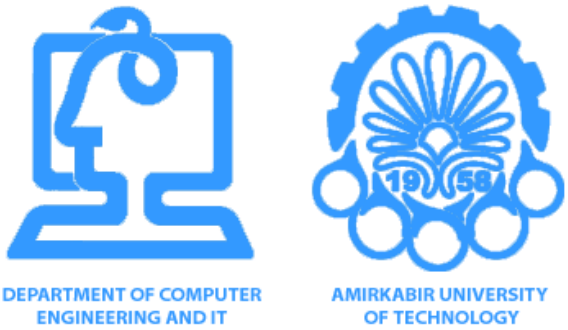


In the name of Allah

Amirkabir University of Technology

Department of Computer Engineering and IT

Artificial Intelligence Course Project

Image Processing Using OpenCV

Programmer: Amir.M Pirhosseinloo

Email: amirphl@aut.ac.ir

Github: amirphl

Linkedin: amirmohammad pirhosseinloo

Professor: Dr Ahmad Nikabadi

“What we need to do is always lean into the future; when the world changes around you and when it changes against you – what used to be a tail wind is now a head wind – you have to lean into that and figure out what to do because complaining isn’t a strategy.”

Jeff Bezos

Contents

1. Project definition

In this project, I have implemented an application which receives an image and a video as input and performs some processing on it. Image processes include:

a) separating blue channel
b) converting to greyscale
c) smoothing with Gaussian filter
d) rotating 90 degrees
e) resizing (reducing to half width)
f) edge detection
g) segmentation
h) face detection

Video processing consists of extracting 5 initial frames of video and displaying it with 500ms delay.

2. Structure of Application

The application is implemented using OpenCV library and c++ programming language.

main.cpp contains the implementation and you should run it to see the results.

Before running main.cpp, CmakeList.txt must be configured.

In CmakeList.txt you must configure:

a) OpenCV_ROOT_DIR
b) OpenCV_DIR
c) OpenCV REQUIRED PATHS

Before configuring, make sure that your OpenCV libraries contains core.hpp, highgui.hpp, imgproc.hpp, objdetect.hpp and cv.hpp.

After configuration, you can build project and run it.

For running, you must provide 4 program arguments:

a) first, path to input image
b) second, path to input video
c) third, path to first classifier for face detection
d) fourth, path to second classifier for face detection

This is an example of input:

```
C:\Users\amirphl\CLionProjects\Imgae_Processing_Project_Phase_1\Lenna.png
C:\Users\amirphl\Desktop\videos\1.mp4
C:\opencv\mingw-build\install\etc\haarcascades\haarcascade_fullbody.xml
C:\opencv\mingw-
build\install\etc\haarcascades\haarcascade_frontalface_default.xml
```

Haar cascade classifiers are available at /path/to/opencv/install/etic/haarcascades/…

After running program, results display in order, for see next result, you should press space key. Each result is saved in directory which the code is running.

Challenges I met are available at challenges/ folder.

References used are available at references/ folder.

3. Results for Lenna picture

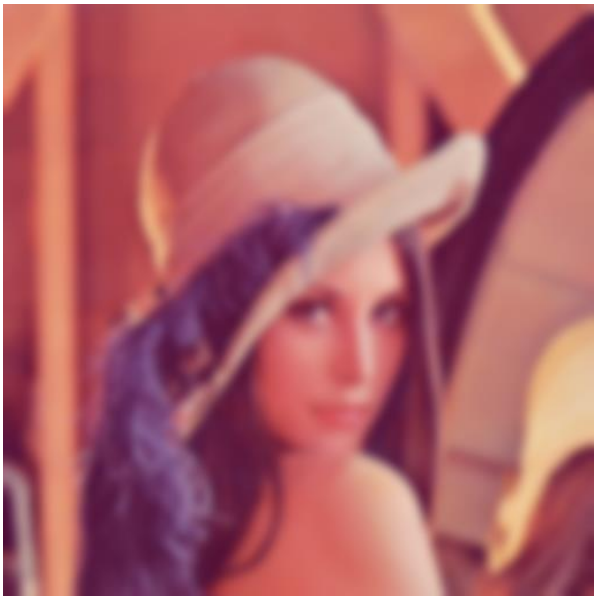
a. input(Lenna)

b.

c. blue channel

d. greyscale image

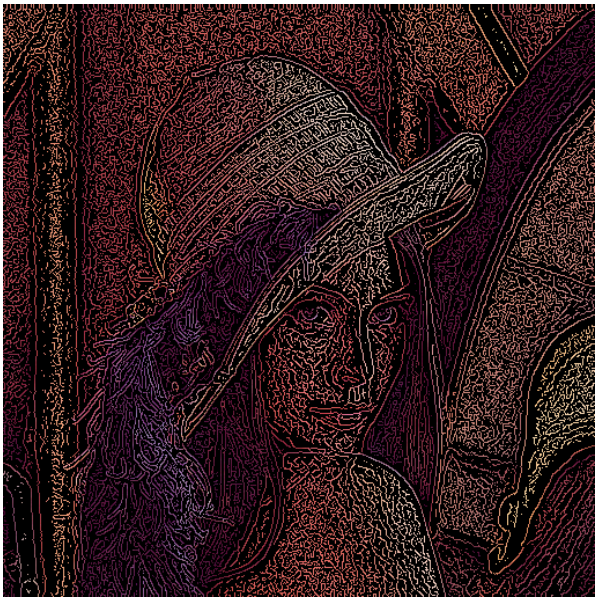
e. smoothed image with Gaussian filter



f. resized image

g. rotated image

h. detected edges



i. segmented image

j. face detected