**Короткий опис розробленого функціоналу.**

Розроблена система визначає чи є на фото об’єкт, і якщо він є, то знаходить його на фото й замінює на зображення. У випадку роботи з відео можлива заміна знайденого об’єкта на кадри з іншого відео. За необхідності можливе швидке перетворення програми у повноцінний VR додаток( портувавши її на Андроїд або ж шляхом підключення до ip-камери( або веб-камери).

**Порядок обробки кожного окремого фото.**

1. Застосовуємо дескриптор AKAZE до зображення, отримуємо набір особливих точок.
2. За допомогою наперед навченого кластеризатора визначаємо клас кожної особливої точки.
3. Будеємо частотну таблицю класів зображення.
4. Класифікатор аналізує частотну таблицю й дає відповідь чи є на фото об’єкт.
5. Якщо об’єкт присутній виконується зіставлення особливих точок нашого зображення й наперед визначеного взірцевого.
6. На основі отриманого зіставлення будується відображення еталонного зображення на досліджуване.
7. Відображення застосовується до рамки еталонного зображення й таким чином визначається контур нашого об’єкта.
8. Отримане відображення застосовується до зображення-жертви, що має такі ж розміри як і еталонне. Усі точки образу, що потрапили у знайдений контур накладаються поверх початкового зображення.

**Порядок обробки відео.**

1. П 1-4 застосовуються до кожного кадру, поки не буде виявлено наш об’єкт.
2. Для наступного кадру виконуємо П 5-8.
3. Виконуємо П.2 доки можлива побудова відображення та площа виявленого контуру не дуже сильно відрізняється від площі контуру, виявленого на попередньому кроці.
4. Якщо П3. Не виконується переходимо до П1.

**Технічні особливості.**

При накладанні відео на відео результат роботи зберігається, але відео не відтворюється(пише що файл пошкоджено).

Навчена нами пара дескриптор-класифікатор працює не набагато швидше ніж виконується побудова відображення і виконується саме відображення( однак у випадку більш швидкого й точного класифікатора таким чином можна отримати більш суттєвий виграш у швидкодії).

Під’єднання до камери телефона як до ip-камери робить наш виріб більш VR-ним, але частота передачі кадрів далека до online.

Під’єднання до web-камери ноутбука «вбиває VR», а якість зображення не дуже висока.

**Опис роботи пари дескприптор-класифікатор.**

Враховуючи досвід попередньої роботи було вирішено використовувати пару дескриптор AKAZE – класифікатор Random Forest. До цього нами було розглянуто три дескриптори, найбільш швидким виявився ORB, проте він давав доволі неоднозначні результати. З іншого боку, SIFT при ретельному підборі параметрів давав найбільш якісні результати, проте страждала швидкодія. Саме тому з точки зору цих особливостей було обрано дескриптор AKAZE , що дає якість, подібну до SIFT, проте швидший за нього. Класифікатор Random Forest показав адекватні результати при збільшенні порогу класифікації, проте ми відчуваємо, що наш класифікатор не є найкращим вибором з точки зору точності.

Зібраний нами датасет не дуже великий, тому ми як і раніше застосовуємо підхід BOVW.

**Особливості роботи з відео (детальні).**

Експерименти з заміни зображення нашого предмету без класифікації показали, що сам факт успішної побудови відображення означає досить високу ймовірність знаходження предмета у кадрі.

Перехід між двома станами дасть виграш у часі у випадку якісного та швидкого класифікатора.

Перший кадр на якому виявлено об’єкт ми пропускаємо, задля плавності роботи програми( й цілком природньо очікувати його наявність на наступних кадрах).

Оскільки об’єкт зникає з кадру поступово, то досить ймовірно що класифікатор перестане його бачити до його зникнення, в той час як побудова відображення все ще можлива.

Також в нас є евристика, що дуже швидко дозволяє визначати коректність побудованого відображення: ми порівнюємо площу контуру на цьому й минулому кадрі. У більшості випадків розміри об’єкта в кадрі змінюються не дуже швидко, тому великі відхилення – скоріш за все, ознака помилки. Також при правильному відображення зберігається порядок обходу кутів рамки, тому ми визначаємо площу як півдобуток діагоналей(чи того, що має ними бути) та синусу кута між ними.

Також можна перевіряти на перетин образи діагоналей, але нам трошки лінь))).

**Швидкодія отриманої системи.**

Тести, проведені для накладання відео, показали швидкодію близько 2 – 3.5 кадрів на секунду. Варто зазначити, на основі результатів можна стверджувати, що левову частку часу, необхідного для процесу обробки кадру чи зображення є отримання дескриптора. Також у цей час входить час зчитування кадру чи зображення, проте, з практики це займає порівняно менше часу. Це видно, наприклад, при порівнянні кадрів, на які накладалися зображення та не накладалися. Дані про швидкодію наведено у файлі results.csv.

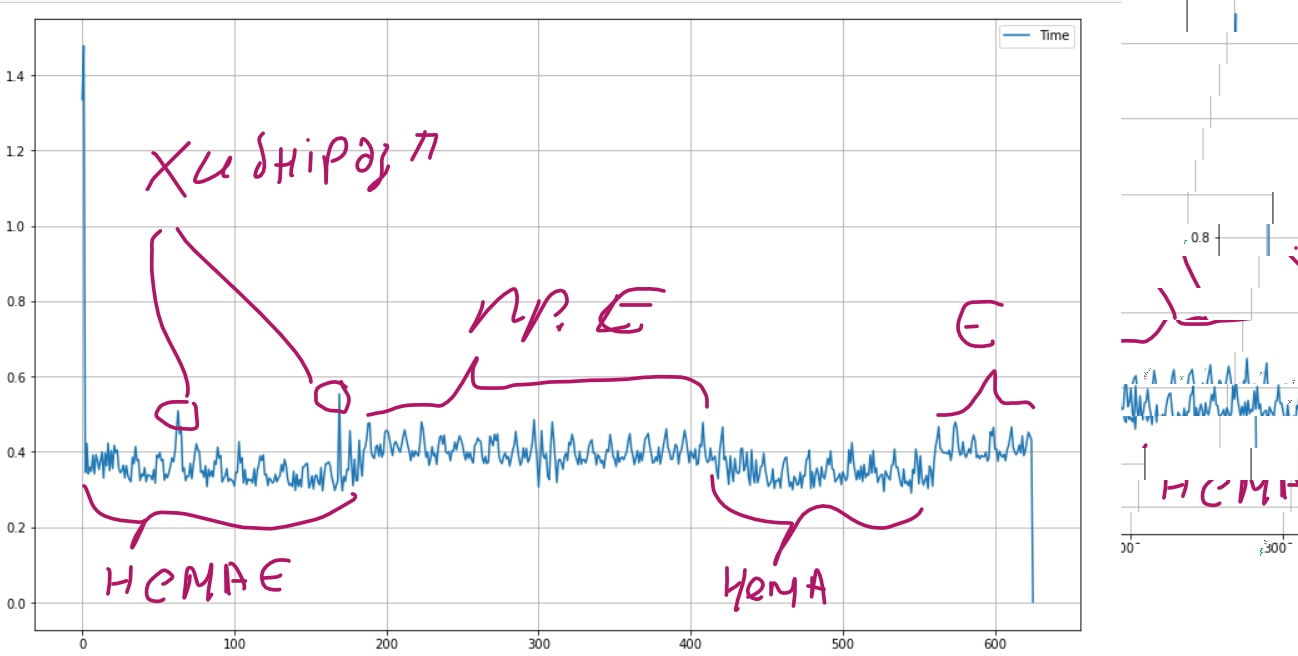


Рис. Графік швидкодії

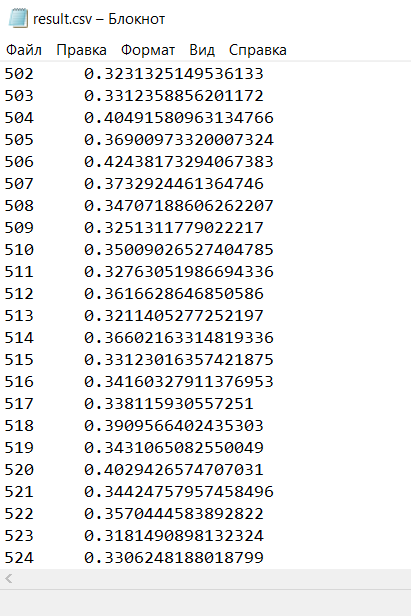


Рис. Частина даних про швидкодію

**Висновок.**

В ході експериментів виявилось, що BOVW дуже важко застосовувати на дуже різних між собою фото( навіть наявність килима на фоні змушує класифікатор й кластеризатор біситися), а ось досить для обробки одноманітних зображень (наприклад рентгенівських знімків якоїсь однієї частини тіла) цей підхід цілком прийнятний. Побудувати онлайновий застосунок виявилось складніше, ніж ми очікували. Пара класифікатор – кластеризатор виявилась достатньо повільною, тому варто було б використовувати замість Python якусь більш «швидку» мову. За якістю було отримано цілком прийнятні результати, багато в чому завдяки особливостям дескриптора AKAZE.