

Пример дешевого использования современных технологий.

1. Берем дешевое оборудование от HikVision под любым суббрендом - стоимость 2 камер с кабелями - около 5 тыс



только не забываем что HikVision обычно идет с бэкдором в комплекте, так что не желательно чтобы в инет торчало (что впрочем всегда помогает).

2. Настраиваем сохранение потока с камеры на писк или сервер, где будет происходить обработка:

```
ffmpeg -i "rtsp://192.168.1.10:554/user=admin&password=&channel=3&stream=0.sdp?" -c copy -flags +global_header -f segment -segment_time 1800 -segment_format_options movflags=+faststart -reset_timestamps 1 channel_3_%d.mp4 > log_stream
```

команда сохраняет получасовые видеосики с соответствующего канала/камеры.

Раньше надо было мудохаться с каким нибудь zonerfinder теперь же полно нормально работающих программных решений позволяющих работать с потоками видео, что называется из коробки – например выделение участков на видео с движением в отдельные отрывки

```
dvr-scan --threshold 2 -tb 1s -tp 1s -l 3s --kernel-size -1 -bb -i filename -m opencv -d detectedevents > log_scan
```

команда сохраняет моменты, где в видео было движение (+ настройки шумов и продолжительности) в отдельные файлы в папку detectedevents

3. Анализ – берем самый простой и популярный ultralytics - уже вышла в прод yolo8 , но yolo5 тоже норм работает

```
#git clone https://github.com/ultralytics/yolov5.git
```

```
#cd yolov5
#pip install -r requirements.txt
python detect.py --source ./detectedevents/202405261225.mp4 --save-txt --save-csv
```

раскраска видоса подписями с вероятностью совпадения и выгрузкой в виде уже готовом для загрузки в базу данных.

В таблички вида – дата время, канал , координаты на изображении, классифицированный объект (вероятность) .



```
202405261225.mp4, person, 0.59
202405261225.mp4, person, 0.37
202405261225.mp4, person, 0.66
202405261225.mp4, person, 0.30
202405261225.mp4, person, 0.38
202405261225.mp4, person, 0.71
202405261225.mp4, person, 0.80
202405261225.mp4, person, 0.44
202405261225.mp4, dog, 0.48
202405261225.mp4, person, 0.80
202405261225.mp4, person, 0.50
202405261225.mp4, dog, 0.54
202405261225.mp4, person, 0.76
202405261225.mp4, person, 0.58
202405261225.mp4, dog, 0.65
202405261225.mp4, person, 0.78
202405261225.mp4, person, 0.31
202405261225.mp4, dog, 0.33
202405261225.mp4, person, 0.76
202405261225.mp4, dog, 0.54
202405261225.mp4, person, 0.75
202405261225.mp4, person, 0.26
202405261225.mp4, dog, 0.70
```

```
--More-- (7%)
```

Yolov уже подготовлен для загрузки в FPGA кстати - Келдыш (https://library.keldysh.ru/prep_vw.asp?pid=9612)

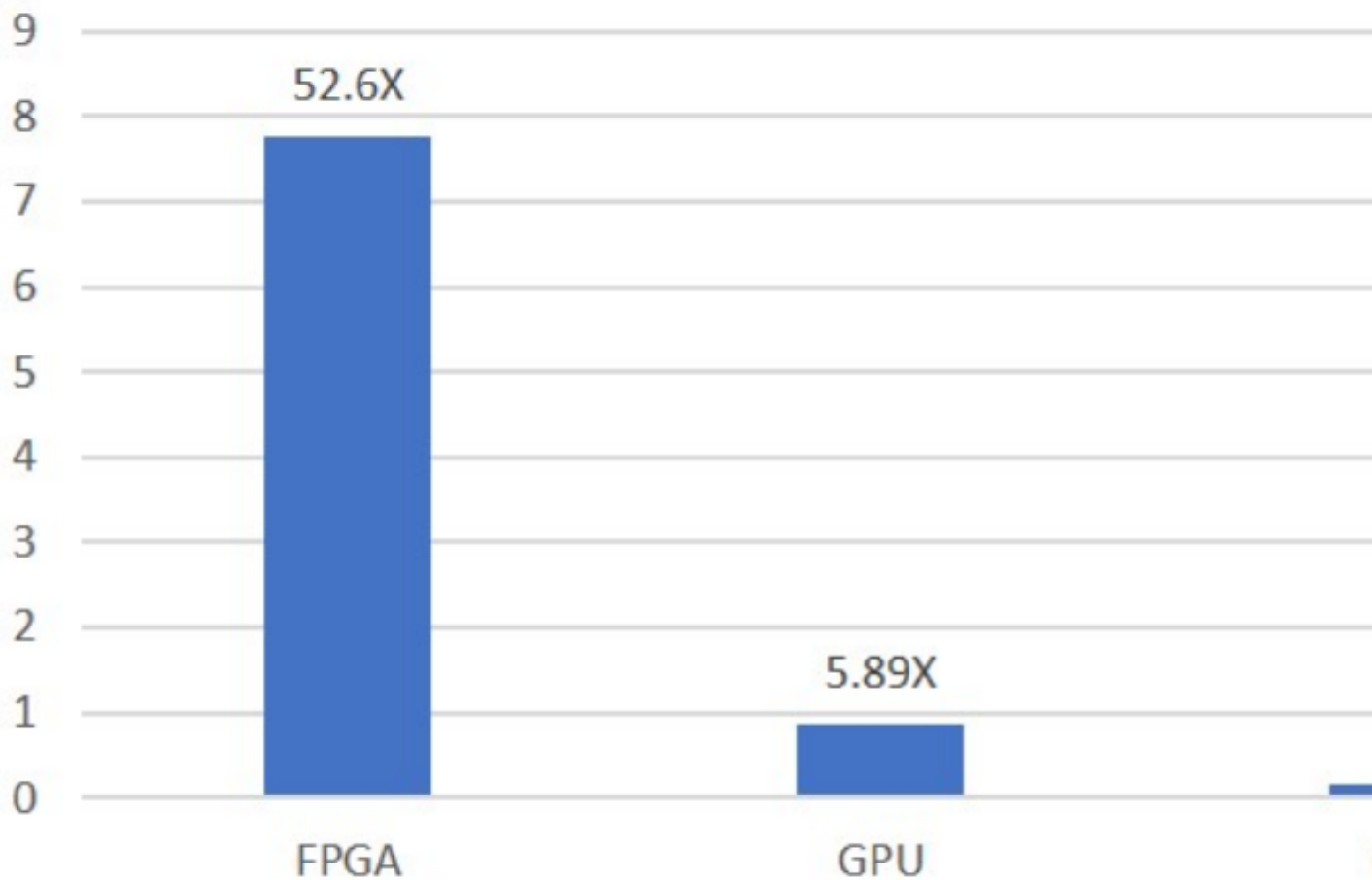
для yolov2 - https://github.com/markcxli/FPGA_DPU

и прч.

FPGA в 50 энергоэффективней GPU

Power efficiency

Power efficiency for CPU, GPU and FPGA



Power efficiency = speed/power, where GPU is **5.89** times better than CPU

User Stories:

- Navigation for Robots
- Surveillance
- Self-Driving cars Use YOLOv2 algorithm

А еще можно это же все записать на железо современного "персонального переносного следящего устройства", т.е. мобильника-дебильника. Все оптимизировать под и прч.

А можно и сразу условно говоря в "железо", чтобы работало быстро и не ело батарейку, а еще можно попробовать сделать бизнес на продаже интегрированных решений для, как пыталось сделать нвидиа - <https://www.nvidia.com/ru-ru/autonomous-machines/embedded-systems/> , не очень пока летает судя по продажам, тем более конкуренты не спят...

Но сегодня не про технические детали, сегодня о другом.

Очевидно, что можно навесить любую метамодель на обработанные данные – доработать – у автомобилей выделять номера (и сразу по ним делать запросы к API к другим бгг системам), человечков сохранять в виде отдельных изображений с подгрузкой к виртуальным профилям... (так работает интернет слежка со стороны типа коммисов например, когда по поведению выделяют пользователей ты-тупа или вероятно уже и телесрама, если вдруг кто без симкарты зарегистрился).

Если речь про "оффлайн" - дополнительный поток для "обогащения" – с bluetooth и wifi детекторов – поток событий, соответственно если человек прошел несколько раз перед и специфично исказив "электромагнитное" поле на районе то, можно уже составить соответствующую картинку.

После заливки всего добра в БД современные аналитические системы позволяют отвечать на вопросы вроде – кто ходит по утрам на районе и каким маршрутом.

Или покажи всех залетных в этой локации.

С включением по соответствующему триггеру, соответствующих "скриптов" и это далеко не всегда "продажа" какой нибудь ненужной ботвы.

Систем на рынке работающих в таком режиме уже полным полно - не только причем в "информационном" сегменте, но начали появляться решения вокруг правового поля или "индустриального", и Россия одна из лидеров, есть уже и стандарты проектирования мета систем, увязкой с цифровыми двойниками, повышение надежности одновременно с простотой проектирования за счет "модульности" и прч... соответственно дело только в исходных наборах... и что гораздо важнее речь совсем не про "арифмометры", а про **мета модели**, которые работают уже с агрегированными данными.

Вот тут начинается самое интересное – системы из "модель" – "поток данных" - "триггер" – "реакция" существовали и до появления цифры как таковой.

И реально ржак - сейчас на фоне этого "прогресса" стали легко обнаруживаться условно говоря "Т-800" (в т.ч. самособранные) – когда всех загнали в телесрам с ты-тупом (до того как, в другие платформы хе хе) – даже собранный на коленке анализатор, но сканирующий всю поляну, моментально выявляет странные аномалии в шаблонах поведения или генерируемых "словесных" потоках отдельных биоединиц, а также механику срабатывания "триггеров" на, по которым уже соответственно восстанавливаются и "архитектуры с принципами устройства", работающих в мета пространстве самих Систем (и анти-систем, бгг).

Следующий уровень это конечно разбор принципов координации и возникновения самих Систем и там уже механики не так много кстати - работают концептуальные моменты.

Для затравки

Паск создал группу автоматов, которые в изменяющейся среде должны были выработать целесообразное поведение обеспечивающее им сохранение устойчивости — «выживание». При этом выживали только те автоматы, которые способны были эффективно находить «пищу». И вот оказалось что автоматы самостоятельно выработали для себя совершенно новые формы поведения. В процессе эксперимента они объединялись в группы, в которых каждый член действовал не только в своих интересах, но и в интересах группы в целом. Причем это свойство не было предварительно запрограммировано. Автоматы вырабатывали его сами.

Опыты по моделированию эволюционного развития на вычислительной машине «Киев», проведенные под руководством В. Глушкова, позволили отразить другую — негативную — сторону групповых явлений. Так, например, уменьшение количества пищи для каждого автомата, при котором автомат еще «не умирает», ниже 13 условных единиц (и для машин это оказалось роковым!) произвело губительное действие на всю группу автоматов — она прекратила свое существование. По-видимому, сыграло роль то обстоятельство, что при взаимодействии между автоматами условия оказались слишком жесткими: «индивидуумы» стали мешать друг другу, и группа в целом не успевала организовать устойчивую структуру до своей гибели. К аналогичному результату привело и увеличение «возраста размножения» автоматов до 21 единицы.

Групповой эффект, выражающийся в «вымирании» при уменьшении максимально допустимого для системы «голода», то есть количества операций, в течение которых автомат успевает целесообразно перестроиться, имеет своим аналогом в обществе, например, информативный голод.

Одним из распространенных групповых явлений современного мира, общества, разделенного на две социальные сис-