openCV

Link used to build and install openCV 4 :

https://raspberrypi.stackexchange.com/questions/116592/install-opencv-in-a-raspberry-pi-3-with-c

Other useful links :

https://www.raspberrypi.org/forums/viewtopic.php?t=262144

Link used to build test codes using cmake :

https://docs.opencv.org/master/db/df5/tutorial_linux_gcc_cmake.html

Steps

Create a program using OpenCV

Let's use a simple program such as DisplayImage.cpp shown below.

```
#include <stdio.h>
#include <opencv2/opencv.hpp>
using namespace cv;
int main(int argc, char** argv )
{
if ( argc != 2 )
{
printf("usage: DisplayImage.out <Image_Path>\n");
return -1;
}
Mat image;
image = imread( argv[1], 1 );
if ( !image.data )
{
printf("No image data \n");
return -1;
}
namedWindow("Display Image", WINDOW_AUTOSIZE );
imshow("Display Image", image);
waitKey(0);
return 0;
}
```

Create a CMake file

Now you have to create your CMakeLists.txt file. It should look like this:

```
cmake_minimum_required(VERSION 2.8)
project( DisplayImage )
find_package( OpenCV REQUIRED )
include_directories( ${OpenCV_INCLUDE_DIRS} )
add_executable( DisplayImage DisplayImage.cpp )
target_link_libraries( DisplayImage ${OpenCV_LIBS} )
```

Generate the executable

This part is easy, just proceed as with any other project using CMake:

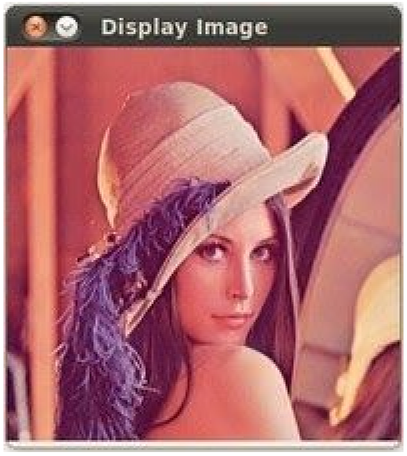
```
cd <DisplayImage_directory>
cmake .
make
```

Result

By now you should have an executable (called DisplayImage in this case). You just have to run it giving an image location as an argument, i.e.:

```
./DisplayImage lena.jpg
```

You should get a nice window as the one shown below:



Useful Links

https://my-first-project.dev/2019/12/01/object-detection-and-tracking-with-opencv-and-c-plus-plus/

People Counter using HOG (Histogram of Oriented Gradients)

https://www.youtube.com/watch?v=28xk5i1_7Zc
https://github.com/intel-iot-devkit/people-counter-opencv