**C++ Code to capture Images**

#include <opencv2/opencv.hpp>

#include <raspicam\_cv.h>

#include <iostream>

#include <chrono>

#include <ctime>

using namespace std;

using namespace cv;

using namespace raspicam;

Mat frame;

void Setup (int argc,char \*\*argv, RaspiCam\_Cv &Camera

{

Camera.set (CAP\_PROP\_FRAME\_WIDTH, (“-w”, argc, argv, 400 ) )

Camera.set (CAP\_PROP\_FRAME\_HEIGHT, (“-h”, argc, argv,240 ) );

Camera.set (CAP\_PROP\_BRIGHTNESS, (“-br”, argc,argv,50 ) );

Camera.set (CAP\_PROP\_CONTRAST, (“-co”, argc,argv,50 ) );

Camera.set (CAP\_PROP\_SATURATION, ( "-sa",argc,argv,50 ) );

Camera.set (CAP\_PROP\_GAIN, ( "-g",argc,argv ,50 ) );

Camera.set (CAP\_PROP\_FPS, ( "-fps",argc,argv,100));

}

int main(int argc,char \*\*argv)

{

RaspiCam\_Cv Camera;

Setup(argc, argv, Camera);

cout<<"Connecting to camera"<<endl;

if (!Camera.open())

{

cout<<"Failed to Connect"<<endl;

cout<<"Camera Id = "<<Camera.getId()<<endl;

while(1

{

auto start = std::chrono::system\_clock::now();

Camera.grab();

Camera.retrieve( frame);

auto end = std::chrono::system\_clock::now();

std::chrono::duration<double> elapsed\_seconds = end-start;

float t = elapsed\_seconds.count();

int FPS = 1/t;

cout<<"FPS = "<<FPS<<endl;

imshow("orignal", frame);

waitKey(1);

}

return 0;

**}**