codes of opencv using C++

Webcam

```
#include <opencv2/imgcodecs.hpp>
 #include <opencv2/highgui.hpp>
 #include <opencv2/imgproc.hpp>
 #include <iostream>

 using namespace cv;
 using namespace std;


 /////////////////  Images  //////////////////////

 int main() {

     //string path = "/Users/animeshkumarnayak/Downloads/Resources/test_video.mp4";
     VideoCapture cap(0);
     Mat img;

     while (true){
         cap.read(img);

         //string path = "/Users/animeshkumarnayak/Downloads/Resources/lambo.png";
         //Mat img = imread(path);
         imshow("Image", img);
         waitKey(1);
         //return 0;
     }
 }
```

……………………………………………………………………………………………….

Video saved.

```
#include <opencv2/imgcodecs.hpp>
 #include <opencv2/highgui.hpp>
 #include <opencv2/imgproc.hpp>
 #include <iostream>

 using namespace cv;
 using namespace std;


 /////////////////  Images  //////////////////////
 int main() {

     string path = "/Users/animeshkumarnayak/Downloads/Resources/
test_video.mp4";
     VideoCapture cap(path);
     Mat img;

     while (true){
         cap.read(img);

         //string path = "/Users/animeshkumarnayak/Downloads/Resources/
lambo.png";
         //Mat img = imread(path);
         imshow("Image", img);
         waitKey(1);
         //return 0;
     }
 }
```

……………………………………………………………………………………………………….

Canny edge detection

```
#include <opencv2/imgcodecs.hpp>
 #include <opencv2/highgui.hpp>
 #include <opencv2/imgproc.hpp>
 #include <iostream>

 using namespace cv;
 using namespace std;


 /////////////////  Images  //////////////////////

 int main() {

     //string path = "/Users/animeshkumarnayak/Downloads/Resources/
test_video.mp4";
     VideoCapture cap(0);
     Mat img;
     Mat imgGray, imgBlur,imgCanny;
     while (true){
         cap.read(img);
         cvtColor(img,imgGray , COLOR_BGR2GRAY);
         GaussianBlur(img,imgBlur ,Size(7,7),5.0 );
         Canny(imgBlur, imgCanny, 50, 150);

         //string path = "/Users/animeshkumarnayak/Downloads/Resources/
lambo.png";
         //Mat img = imread(path);
         //imshow("Image", imgGray);
         imshow("Image",imgCanny);
         waitKey(1);
         //return 0;
     }
 }
```

…….

Erode dilation in canny

```
#include <opencv2/imgcodecs.hpp>
 #include <opencv2/highgui.hpp>
 #include <opencv2/imgproc.hpp>
 #include <iostream>

 using namespace cv;
 using namespace std;


 /////////////////  Images  //////////////////////

 int main() {

    //string path = "/Users/animeshkumarnayak/Downloads/Resources/
test_video.mp4";
    VideoCapture cap(0);
    Mat img;
    Mat imgGray, imgBlur,imgCanny,imgDil,imgErode;
    while (true){
        cap.read(img);
        cvtColor(img,imgGray , COLOR_BGR2GRAY);
        GaussianBlur(img,imgBlur ,Size(7,7),5.0 );
        Canny(imgBlur, imgCanny, 50, 150);

        Mat kernal = getStructuringElement(MORPH_RECT, Size(3 ,3));
        dilate(imgCanny, imgDil, kernal);
        erode(imgDil,imgErode,kernal);

        //string path = "/Users/animeshkumarnayak/Downloads/Resources/
lambo.png";
        //Mat img = imread(path);
        //imshow("Image", imgGray);
        imshow("Image",imgErode);
        waitKey(1);
        //return 0;
    }
 }
```

………

Resize.

```
#include <opencv2/imgcodecs.hpp>
 #include <opencv2/highgui.hpp>
 #include <opencv2/imgproc.hpp>
 #include <iostream>

 using namespace cv;
 using namespace std;


 /////////////////  Images  //////////////////////

 int main() {

     string path = "/Users/animeshkumarnayak/Downloads/
Aesthetic_Desktop_Wallpaper.png";
     //VideoCapture cap(0);
     Mat img=imread(path);
     Mat imgResize;


     //cout << img.size() <<endl;
     resize(img, imgResize, Size (640,480));

     //resizing using scaling
     resize(img, imgResize, Size (),0.5,0.5);

     imshow("Image",img);
     imshow("ImageResize",imgResize);
     waitKey(0);
     //return 0;

 }
```

………………………………………………………………………………………

croping and Roi

```
#include <opencv2/imgcodecs.hpp>
 #include <opencv2/highgui.hpp>
 #include <opencv2/imgproc.hpp>
 #include <iostream>

 using namespace cv;
 using namespace std;


 /////////////////  Images  //////////////////////

 int main() {

    string path = "/Users/animeshkumarnayak/Downloads/Resources/dep.jpeg";
    //VideoCapture cap(0);
    Mat img=imread(path);
    Mat imgResize, imgCrop;

    //resizing using scaling
    cout << img.size() <<endl;
    resize(img, imgResize, Size (),0.5,0.5);
    Rect roi(90,100,250,250);
    imgCrop = img(roi);

    imshow("Image",img);
    imshow("ImageResize",imgCrop);
    waitKey(0);
    //return 0;

 }
……………………………………………………………………………………
//logo maker

#include <opencv2/imgcodecs.hpp>
#include <opencv2/highgui.hpp>
#include <opencv2/imgproc.hpp>
#include <iostream>

 using namespace cv;
 using namespace std;


 /////////////////  Images  //////////////////////

 int main() {

    //string path = "/Users/animeshkumarnayak/Downloads/Resources/
dep.jpeg";
    //VideoCapture cap(0);
    Mat img(512,512,CV_8UC3,Scalar(0,255,255));
    circle(img,Point(256,256),155,Scalar(255,255,255),FILLED);
    circle(img,Point(256,256),185,Scalar(0,0,255),2);
    rectangle(img,Point(130,226),Point(382,286),Scalar(155,235,155),FILLED);
    line(img,Point(130,296),Point(382,296),Scalar(0,0,128),3);

    putText(img, " Annie_unplugged  ", Point(137,262) ,
FONT_HERSHEY_DUPLEX,0.75, Scalar(0,0,0),2);


    //Mat imgResize, imgCrop;


    imshow("Image",img);
    //imshow("ImageResize",imgCrop);
    waitKey(0);
    //return 0;

 }

……………………………………………………………………………………

//image face detctor

#include <opencv2/imgcodecs.hpp>
#include <opencv2/objdetect.hpp>
 #include <opencv2/highgui.hpp>
 #include <opencv2/imgproc.hpp>
 #include <iostream>

 using namespace cv;
 using namespace std;


 ///////////////// Images //////////////////////

 int main() {

    string path = "/Users/animeshkumarnayak/Downloads/Resources/dep.jpeg";
    //VideoCapture cap(0);
    Mat img = imread(path);

    CascadeClassifier faceCascade;
    faceCascade.load("/Users/animeshkumarnayak/Downloads/Resources/
haarcascade_frontalface_default.xml");

    vector<Rect> faces;
    faceCascade.detectMultiScale(img, faces,1.1,10);

    for (int i =0;i<faces.size();i++)
    {
        rectangle(img, faces[i].tl(), faces[i].br(), Scalar(255,255,255),2);
    }

    imshow("Image",img);
    waitKey(0);


…………………………………………………………………………………………………
//webcam face detctor

#include <opencv2/imgcodecs.hpp>
#include <opencv2/objdetect.hpp>
 #include <opencv2/highgui.hpp>
 #include <opencv2/imgproc.hpp>
 #include <iostream>

 using namespace cv;
 using namespace std;

 int main() {

    //string path = "/Users/animeshkumarnayak/Downloads/Resources/
dep.jpeg";
    VideoCapture cap(0);
    Mat img ;
    while (true){
        cap.read(img);
        CascadeClassifier faceCascade;
        faceCascade.load("/Users/animeshkumarnayak/Downloads/Resources/
haarcascade_frontalface_default.xml");

        vector<Rect> faces;
        faceCascade.detectMultiScale(img, faces,1.1,10);

        for (int i =0;i<faces.size();i++)
        {
            rectangle(img, faces[i].tl(), faces[i].br(), Scalar(255,255,255),2);
        }

        imshow("Image",img);
        waitKey(1);
    }
 }

……………………………………
//
//webcam numberplate detctor and cropping it
```

```
#include <opencv2/imgcodecs.hpp>
#include <opencv2/highgui.hpp>
#include <opencv2/imgproc.hpp>
#include <opencv2/objdetect.hpp>
#include <iostream>

using namespace cv;
using namespace std;

////////////////  License Plate Detector ////////////////////////

int main() {

    Mat img;
    VideoCapture cap(0);

    CascadeClassifier plateCascade;
    plateCascade.load("/Users/animeshkumarnayak/Downloads/Resources/haarcascade_russian_plate_number.xml");

    //if (plateCascade.empty()) { cout << "XML file not loaded" << endl; }

    vector<Rect> plates;

    while (true) {

        cap.read(img);
        plateCascade.detectMultiScale(img, plates, 1.1, 10);

        for (int i = 0; i < plates.size(); i++)
        {
            Mat imgCrop = img(plates[i]);
            //imshow(to_string(i), imgCrop);
            imwrite("/Users/animeshkumarnayak/Downloads/Resources/Plates/" + to_string(i) + ".png", imgCrop);
            rectangle(img, plates[i].tl(), plates[i].br(), Scalar(255, 0, 255), 2);
        }

        imshow("Image", img);
        waitKey(1);
    }
}
```