## Описание проблемы

Как правило, кино снимают в таком соотношении сторон, в котором ширина кадра больше высоты, потому что глаза человека расположены горизонтально по отношению друг к другу. Будем называть такие видео горизонтальными.

Мы в Иви показываем кино на разных устройствах, в том числе и мобильных телефонах. А телефоном удобнее пользоваться, когда держишь его вертикально, то есть ширина экрана меньше высоты.

Если показать горизонтальное видео на вертикальном экране, останется много неиспользуемого места, и масштаб будет сильно меньше, чем мог бы быть.

Чтобы смотреть видео на телефонах было более приятно, мы хотим научиться автоматически превращать горизонтальные видео в вертикальные.

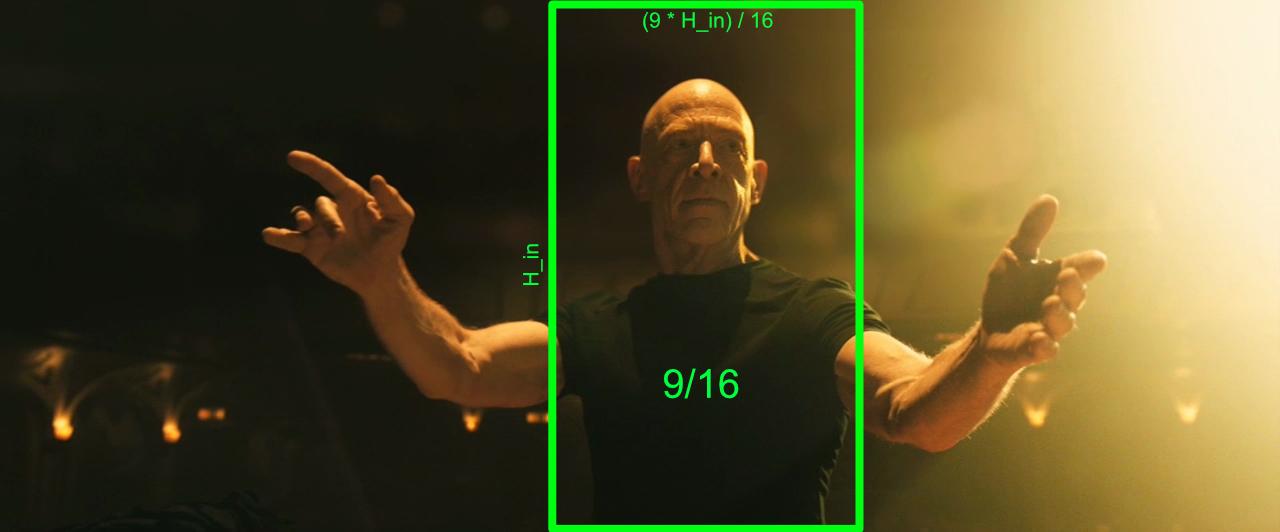
## Что потребуется сделать?

Вам нужно написать код на языке Python, который принимает на вход пути к видео-файлу и его метаданным, а на выходе для каждого кадра видео отдаёт координату кропа, который превратит входное горизонтальное видео в вертикальное.

**Координата кропа** — расстояние от левого края кадра до левой границы кропа в пикселях.

Полученный кроп должен иметь соотношение сторон 9/16, а размер его для каждого видео всегда будет один и тот же.

Если исходное видео имеет ширину **W\_in** и высоту **H\_in**, то видео, которое получится в результате кропа должно иметь такую же высоту, как и у исходного, но ширина его должна быть **9 \* H\_in / 16**.



Вместе с кодом вам нужно приложить скрипт, который собирает Docker-образ со всеми необходимыми зависимостями, внутри которого можно вызвать скрипт вертикализации и получить результат, и инструкцию, как это всё билдить и запускать.

## Критерии хорошей вертикализации

* Кроп должен всё самое интересное, что происходит сейчас в кадре
* Кроп не должен создавать эффект тряски или экшн-камеры, закреплённой на теле персонажей
* Если в кадре есть персонаж, его лицо должно попадать в кроп
* Если в кадре несколько персонажей, то приоритет такой: 1) персонаж, который в данный момент разговаривает, 2) персонаж, который находится выше в списке актёров, 3) все остальные персонажи

Эти критерии справедливы для большинства кропов. Но так же есть и исключения. В валидационном датасете они есть.

## Данные

Это шоты из 20 фильмов. Шот — это отрезок видео от одной монтажной склейки до другой.

Для каждого фильма даны список актёров в порядке убывания их значимости в фильме.

Названия фильмов и их сопоставления шотам не даны.

Для каждого кадра каждого шота видео даны метаданные:

* координаты лиц в кадре
* ID актёра, которому принадлежит каждое лицо лицо
* для каждого лица флаг, говорит ли в данный момент персонаж или нет,
* ID трека лица

Для валидационной подвыборки шотов так же даны координаты кропов, которые подготовила команда Иви. На них можно валидироваться.

Все остальные видео — неразмеченные. Участники могут доразметить данные самостоятельно, если посчитают нужным.

## Структура папок и файлов

В папке *shot\_videos* хранятся видео-файлы в формате \*.mp4. Их имена до точки соответствуют именам фич в папках *unlabelled\_features* и *validation\_set*.

**Формат файла с ID актёров actors\_lists.json:**

[

[151,5,74,175,48,27,153,130,70],

[154,138,64,55,243,79,...,12],

...

[1,2,3,4,..,10],

]

Здесь один вложенный список соответствует списку актёров одного фильма. Чем меньше индекс ID актёра, тем важнее этот актёр в касте фильма.

**Формат файла с незазмеченными фичами из папки unlabelled\_features:**

[

[

{

"track": 0,

"score": -0.800000011920929,

"x1": 306.6274108886719,

"y1": 108.55032348632812,

"x2": 405.6669921875,

"y2": 207.58990478515625,

"actor\_id": -1

}

],

…,

[

{

"track": 0,

"score": -0.8750000149011612,

"x1": 306.6034355163574,

"y1": 108.55078125,

"x2": 405.69096755981445,

"y2": 207.63831329345703,

"actor\_id": -1

}

]

]

Тут каждый элемент массива первого уровня соответствует одному кадру.

А каждый элемент массива второго уровня — лицу, найденному в этом кадре. Лиц в кадре может быть несколько.

* *track* — это ID трека лиц. То есть лица с одинаковым ID трека скорее всего принадлежат одному и тому же человеку.
* *score* – это скор детектора речевой активности. Если он меньше 0, значит персонаж молчит, если больше — говорит.
* *x1, y1, x2, y2* – координаты левого верхнего и правого нижнего углов бокса, который очерчивает лицо
* *actor\_id* – ID актёра из файла actors\_lists.json. Если там значение -1, значит актёра не удалось распознать.

**Формат файлов из папки validation\_set почти такой же.**

[

{

"x\_crop": 440.95,

"faces": [

{

"track": 0,

"score": -0.5999999940395355,

"x1": 497.5221748352051,

"y1": 103.68914031982422,

"x2": 587.8559684753418,

"y2": 194.02293395996094,

"actor\_id": 180

}

]

},

{

"x\_crop": 440.95,

"faces": [

{

"track": 0,

"score": -0.5249999985098839,

"x1": 497.55559158325195,

"y1": 103.68914031982422,

"x2": 587.8893852233887,

"y2": 194.02293395996094,

"actor\_id": 180

}

]

},

Здесь добавляется координата кропа: *x\_crop* – это расстояние от левого края кадра до левого края кропа.

Данные по лицам же тут лежат в поле *faces*.

## Критерии оценки

У нас есть тестовый набор видео, которые мы не покажем участникам.

После окончания хакатона, мы обработаем эти видео скриптами учатсников, и оценим по следующим критериям.

**Субъективные:** Жюри экспертов из Иви отсмотрит получившиеся кропа и выставит им оценки от 1 до 5 по критериям из пункта **«Критерии хорошей вертикализации»**

**Оценка по метрике:** мы посчитаем метрику MSE для координат кропов. Истинными координатами будем считать координаты кропов, которые подготовила редакция, а предсказанными те, которе получены с помощью алгоритмов участников.

Победителя может быть два: один по субъективной оценке, а другой по метрике MSE. Но не обязательно. Это обсуждаемо.