Université Concordia University
FACULTY OF ENGINEERING AND COMPUTER SCIENCE
Department of Computer Science and Software Engineering

COMP498G/691G COMPUTER VISION

TUTORIAL 01

Today's Tutorial

- OpenCV
  - Loading/Saving images
  - Basic data structures
  - Filtering
  - Camera
- Questions

http://docs.opencv.org/2.4/index.html

Loading and image

```
1  // OpenCV
2  #include <opencv2/core/core.hpp>
3  #include <opencv2/highgui/highgui.hpp>
4  #include <stdio.h>
5
6  using namespace cv;
7
8  int main( int argc, char** argv )
9  {
10  char* imageName = argv[1];
11
12  Mat image;
13  image = imread( imageName, 1 );
14
15  if( argc != 2 || !image.data )
16  {
17    printf( " No image data n " );
18    return -1;
19  }
20  namedWindow( imageName, CV_WINDOW_AUTOSIZE );
21  imshow( imageName, image );
22  waitKey(0);
23
24  return 0;
25 }
```

Convert RGB image to Grayscale

```
#include <opencv2/core/core.hpp>
#include <opencv2/highgui/highgui.hpp>
#include <stdio.h>

using namespace cv;

int main( int argc, char** argv )
{
 char* imageName = argv[1];

 Mat image;
 image = imread( imageName, 0 );

 if( argc != 2 || !image.data )
 {
   printf( " No image data n " );
   return -1;
 }

 namedWindow( imageName, CV_WINDOW_AUTOSIZE );
 imshow( imageName, image );
 waitKey(0);

 return 0;
}
```

Blurring an image

```
// OpenCV
#include <opencv2/core/core.hpp>
#include <opencv2/highgui/highgui.hpp>
#include <opencv2/imgproc/imgproc.hpp>
#include <stdio.h>

using namespace cv;

int main( int argc, char** argv )
{
 char* imageName = argv[1];

 Mat image;
 image = imread( imageName, 1 );

 if( argc != 2 || !image.data )
 {
   printf( " No image data n " );
   return -1;
 }
 blur(image,image,Size(10,10));
 namedWindow( imageName, CV_WINDOW_AUTOSIZE );
 imshow( imageName, image );
 waitKey(0);

 return 0;
}
```

http://docs.opencv.org/2.4/doc/tutorials/imgproc/gausian_median_blur_bilateral_filter/gausian_median_blur_bilateral_filter.html

Pixel access

```cpp
#include <opencv2/core/core.hpp>
#include <opencv2/highgui/highgui.hpp>
#include <opencv2/imgproc/imgproc.hpp>
#include <stdio.h>

using namespace cv;

int main( int argc, char** argv )
{
 char* imageName = argv[1];

 Mat image;
 image = imread( imageName, 1 );

 if( argc != 2 || !image.data )
 {
   printf( " No image data n " );
   return -1;
 }

 // Pixel access
 uchar pixValue;
 for (int i = 0; i < image.cols; i++) {
    for (int j = 0; j < image.rows; j++) {
        Vec3b &intensity = image.at<Vec3b>(j, i);
        for(int k = 0; k < image.channels(); k++) {
            // calculate pixValue
            image.at<Vec3b>(j, i)[0] = 2*image.at<Vec3b>(j, i)[0];
            image.at<Vec3b>(j, i)[1] = 2*image.at<Vec3b>(j, i)[1];
            image.at<Vec3b>(j, i)[2] = 2*image.at<Vec3b>(j, i)[2];
        }
     }
 }

 namedWindow( "talkera.org/opencv", CV_WINDOW_AUTOSIZE );
 imshow( "talkera.org/opencv", image );
 waitKey(0);

 return 0;
}
```

```cpp
// OpenCV Diplay camera.
#include <opencv2/objdetect/objdetect.hpp>
#include <opencv2/highgui/highgui.hpp>
#include <opencv2/imgproc/imgproc.hpp>

#include <iostream>
#include <stdio.h>

using namespace std;
using namespace cv;


int main( int argc, const char** argv )
{
   VideoCapture cap(0); // open the default camera
   if(!cap.isOpened()) // check if we succeeded
     return -1;

   Mat edges;
   namedWindow("edges",1);

   for(;;)
   {
     Mat frame;
     cap >> frame;
     imshow("webcam", frame);
     if(waitKey(30) >= 0) break;
    }
}
```

CONCORDIA.CA