**실습 코드 1번 (HOG 알고리즘)**

#include <iostream>

#include <opencv2/opencv.hpp>

using namespace std;

using namespace cv;

int main()

{

VideoCapture cap("C:/opencv/프로젝트 경로/walking.JPG");

if (!cap.isOpened()) {

cout << "Video Open Fail" << endl;

return - 1;

}

HOGDescriptor hog; // HOGDescriptor 객체 hog 선언

hog.setSVMDetector(HOGDescriptor::getDefaultPeopleDetector()); // 보행자 검출을 위한 훈련된 SVM 분류기 계수 등록

Mat frame;

while(1) {

cap >> frame;

if (frame.empty())

break;

vector<Rect> detected;

hog.detectMultiScale(frame, detected); // 보행자 검출 수행

for (Rect r : detected) { // 검출된 정보를 사각형으로 표시

Scalar c = Scalar(255, 0, 0);

rectangle(frame, r, c, 3);

}

imshow("frame", frame);

waitKey();

}

}

**실습 코드 2번 (HOG 알고리즘)**

#include "opencv2/opencv.hpp"

#include <iostream>

using namespace cv;

using namespace std;

int main()

{

VideoCapture cap("vtest.mp4");

if (!cap.isOpened()) {

cerr << "Video open failed!" << endl;

return -1;

}

HOGDescriptor hog;

hog.setSVMDetector(HOGDescriptor::getDefaultPeopleDetector());

Mat frame;

while (true) {

cap >> frame;

if (frame.empty())

break;

vector<Rect> detected;

hog.detectMultiScale(frame, detected);

for (Rect r : detected) {

Scalar c = Scalar(rand() % 256, rand() % 256, rand() % 256);

rectangle(frame, r, c, 3);

}

imshow("frame", frame);

if (waitKey(10) = = 27)

break;

}

return 0;

}

**실습 코드 3번 (HOG 알고리즘)**

#include "opencv2/opencv.hpp"

#include <iostream>

using namespace cv;

using namespace std;

int main()

{

VideoCapture cap("vtest.avi");

if (!cap.isOpened()) {

cerr << "Video open failed!" << endl;

return -1;

}

HOGDescriptor hog;

hog.setSVMDetector(HOGDescriptor::getDefaultPeopleDetector());

Mat frame;

while (true) {

cap >> frame;

if (frame.empty())

break;

vector<Rect> detected;

hog.detectMultiScale(frame, detected);

for (Rect r : detected) {

Scalar c = Scalar(rand() % 256, rand() % 256, rand() % 256);

rectangle(frame, r, c, 3);

}

imshow("frame", frame);

if (waitKey(10) = = 27)

break;

}

return 0;

}

**실습 코드 4번 (QR코드)**

void decode\_qrcode()

{

VideoCapture cap(0);

if (!cap.isOpened()) {

cerr << "Camera open failed!" << endl;

return;

}

QRCodeDetector detector;

Mat frame;

while (true) {

cap >> frame;

if (frame.empty()) {

cerr << "Frame load failed!" << endl;

break;

}

vector<Point> points;

String info = detector.detectAndDecode(frame, points);

if (!info.empty()) {

polylines(frame, points, true, Scalar(0, 0, 255), 2);

putText(frame, info, Point(10, 30), FONT\_HERSHEY\_DUPLEX, 1, Scalar(0, 0, 255));

}

imshow("frame", frame);

if (waitKey(1) = = 27)

break;

}

}