



PENALTY TRACKER

Выполнили:
Медведев Никита
Федотов Антон

ГЛОБАЛЬНАЯ ЗАДАЧА

Написать программу, которая:

- Выделяет мяч, игроков и ворота из видеофайла с пенальти
- Рисует попадания в ворота
- Подсчитывает число голов

ЧТО ПОЛУЧИЛОСЬ

Имеется программа, которая:

- Находит нужные кадры с воротами
- Выделяет ворота
- Рисует попадания и экспортирует карту попаданий по воротам

НАХОЖДЕНИЕ КАДРА С ВОРОТАМИ

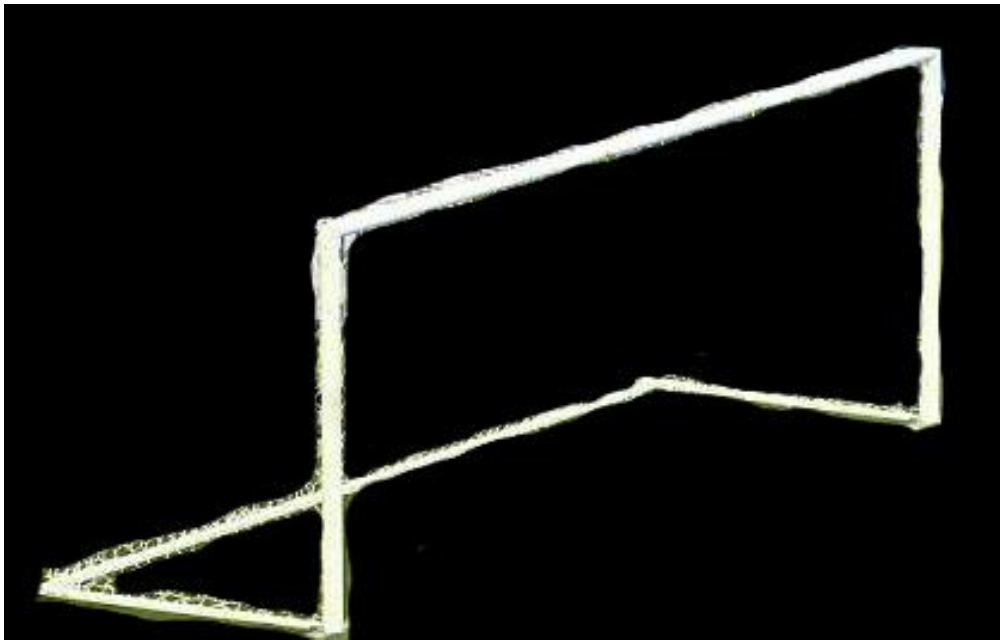


```
double Tracker_Custom::search_frame(const Mat& frame)
{
    int vbins = 30, sbins = 8;
    int histSize[] = { vbins, sbins };
    float vranges[] = { 0, 256 };
    float sranges[] = { 0, 256 };
    const float* ranges[] = { vranges, sranges };
    MatND hist_f, hist_test;
    Mat hsv_frame;
    int channels[] = { 2, 1 };

    calcHist(&test_picture, 1, channels, Mat(), hist_test, 2,
histSize, ranges, true, false);

    cvtColor(frame, hsv_frame, COLOR_BGR2HSV);
    calcHist(&hsv_frame, 1, channels, Mat(), hist_f, 2, histSize,
ranges, true, false);
    double hist_diff = compareHist(hist_f, hist_test, 5);
    //cout << hist_diff << endl;
    return hist_diff;
}
```

ВЫДЕЛЕНИЕ ВОРОТ



```
cvtColor(frame, gray_frame, COLOR_BGR2GRAY);
```

```
threshold(gray_frame, gray_frame, 230, 255,  
THRESH_BINARY);
```

```
matchTemplate(gray_frame, gray_goal_picture,  
goal_selected_frame, TM_CCOEFF);
```

РЕЗУЛЬТАТ

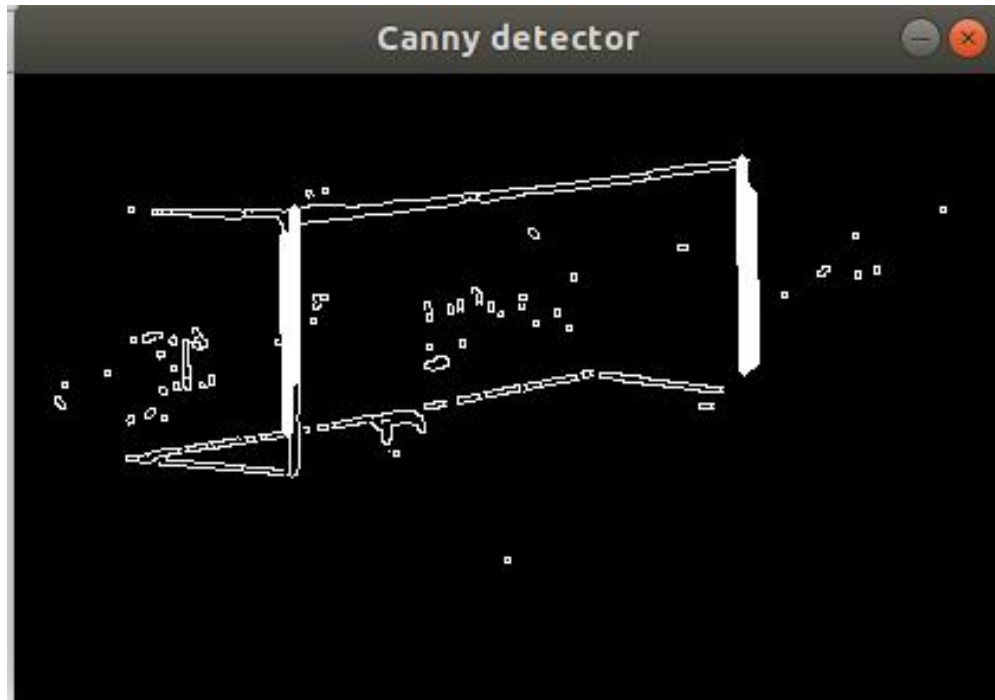
Правильный



Ошибочный

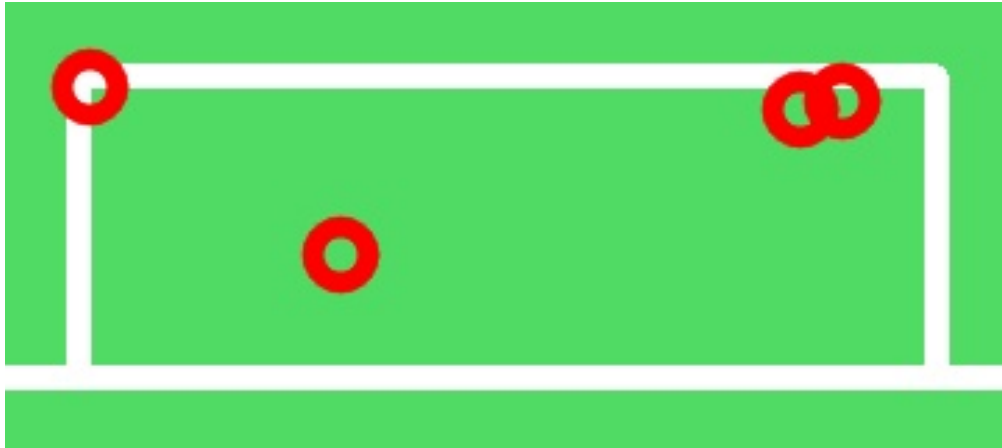


ПРИМЕНЕНИЕ АЛГОРИТМОВ КАННИ И ХАФА



```
Canny(goal_frame, canny_image, 10, 300, 7);  
  
vector<Vec4i> lines;  
  
HoughLinesP(canny_image, lines, 1, CV_PI/180, 50, 50, 10);  
  
for(size_t i = 0; i < lines.size(); i++)  
{  
    Vec4i l = lines[i];  
    if((l[2]-l[0]) < 5 && (l[2]-l[0]) > -5)  
    {  
        line(canny_image, Point(l[0], l[1]), Point(l[2], l[3]),  
            Scalar(255,255,255), 3);  
    }  
}
```

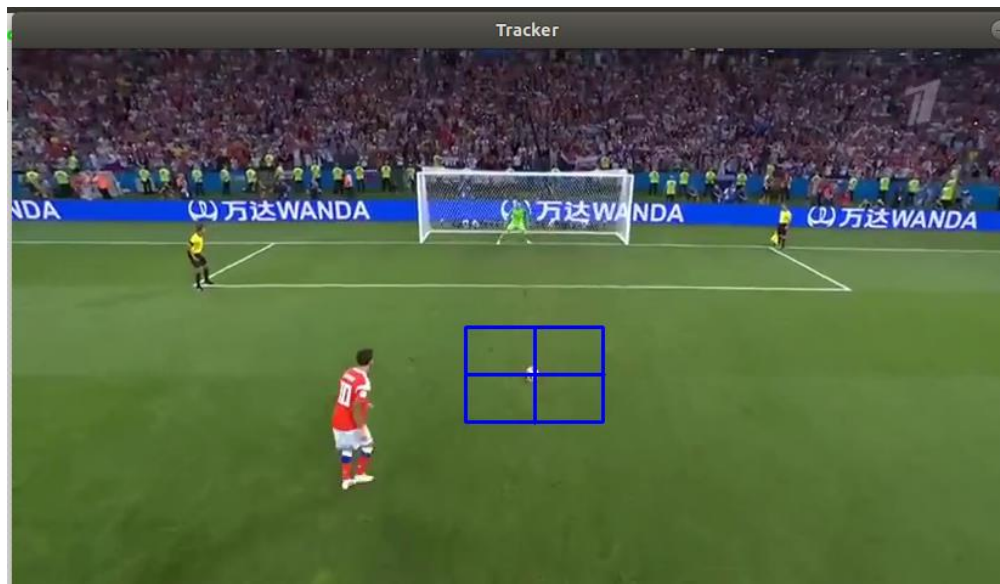
КАРТА ПОПАДАНИЙ



```
void Tracker_Custom::draw_strikes_map(vector<Point> strikes_coords) {  
    for(size_t i = 0; i < strikes_coords.size(); i++)  
    {  
        Point coord = strikes_coords[i];  
  
        coord.x = coord.x * 3 + 30;  
  
        coord.y = coord.y + 30; // from left upper corner, if from lower corner =  
100 - coord.y + 30  
  
        cout << coord.x << "    " << coord.y << endl;  
  
        circle(strikes_map, coord, 10, Scalar(0,0,255), 6, 16);  
    }  
}  
  
void Tracker_Custom::save_strikes_map()  
{  
    imwrite("Strikes_map.jpg", strikes_map);  
}
```


ТРЕКИНГ МЯЧА

Выделяем мяч



Рамка переходит на другой объект или трекинг прекращается (в зависимости от алгоритма)

