**Remote Heart Rate Detector**

**Дистанционный детектор пульса**

**Документ составил студент НИТУ МИСиС**

**Кущ Артем**

**БПМ-18-1**

Оглавление

[Обработка видео 3](#_Toc55048848)

[Поиск лиц 3](#_Toc55048849)

[Поиск лиц на CPU 3](#_Toc55048850)

[Поиск лиц на GPU 3](#_Toc55048851)

[Выбор области (ROI) 3](#_Toc55048852)

[Понижение шумов видеоизображения 3](#_Toc55048853)

[Пирамида и фильтр Гаусса 3](#_Toc55048854)

[Преобразование Фурье для 3-ех мерного тензора 4](#_Toc55048855)

[Amplify filter 5](#_Toc55048856)

[Алгоритм 5](#_Toc55048857)

[Источники 6](#_Toc55048858)

# Обработка видео

## Поиск лиц

Ранее в курсовой работе я подробно рассматривал различные подходы для поиска лица на фото: Elastic Graph Matching, CMM/HMM, PCA, AMM/ASM, CNN-архитектуры.

### Поиск лиц на CPU

За основу была выбрана модель типа Elastic Graph Matching – фильтры Хаара. Модель обладает хорошей точностью и производительностью. Основная проблема заключается в непостоянстве размера рамки поиска для каждого кадра одного видео.

Для решения этой проблемы необходимо обработать моделью все кадры, найти центр для каждого, усреднить размер рамки и с помощью медианного фильтра избавиться от выбросов координат центра (ошибочное определения лица).

### Поиск лиц на GPU

За основу был выбран ансамбль CNN-архитектур – FastMTCNN-pytorch. Эффективная, облегченная и улучшенная архитектура MTCNN. На CPU работает в несколько раз медленнее каскадов Хаара. Некоторые сравнения из статьи автора по скорости обработки изображений на GPU - P100:

|  |  |  |  |
| --- | --- | --- | --- |
| Package | FPS (1080x1920) | FPS (720x1280) | FPS (540x960) |
| facenet-pytorch | 12.97 | 20.32 | 25.50 |
| facenet-pytorch (non-batched) | 9.75 | 14.81 | 19.68 |
| dlib | 3.80 | 8.39 | 14.53 |
| mtcnn | 3.04 | 5.70 | 8.23 |

## Выбор области (ROI)

Было выбрано 4 основных области:

* исходное вырезанное лицо;
* лоб;
* левая щека;
* правая щека.

## Понижение шумов видеоизображения

### Пирамида и фильтр Гаусса

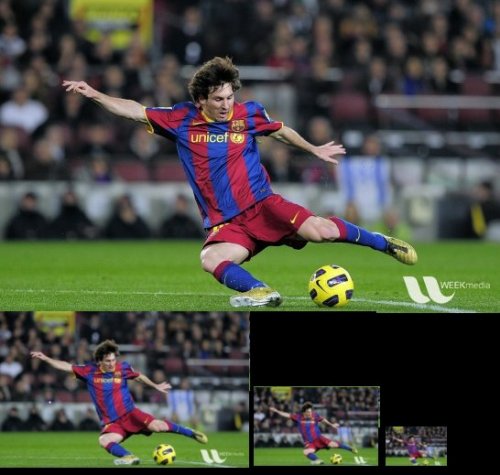
При покадровом извлечении видео все элементы одинакового размера. Но после применения алгоритмов и нейросетей для выявления лиц на кадрах зачастую приходится работать с изображениями разного разрешения.

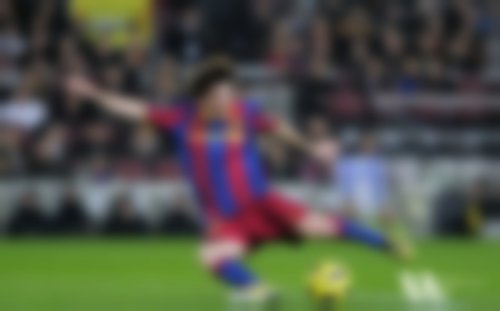
В этом случае нам нужно будет создать набор изображений с различным разрешением и искать объект во всех элементах. Эти наборы изображений с разным разрешением называются пирамидами изображений (потому что, когда они хранятся в стопке с самым большим изображением внизу и самым маленьким изображением вверху, они выглядят как пирамида).

Существует два вида пирамид изображений:

1. пирамида Гаусса
2. пирамида Лапласа

Более высокий уровень (низкое разрешение) в Гауссовой пирамиде формируется путем удаления последовательных строк и столбцов в изображении более низкого уровня (более высокое разрешение). Затем каждый пиксель на более высоком уровне формируется вкладом от 5 пикселей на нижележащем уровне с гауссовыми весами. Таким образом, изображение . становится изображением . Таким образом, площадь уменьшается до одной четвертой от первоначальной площади. Это называется Octave. Аналогично, когда мы поднимаемся в пирамиде (т. е. разрешение уменьшается).

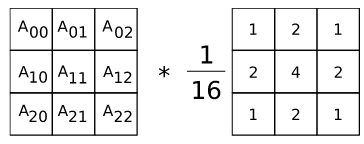
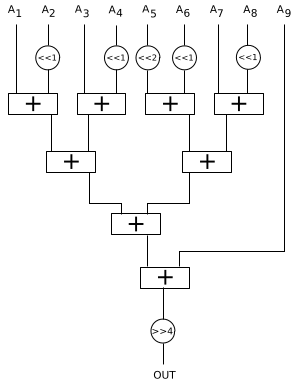




Фильтр Гаусса — электронный фильтр, чьей импульсной переходной функцией является функция Гаусса.

Фильтр Гаусса, как и медианный фильтр, используется для устранения шума в кадре, однако у него есть и побочный эффект — размытие изображения.

Пример для матрицы 3x3:



### Преобразование Фурье для 3-ех мерного тензора

Преобразование Фурье (символ ℱ) — операция, сопоставляющая одной функции вещественной переменной другую функцию вещественной переменной. Эта новая функция описывает коэффициенты («амплитуды») при разложении исходной функции на элементарные составляющие — гармонические колебания с разными частотами.

Преобразование Фурье функции вещественной переменной является интегральным и задаётся следующей формулой:

Применяя данную формулу для 4-ех мерного тензора (набор кадров) по нулевой оси, можем получить частоты необходимых гармонических колебаний. Все сигналы, с частотой меньше 50 и больше 175 необходимо занулить. Данный диапазон соответствует пульсу человека в покое и после силовых упражнений. Обратное преобразование Фурье даст необходимый результат. Если буфер слишком мал, зачастую почти все пиксели кадра будут 0.

### Amplify filter

Усиление результата обработки фильтром Фурье. По умолчанию коэффициент домножения равен 30, но рекомендуется изменять данный параметр вручную в зависимости от освещенности и прочих факторов, таких как частые повороты головой и автобаланс белого.

## Алгоритм

# Источники

* <https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_pyramids/py_pyramids.html>
* <https://ru.wikipedia.org/wiki/Фильтр_Гаусса>
* <https://ru.wikipedia.org/wiki/Признаки_Хаара>
* <https://ru.wikipedia.org/wiki/Преобразование_Фурье>