

Трекинг нескольких человек на нескольких камерах

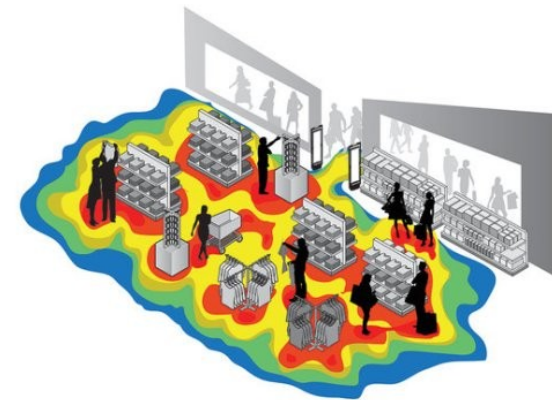
2019

Студент: Арсений Рылов,
Ментор: Юстина Иванова



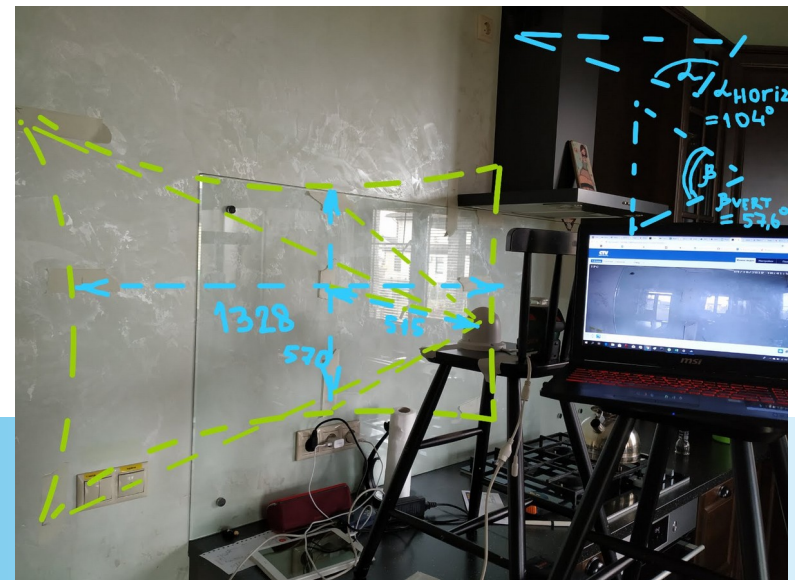
Постановка задачи

- Необходимо отследить перемещения каждого человека по магазину самообслуживания от момента входа до завершения визита. В реальном времени.
- Трекинг – часть более общей задачи: сверки того, что взял человек по видео и того, что прошло по кассе.
- Дальнейшие применения – отслеживание очередей, мест остановки, изучение предпочтений и индивидуальные предложения, heat map на плане



Постановка задачи

- Дано:
- Два магазина – 6 и 36 широкоугольных камер. Камеры, чаще всего, перекрывают друг друга. Слепых зон нет.
- В крупном магазине – более 100 визитов в день
- Одновременно в магазине не более 15 человек



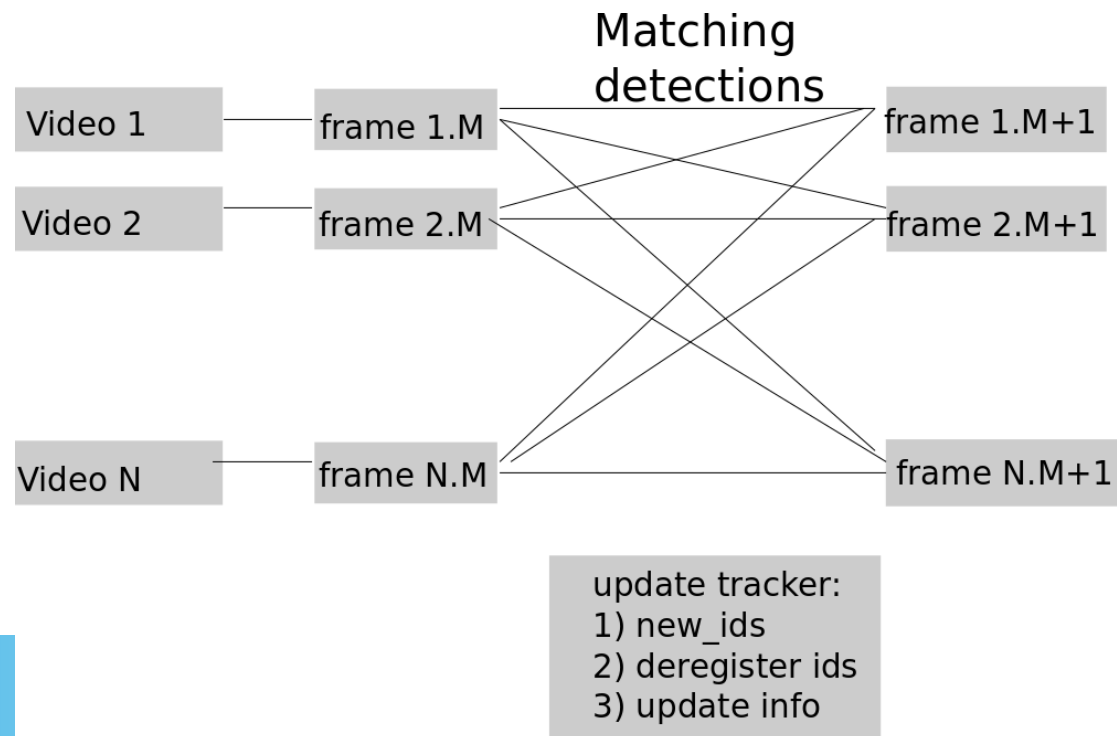
Метрики

- Метрика Accuracy = TP/N $17/35=48\%$
- Ее аналог – Rough accuracy = $1 - \frac{abs(IDs_nmb-N)}{N}$, $1-(51-35)/35=54\%$

IDs_nmb – число объектов созданных трекером.

План решения:

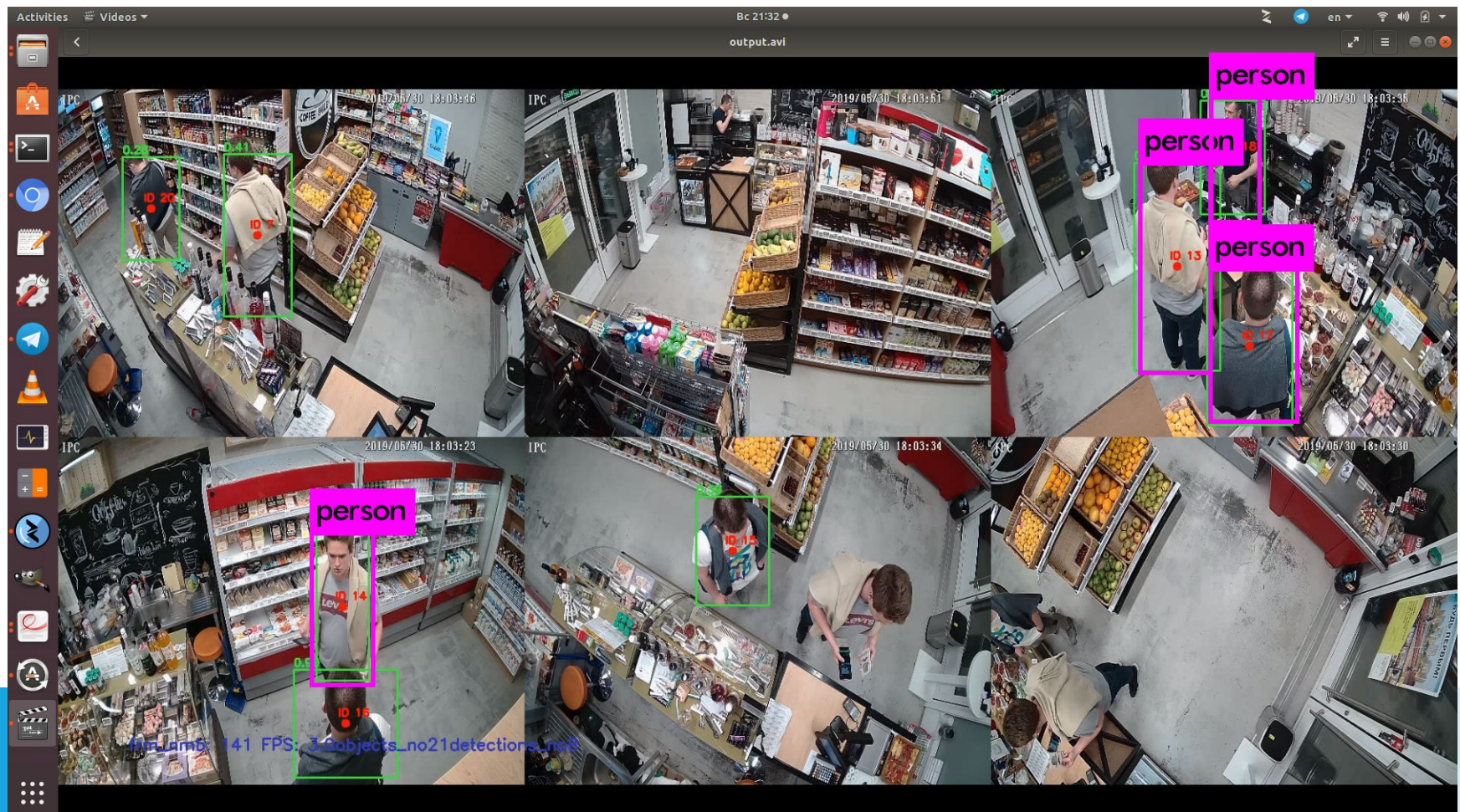
- Детекция. Инициализация трекера
- Извлечение фичей для мэтчинга
- Сравнение детекций между соседними кадрами и обновление трекера



https://github.com/zamarseny/homework_python/blob/master/CV/Diploma/YOLO_and_multi_tracking.ipynb

Проблемы

- Люди плохо детектируются на видах сверху предобученными моделями



Проблемы

- Разброс углов установки камер в вертикальной плоскости. От 0 до 60 градусов к вертикали
- Люди очень по-разному выглядят с разных ракурсов
- Перекрывание детекций
- Влияние фона на фичи
- Отражения на стеклянных поверхностях
- Ограниченность ресурсов: две нейросети изначально не хотелось

Преппроцессинг

- Синхронизация видеопотоков с камер
- Выравнивание освещенности
- Выбор оптимального разрешения для YOLO

Детекция

- YOLO3
- ~~MobileNet~~
- ~~faster_rcnn_~~
- Resnet50
- ~~mask_rcnn_~~
- inception

Детекция

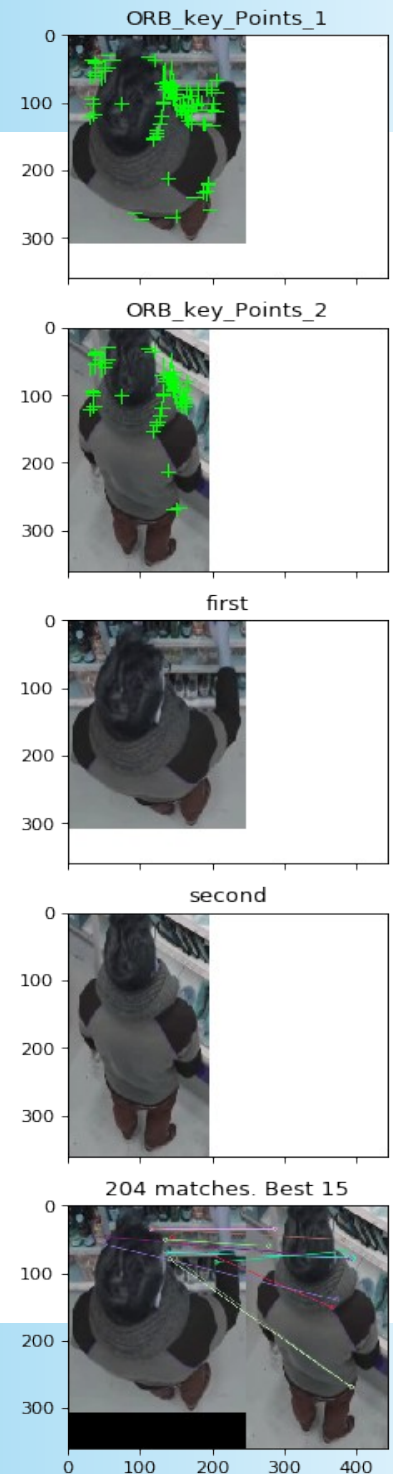
Performance on the COCO Dataset

Model	mAP	FLOPS	FPS
SSD300	41.2 -		46
SSD500	46.5 -		19
SSD321	45.4 -		16
DSSD321	46.1 -		12
R-FCN	51.9 -		12
SSD513	50.4 -		8
DSSD513	53.3 -		6
FPN FRCN	59.1 -		6
Retinanet-50-500	50.9 -		14
Retinanet-101-500	53.1 -		11
Retinanet-101-800	57.5 -		5
YOLOv3-320	51.5	38.97 Bn	45
YOLOv3-416	55.3	65.86 Bn	35
YOLOv3-608	57.9	140.69 Bn	20
YOLOv3-spp	60.6	141.45 Bn	20

Фи́чи

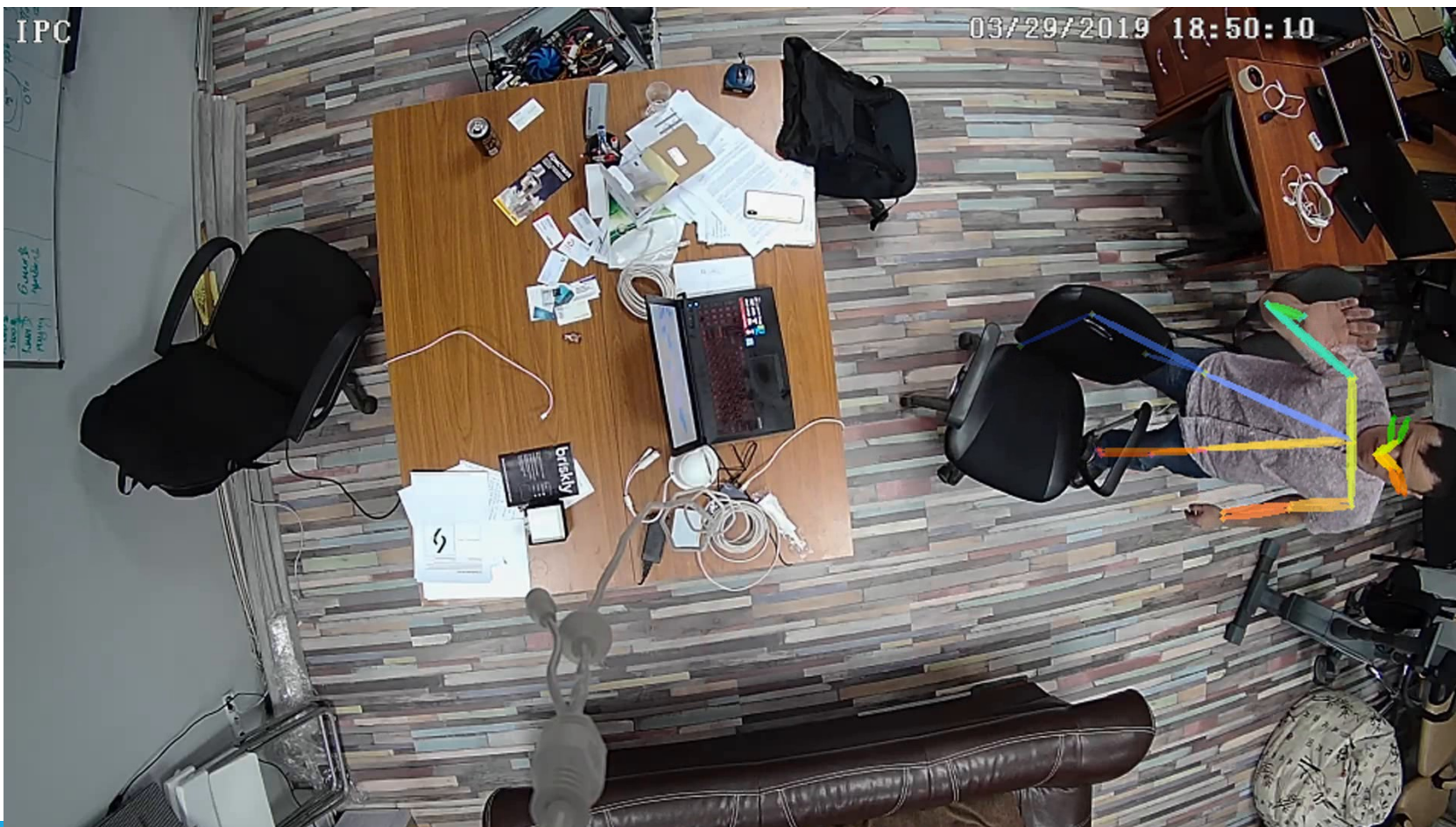
- Координаты детекций
- ~~Средний цвет по трем каналам~~
- ~~Стандартные отклонения цвета~~
- ~~Гистограммы цветов~~
- ~~ORB~~

Мера близости – евклидово
расстояние



Фи́чи

- Цвет каждой конечности



Выводы и заключения

- Для решения проблемы перекрывающихся детекций нужно обучать детектор и реидентификацию на своих данных.
- Нужно сосредоточиться на разметке данных

Следующие шаги

- Натренировать детектор на виды сверху
- Учесть близость камер, скорость человека при мэтчинге
- Применить person reidentification покадрово и к локальным трекам



СПИСОК ИСТОЧНИКОВ

- <https://www.pyimagesearch.com/2018/07/23/simple-object-tracking-with-opencv/>
- People Detection in Fish-eye Top-views. Meltem Demirkus, Ling Wang, Michael Eschey, Herbert Kaestle and Fabio Galasso
- Intelligent multi-camera video surveillance: A review. 2012
Xiaogang Wang
- Unsupervised Person Re-identification by Deep Learning
Tracklet Association
https://www.youtube.com/watch?v=-ljltl_arX8
- Person Re-identification: Past, Present and Future. Liang Zheng, Yi Yang, and Alexander G. Hauptmann