



Целью данной работы является сравнение скорости распознавания объектов с использованием библиотеки OpenCV на CPU и GPU.

Для этого задания воспользуемся классификаторами, предоставляемыми с библиотекой OpenCV (данные классификаторы находятся в каталоге /imageVideo):

- 1) haarcascade\_mcs\_eyeye.xml
- 2) haarcascade\_mcs\_nose.xml

*ROS: Для проверки содержимого bag-файла используется следующая команда:*

```
rosviz info <bagfile>
```

*ROS: Для воспроизведения содержимого bag-файла используется следующая команда:*

```
rosviz play <bagfile>
```

*ROS: Для получения полного списка тем, которые в настоящее время опубликованы в работающей системе используется следующая команда:*

```
rostopic list -v
```

Задание (Работа с видеопотоком):

Разработать 2 модуля ROS, подписывающийся на topic полученный из bag-файла, находящегося в каталоге /imageVideo. Каждый модуль должен использовать свой классификатор. Результатом работы программы является сравнение времени распознавания с использованием CPU и GPU.

Получение объекта cv::Mat из cv\_bridge::CvImagePtr:

```
#include <sensor_msgs/image_encodings.h>
```

```
cv_bridge::CvImagePtr img_ptr = cv_bridge::toCvCopy(msg.sensor_msgs::image_encodings::BGR8);  
cv::Mat img = img_ptr->image;
```

Подготовка изображения для классификатора:

```
cv::Mat img_gray;  
cv::cvtColor(img, img_gray, CV_BGR2GRAY);  
cv::equalizeHist(img_gray, img_gray);
```

Загрузка изображение на GPU:

```
cv::gpu::GpuMat img_gray_gpu;  
  
img_gray_gpu.upload(img_gray);
```

Выбор каскада

```
std::string cascade_name = "haarcascade_mcs_righteye.xml";  
  
cv::CascadeClassifier cascade;  
cv::gpu::CascadeClassifier_GPU cascade_gpu;  
  
face_cascade.load(cascade_name);  
face_cascade_gpu.load(cascade_name);
```

Работа с классификатором:

На CPU:

```
std::vector<cv::Rect> objs;  
cascade.detectMultiScale( img_gray, objs, 1.1, 1, 0|CV_HAAR_SCALE_IMAGE, cv::Size(50, 50) );
```

На GPU:

```
cv::gpu::GpuMat objbuf;  
cascade_gpu.detectMultiScale( img_gray_gpu, objbuf, 1.1, 1, cv::Size(50, 50));
```

Визуализация работы классификатора:

На CPU:

```
for(size_t i=0; i<objs.size(); i++) {  
    cv::rectangle(img_gray, objs[i], cv::Scalar(0, 0, 255), 2, 8, 0);  
}  
cv::imshow("view", img_gray);
```

На GPU:

```
cv::Mat obj_host;  
objbuf.colRange(0,detect).download(obj_host);  
cv::Rect *fc = obj_host.ptr<cv::Rect>();  
  
for(int i=0; i<detect; i++) {  
    cv::rectangle(img_gray, fc[i], cv::Scalar(255));  
}  
cv::imshow("view", img_gray);
```