

# ОТСЛЕЖИВАНИЕ ОБЪЕКТОВ: АДАПТАЦИЯ МЕТОДА ДЛЯ РАСПРЕДЕЛЁННОЙ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ

Жданеня О. А.

Кафедра электронных вычислительных средств,  
Белорусский государственный университет информатики и радиоэлектроники  
Минск, Республика Беларусь  
E-mail: olegzhdaneniya@gmail.com

*Традиционно алгоритмы компьютерного зрения рассматриваются как единый процесс. Однако, распределённые варианты вызывают интерес, так как могут быть эффективнее. Задача отслеживания объектов существует уже несколько десятилетий и продолжает изучаться и развиваться. Имеется разные методы её решения. В данной работе рассматривается адаптация метода отслеживания нескольких объектов с применением YOLOv4 и DeepSORT для распределённой вычислительной системы.*

## ВВЕДЕНИЕ

Будем рассматривать сеть с определённым количеством камер. Стоит задача в режиме реального времени обрабатывать видеопотоки с этих камер, отслеживая объекты на них. Для отслеживания нескольких объектов в данной работе применяется метод основанный на использовании YOLOv4 и DeepSORT[1]: YOLOv4[2] выполняет роль детектора, позволяющего обнаружить объекты на изображении; DeepSORT[3] является трекером, отслеживающим перемещения нескольких объектов.

В традиционном подходе метод рассматривается как единое целое: имеется один видеопоток, который обрабатывается с помощью одной реализации метода с YOLOv4 и DeepSORT.

В данной работе рассматривается попытка адаптации метода для распределённой вычислительной системы с целью минимизации времени обработки видеопотока, а также затрачиваемых вычислительных ресурсов.

## I. ВАРИАНТЫ РАСПРЕДЕЛЁННОЙ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ

Рассматриваемый метод отслеживания объектов включает две части: детектор и трекер. Детектор занимается обнаружением объектов на изображении, вне зависимости от видеопотока, с которого обрабатываются кадры. Для трекера наоборот, для корректной работы алгоритма важно обрабатывать кадры только из одного видеопотока в хронологической последовательности. Согласно поставленным условиям предполагается два варианта распределённой вычислительной системы для дальнейшего анализа.

В первом варианте для каждого видеопотока создаётся подсистема с детектором и трекером. То есть количество подсистем обработки видеопотоков будет равно количеству камер в сети.

Во втором варианте создаётся одна подсистема с детектором, которая будет обрабатывать несколько видеопотоков, и для каждого видео

потока создаётся подсистема с трекером. Для большого количества обрабатываемых потоков необходимо будет создавать несколько подсистем с детектором.

## II. СРАВНЕНИЕ ВАРИАНТОВ РАСПРЕДЕЛЁННОЙ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ

Для проверки работы предложенных вариантов распределённых вычислительных систем используются тестовые данные из MOT16[4]. Каждый вариант проверяется на одновременной обработке одного, двух и трёх видеопотоков с одинаковыми данными. Критерием оптимальности в данной работе является минимальное время обработки видеопотока или же максимальная скорость обработки кадров.

Во время эксперимента было измерено среднее время обработки видеопотоков, содержащих одинаковые данные(см. рис. 1).

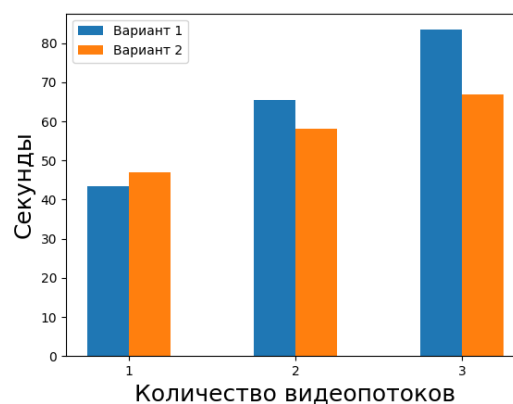


Рис. 1 – Среднее время обработки видеопотоков

Также для каждого видеопотока была измерена скорость обработки кадров(FPS)(см. рис. 2).

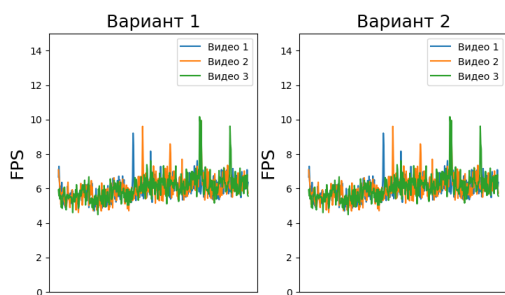


Рис. 2 – Скорость обработки кадров

В ходе экспериментов были получены данные об использовании вычислительных ресурсов: использование центрального процессора (см. рис. 3), использование графического процессора (см. рис. 4), использование ОЗУ (см. рис. 5), использование графической памяти (см. рис. 6). Данные были получены при обработке трёх видеопотоков.

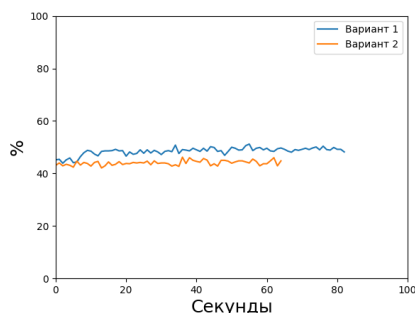


Рис. 3 – Загруженность центрального процессора

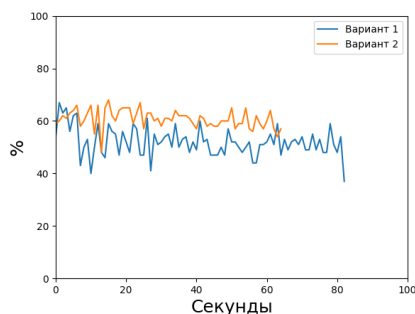


Рис. 4 – Загруженность графического процессора

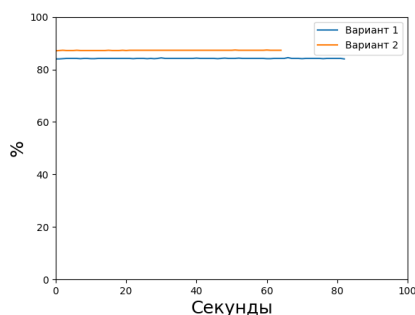


Рис. 5 – Использование ОЗУ

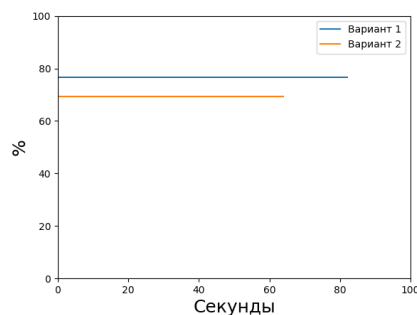


Рис. 6 – Использование графической памяти

### III. ВЫВОДЫ

Исходя из полученных в эксперименте данных, наиболее оптимальным по времени обработки видеопотока является второй вариант системы, при обработке нескольких видеопотоков. Также он требует меньше ресурсов центрального процессора и графической памяти по сравнению с первым вариантом. Однако, при увеличении количества обрабатываемых видеопотоков, использование только одной подсистемы с детектором будет замедлять работу всей системы. Для сохранения производительности системы необходимо использовать несколько подсистем с детектором. Первый вариант системы оптимален по времени при обработке одного видео потока.

### IV. СПИСОК ЛИТЕРАТУРЫ

1. yolov4-deepsort [Electronic resource] / J. Wotherspoon. – 2020. – Mode of access: <https://www.github.com/theAIGuysCode/yolov4-deepsort/>. – Date of access: 10.10.2022.
2. Bochkovskiy, A. YOLOv4: Optimal Speed and Accuracy of Object Detection / A. Bochkovskiy, Chien-Yao Wang, Hong-Yuan Mark Liao // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). – 2021. – P. 13029–13038.
3. Wojke, N. Simple Online and Realtime Tracking with a Deep Association Metric / N. Wojke, A. Bewley, D. Paulus // IEEE International Conference on Image Processing. – 2017. – P. 3645–3649.
4. Multiple Object Tracking Benchmark MOT16 [Electronic resource] / A. Milan, L. Leal-Taixé, I. Reid, S. Roth, K. Schindler – 2016. – Mode of access: <https://motchallenge.net/data/MOT16/>. – Date of access: 10.10.2022.