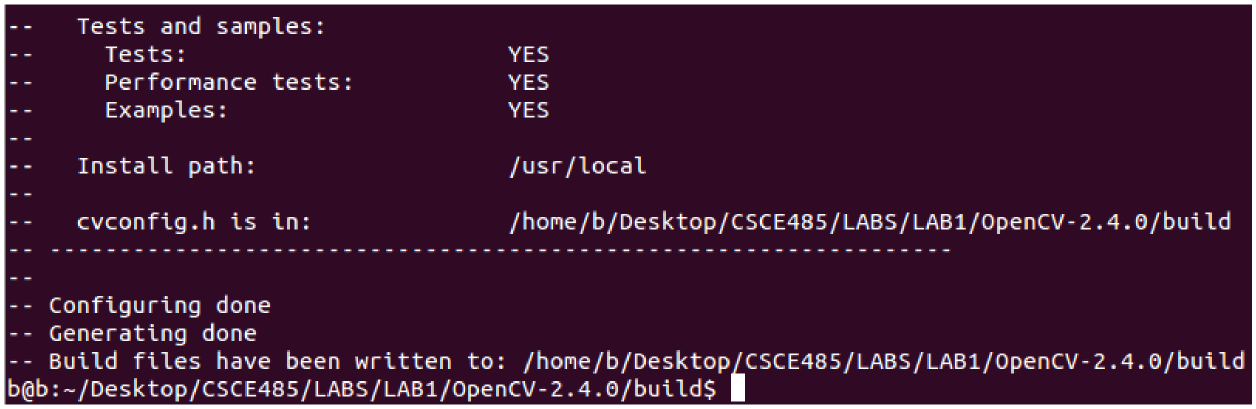
CSCE - 485
LAB1 - Familiarity with Linux OpenCV
Bruno Lopes

1.

The article presented explained the use of computer vision for video analytics and explores a few different applications in current scenarios. Professor Siewert also explains the power of video analytics on practical applications, and the demand for new solutions.
The the third part of the Cloud Scaling article covers the complex process on how video analytics is captured, processed and transmitted on embedded intelligent sensors and cloud based processing. The article describes the ideal but complicated process of inverse rendering and how a combination of autonomous systems with high capacity processing computers could achieve indistinguishable real time interaction.

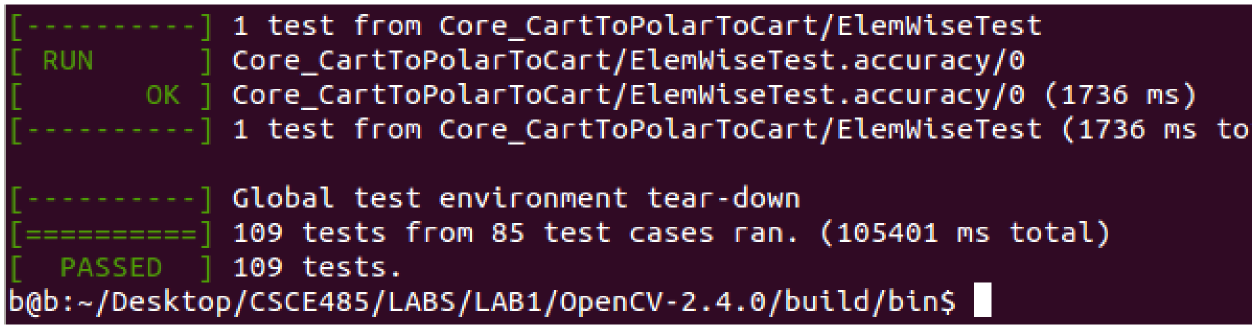
2.

OpenCV installation:



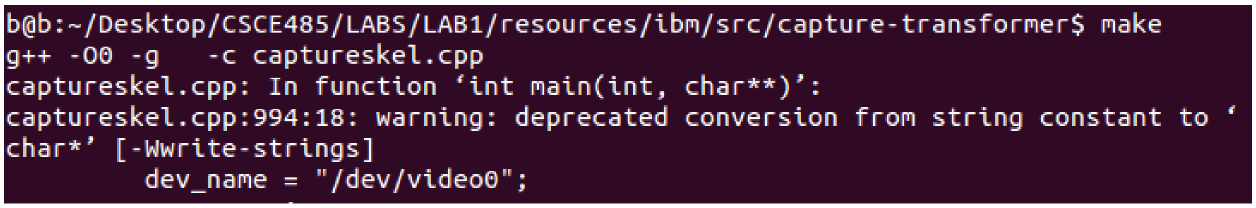
```
--   Tests and samples:
--     Tests:                      YES
--     Performance tests:          YES
--     Examples:                   YES
--
--   Install path:                 /usr/local
--
--   cvconfig.h is in:             /home/b/Desktop/CSCE485/LABS/LAB1/OpenCV-2.4.0/build
-- -----------------------------------------------------------------
--
-- Configuring done
-- Generating done
-- Build files have been written to: /home/b/Desktop/CSCE485/LABS/LAB1/OpenCV-2.4.0/build
b@b:~/Desktop/CSCE485/LABS/LAB1/OpenCV-2.4.0/build$
```

Testing OpenCV library:



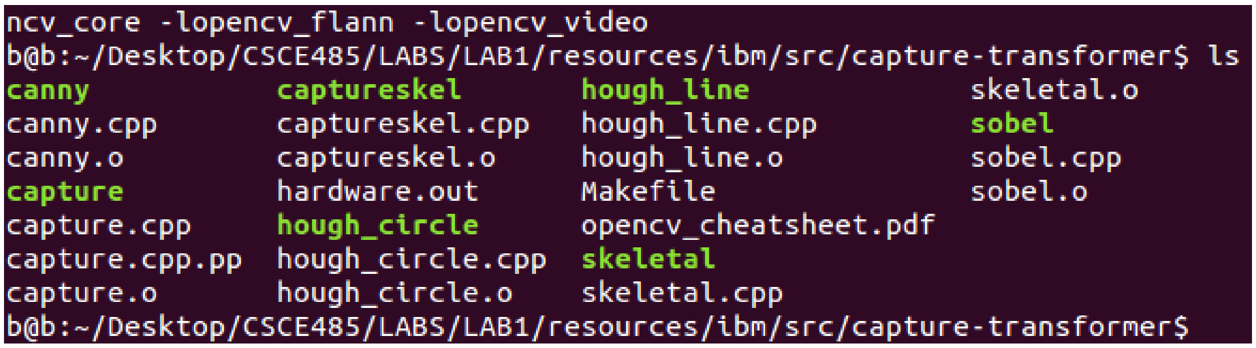
```
[----------] 1 test from Core_CartToPolarToCart/ElemWiseTest
[ RUN      ] Core_CartToPolarToCart/ElemWiseTest.accuracy/0
[       OK ] Core_CartToPolarToCart/ElemWiseTest.accuracy/0 (1736 ms)
[----------] 1 test from Core_CartToPolarToCart/ElemWiseTest (1736 ms to

[----------] Global test environment tear-down
[==========] 109 tests from 85 test cases ran. (105401 ms total)
[  PASSED  ] 109 tests.
b@b:~/Desktop/CSCE485/LABS/LAB1/OpenCV-2.4.0/build/bin$
```

Building the code in capture transformer:



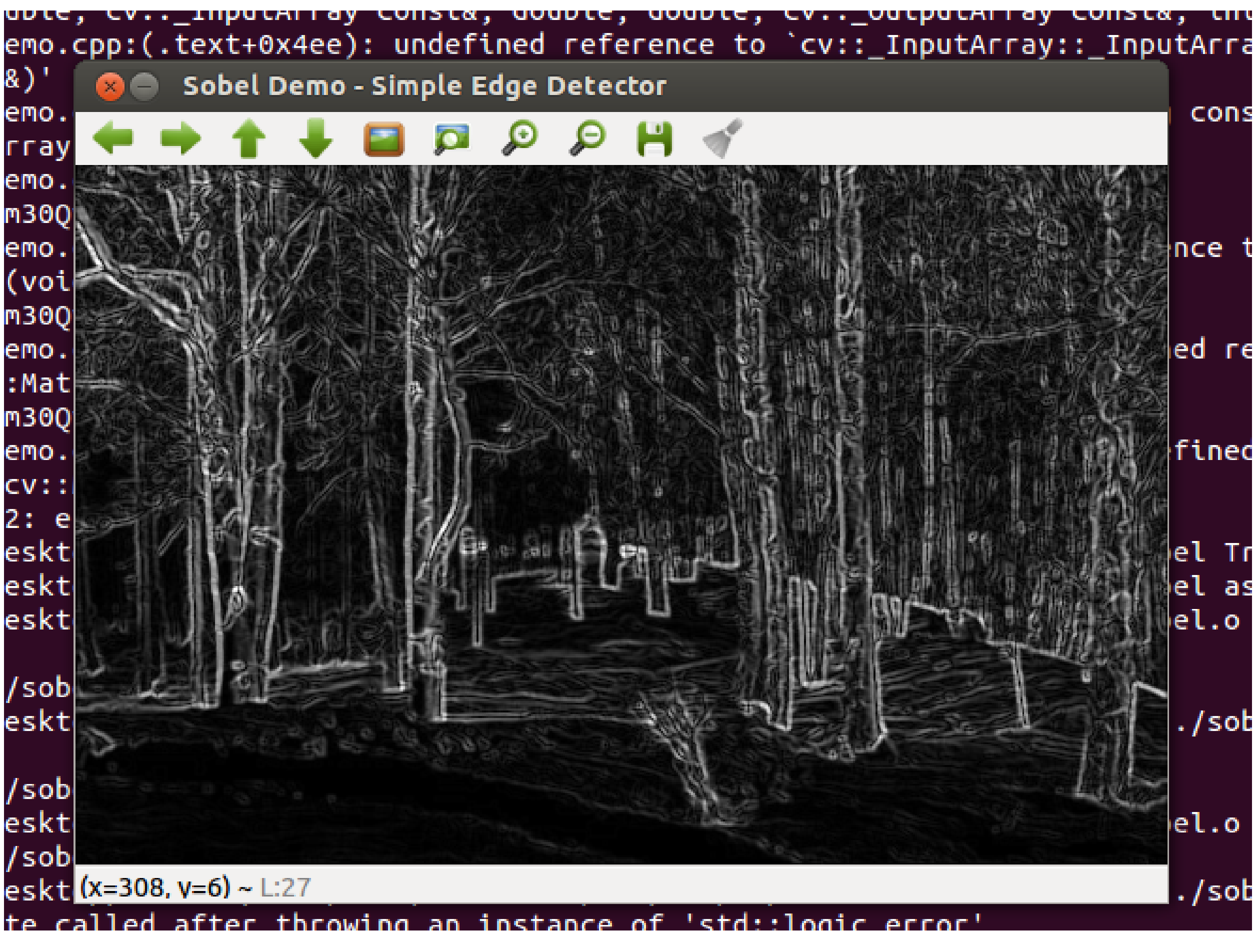
```
b@b:~/Desktop/CSCE485/LABS/LAB1/resources/ibm/src/capture-transformer$ make
g++ -O0 -g   -c captureskel.cpp
captureskel.cpp: In function ‘int main(int, char**)’:
captureskel.cpp:994:18: warning: deprecated conversion from string constant to ‘
char*’ [-Wwrite-strings]
         dev_name = "/dev/video0";
```

Sobel created:



```
ncv_core -lopencv_flann -lopencv_video
b@b:~/Desktop/CSCE485/LABS/LAB1/resources/ibm/src/capture-transformer$ ls
canny            captureskel      hough_line           skeletal.o
canny.cpp        captureskel.cpp  hough_line.cpp       sobel
canny.o          captureskel.o    hough_line.o         sobel.cpp
capture          hardware.out     Makefile             sobel.o
capture.cpp      hough_circle     opencv_cheatsheet.pdf
capture.cpp.pp   hough_circle.cpp skeletal
capture.o        hough_circle.o   skeletal.cpp
b@b:~/Desktop/CSCE485/LABS/LAB1/resources/ibm/src/capture-transformer$
```

Sobel transform result:

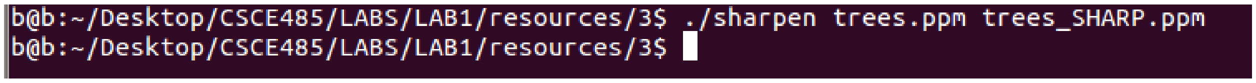


```
ubte, cv::_InputArray const&, double, double, cv::_OutputArray const&, int
emo.cpp:(.text+0x4ee): undefined reference to `cv::_InputArray::_InputArra
&)'
emo.                                                                  cons
rray
emo.
m30Q
emo.                                                                  nce t
(voi
m30Q
emo.                                                                  ed re
:Mat
m30Q
emo.                                                                  fined
cv::
2: e
eskt                                                                  el Tr
eskt                                                                  el as
eskt                                                                  el.o

/sob
eskt                                                                  ./sob

/sob
eskt                                                                  el.o
/sob
eskt                                                                  ./sob
te called after throwing an instance of 'std::logic_error'
```

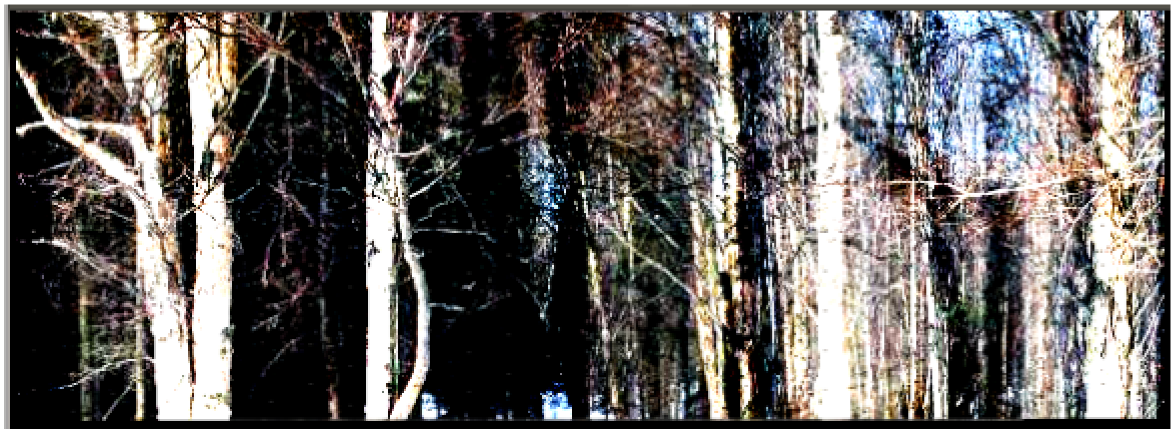
Sobel transform is a image processing algorithm that uses a discrete differentiation operation in order to provide edge detection from an image. The well explained OpenCV Sobel tutorial navigates to the very fundamental processes and functions used to apply the Sobel transform on a static image and generate a result. The transform used on the demo code apply the Sobel Operator onto a local image file in order to generate an output with edge detection lines in gradient, resulted from the original input. The process includes a GaussianBlur, that uses the mathematical Laplace Transform in order to detect boundaries and reduce noise.

3.

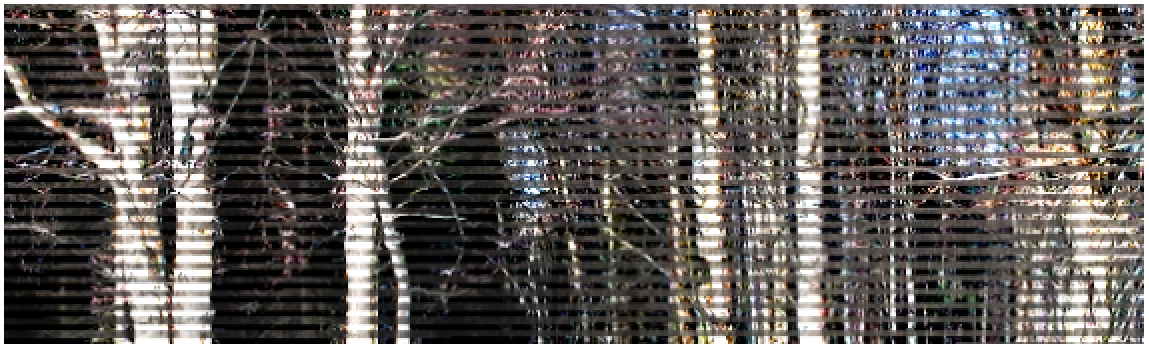
./sharpen convolution on **trees.ppm**:

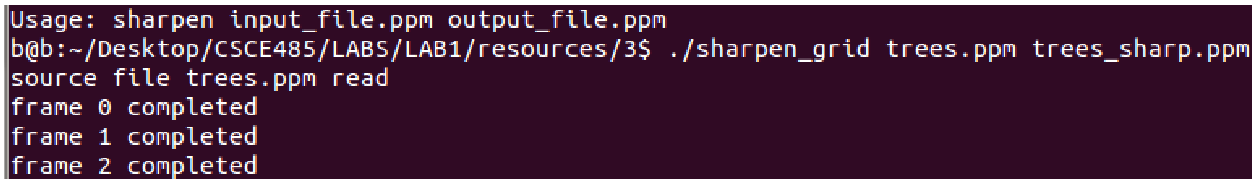


```
b@b:~/Desktop/CSCE485/LABS/LAB1/resources/3$ ./sharpen trees.ppm trees_SHARP.ppm
b@b:~/Desktop/CSCE485/LABS/LAB1/resources/3$ 
```

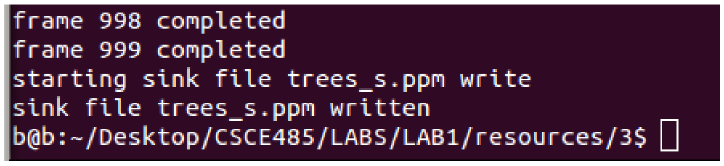


./sharpen_grid frame-by-frame convolution on **trees.ppm**:



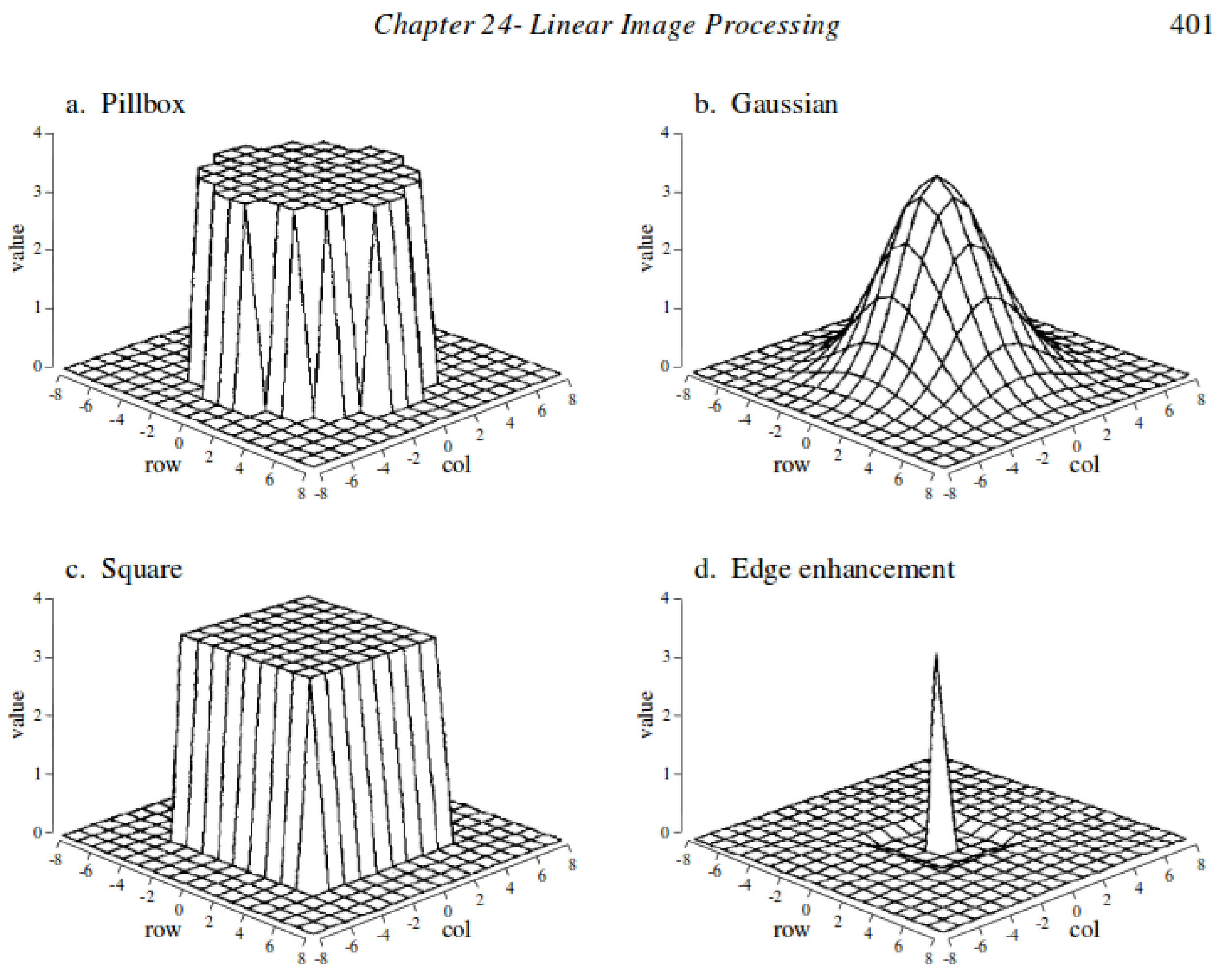


```
Usage: sharpen input_file.ppm output_file.ppm
b@b:~/Desktop/CSCE485/LABS/LAB1/resources/3$ ./sharpen_grid trees.ppm trees_sharp.ppm
source file trees.ppm read
frame 0 completed
frame 1 completed
frame 2 completed
```



```
frame 998 completed
frame 999 completed
starting sink file trees_s.ppm write
sink file trees_s.ppm written
b@b:~/Desktop/CSCE485/LABS/LAB1/resources/3$ 
```

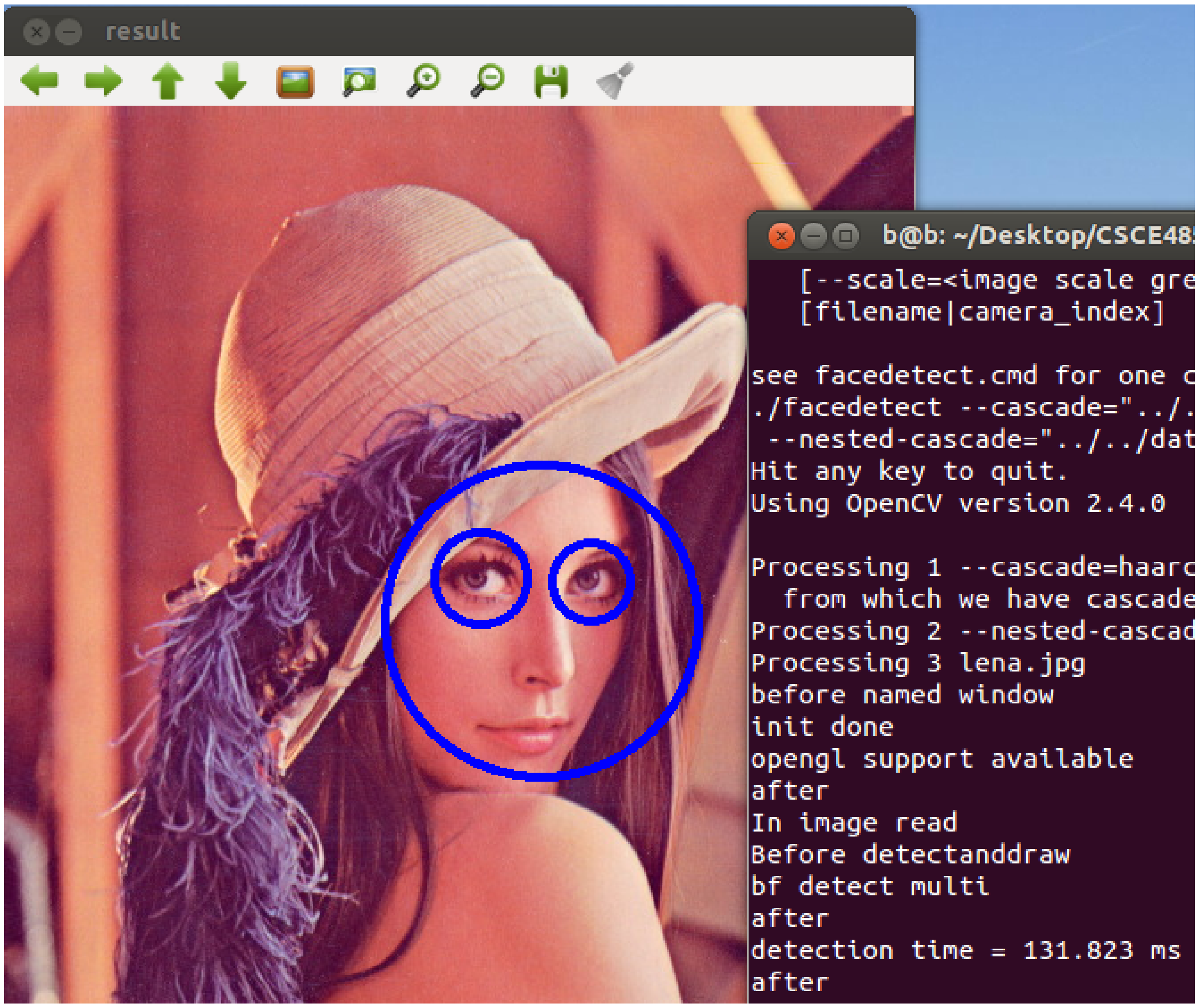
Point Spread Function can perform as a high-pass filter and provide edge enhancement, it works as the mathematical representation of an unresolved point source. The linear smoothing filtering alone performs poorly because linear filtering cannot properly separate signals generated by non-linear operations, different strategies can be used to tackle different challenges and because of the algorithm boundaries detection done on PSF Gaussian can be used as an edge enhancing filter:



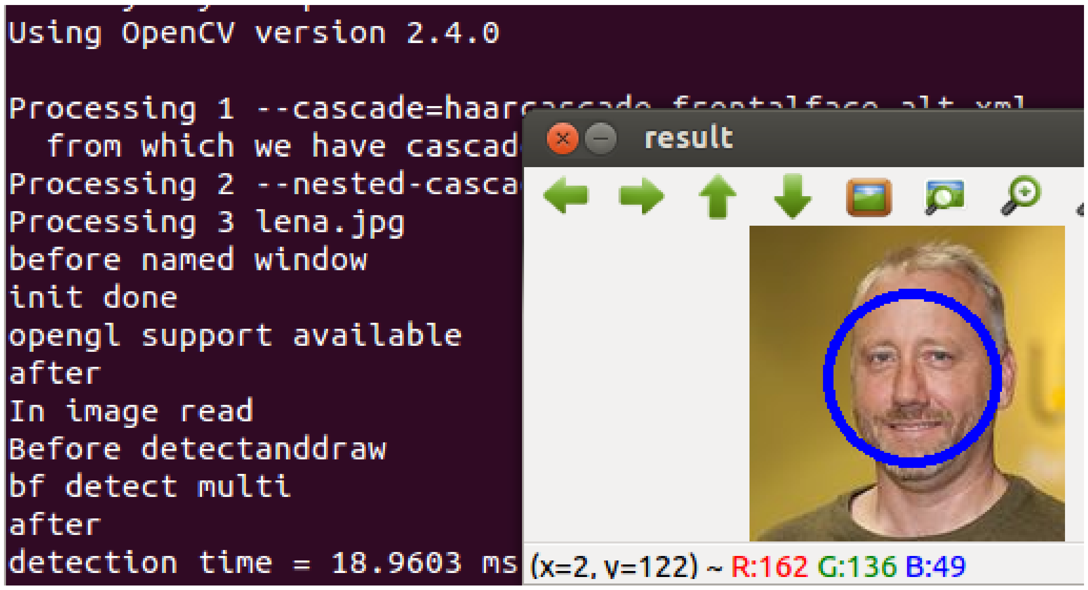
The sample code **sharpen** differs from **sharpen_grid** in complexity and strategy. The **sharpen_grid** is build to analyze and calculate frame by frame (double loop IMG_HEIGHT and IMG_WIDTH), and thread each convolve separately leading to a possible scalability. The **sharpen** sample is simpler, it operates on a similar structure however instead of looping on an amount of frames it performs a quick general approximation across the image (single loop IMG_HEIGHT * IMG_WIDTH)resulting on a quicker smoothing filter. I believe that **sharpen** would be a better real time frame implementation because it does not waste time rendering the whole image, only the significant areas between frames.

4.

faceDetect demo:



```
   [--scale=<image scale gre
   [filename|camera_index]

see facedetect.cmd for one c
./facedetect --cascade="../.
 --nested-cascade="../../dat
Hit any key to quit.
Using OpenCV version 2.4.0

Processing 1 --cascade=haarc
  from which we have cascade
Processing 2 --nested-cascad
Processing 3 lena.jpg
before named window
init done
opengl support available
after
In image read
Before detectanddraw
bf detect multi
after
detection time = 131.823 ms
after
```



```
Using OpenCV version 2.4.0

Processing 1 --cascade=haarcascade_frontalface_alt.xml
  from which we have cascad
Processing 2 --nested-casca
Processing 3 lena.jpg
before named window
init done
opengl support available
after
In image read
Before detectanddraw
bf detect multi
after
detection time = 18.9603 ms
```

The Object detection algorithm provided by Viola-Jones initially reduces the working window area to a 24x24 pixel frame then it performs a series of sum/subtraction of black and white pixel sized rectangles depending on the analysis predefined by the machine to capture image features. Once this “face” window is defined the algorithm focus groups of features that are applied individually based on the features found, the more features found the higher face detection accuracy becomes, the algorithm performance comes from applying as many feature as possible without wasting processing time, the concept of Cascade of Classifiers deal with this ratio and tend to optimize the use of features.

5.

makeborder demo:

I had a few issues getting my code to work, my main strategy was to use:
borderType = BORDER_CONSTANT and color it using color-code **14** for yellow

I plan to seek support along the week,