



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ _____ «Информатика и системы управления»

КАФЕДРА _____ «Теоретическая информатика и компьютерные технологии»

Летучка №5

по курсу «Языки и методы программирования»

**«Введение в CV на примере реализации задачи Key point detection
на C++»**

Студент группы ИУ9-22Б Лавров Р. Д.

Преподаватель Посевин Д. П.

Москва 2025

1 Задание

1.1. Определение кисти руки

2 Результаты

Листинг 1 — handPoseVideo.cpp

```
1 #include <opencv2/dnn.hpp>
2 #include <opencv2/imgproc.hpp>
3 #include <opencv2/highgui.hpp>
4 #include <iostream>
5
6 using namespace std;
7 using namespace cv;
8 using namespace cv::dnn;
9
10 const int POSE_PAIRS[20][2] =
11 {
12     {0,1}, {1,2}, {2,3}, {3,4},           // thumb
13     {0,5}, {5,6}, {6,7}, {7,8},           // index
14     {0,9}, {9,10}, {10,11}, {11,12},       // middle
15     {0,13}, {13,14}, {14,15}, {15,16},     // ring
16     {0,17}, {17,18}, {18,19}, {19,20}     // small
17 };
18
19 string protoFile = "hand/pose_deploy.prototxt";
20 string weightsFile = "hand/pose_iter_102000.caffemodel";
21
22 int nPoints = 22;
23
24 int main(int argc, char **argv)
25 {
26     float thresh = 0.01;
27
28     cv::VideoCapture cap("asl.mp4");
29
30     if (!cap.isOpened())
31     {
32         cerr << "Unable to connect to camera" << endl;
33         return 1;
34     }
35
36     Mat frame, frameCopy;
37     int frameWidth = cap.get(CAP_PROP_FRAME_WIDTH);
38     int frameHeight = cap.get(CAP_PROP_FRAME_HEIGHT);
39     float aspect_ratio = frameWidth/(float)frameHeight;
40     int inHeight = 368;
```

Листинг 2 — handPoseVideo.cpp

```

1  int inWidth = (int(aspect_ratio*inHeight) * 8) / 8;
2  cout << "inWidth = " << inWidth << " ; inHeight = " << inHeight << endl;
3  VideoWriter video("Output-Skeleton.avi",VideoWriter::fourcc('M','J','P','G'), 10, Size(
4  frameWidth,frameHeight));
5  Net net = readNetFromCaffe(protoFile, weightsFile);
6  double t=0;
7  while(1) {
8      double t = (double) cv::getTickCount();
9      cap >> frame;
10     frameCopy = frame.clone();
11     Mat inpBlob = blobFromImage(frame, 1.0 / 255, Size(inWidth, inHeight), Scalar(0, 0, 0),
12     false, false);
13     net.setInput(inpBlob);
14     Mat output = net.forward();
15     int H = output.size[2];
16     int W = output.size[3];
17     vector<Point> points(nPoints);
18     for (int n=0; n < nPoints; n++) {
19         Mat probMap(H, W, CV_32F, output.ptr(0,n));
20         resize(probMap, probMap, Size(frameWidth, frameHeight));
21         Point maxLoc;
22         double prob;
23         minMaxLoc(probMap, 0, &prob, 0, &maxLoc);
24         if (prob > thresh) {
25             circle (frameCopy, cv::Point((int)maxLoc.x, (int)maxLoc.y), 8, Scalar(0,255,255),
26             -1);
27             cv::putText(frameCopy, cv::format("%d", n), cv::Point((int)maxLoc.x, (int)
28             maxLoc.y), cv::FONT_HERSHEY_COMPLEX, 1, cv::Scalar(0, 0, 255), 2);
29         }
30         points[n] = maxLoc;
31     }
32     int nPairs = sizeof(POSE_PAIRS)/sizeof(POSE_PAIRS[0]);
33     for (int n = 0; n < nPairs; n++) {
34         Point2f partA = points[POSE_PAIRS[n][0]];
35         Point2f partB = points[POSE_PAIRS[n][1]];
36         if (partA.x<=0 || partA.y<=0 || partB.x<=0 || partB.y<=0)
37             continue;
38
39         line (frame, partA, partB, Scalar(0,255,255), 8);
40         circle (frame, partA, 8, Scalar(0,0,255), -1);
41         circle (frame, partB, 8, Scalar(0,0,255), -1);
42     }
43     t = ((double)cv::getTickCount() - t)/cv::getTickFrequency();
44     cout << "Time Taken for frame = " << t << endl;
45     cv::putText(frame, cv::format("time taken = %.2f sec", t), cv::Point(50, 50), cv::
46     FONT_HERSHEY_COMPLEX, .8, cv::Scalar(255, 50, 0), 2);
47     imshow("Output-Skeleton", frame);
48     video.write(frame);
49     char key = waitKey(1);
50     if (key==27)
51         break;
52 }
53 // When everything done, release the video capture and write object
54 cap.release();
55 video.release();
56 return 0;
57 }

```



3 Вывод

Благодаря этой летучке, я продолжил познакомиться с библиотекой OpenCV и её возможностями.