

VISIONHACK

Классификация типов событий на записях с видеорегистраторов

Команда DeepMIPT (2-е место)

Place	Team	Date/time of the last attempt	Score on the last attempt					
			«Bridge»	«City»	«Road bump»	«Screen wipers»	«Zebra»	Total
🏆 1	DoubleA Team (Lomonosov Moscow State University) ◦ Andrey Belyaev ◦ Alexander Gromov ◦ Konstantin Sofiyuk	13.09.2017 15:25	94.7096	54.3503	74.8842	96.875	83.8442	404.663
🏆 2	Deep MIPT (Moscow Institute of Physics and Technology) ◦ Artur Fattahov ◦ Artyom Kupriyanov ◦ Ahmedhan Shabanov ◦ Kirill Tushin	13.09.2017 16:06	93.0556	0	43.6709	95.3125	92.848	324.887
🏆 3	GMLvision (Lomonosov Moscow State University) ◦ Ilia Petrov ◦ Sergey Dukanov ◦ Vladimir Guzov	13.09.2017 14:10	84.2929	3.84903	62.1951	93.75	76.0217	320.109

VISIONHACK

- Отборочный этап

Основной этап:

- 2 дня в МИСиС
- Призовой фонд — 35к \$
- Участвовать могут студенты и аспиранты
- Команды от 2 до 5 человек
- Разрешается использовать только свои компьютеры



Данные

- Видеоролики 10с, Full HD, 30 fps
- Trainset - 200 роликов
- Validationset - 200 роликов (Public Leaderboard)
- Testset - 400 роликов (был выдан за 4 часа до конца)

Train

Validation

Test

Задача

Детектировать наличие на видео следующие типы событий:

- Проезд пешеходного перехода
- Въезд под мост или в тоннель
- Включение дворников
- Въезд / выезд из города
- Наезд на лежащий полицейский









Д. СЕНИНО

~~Д. СЕНИНО~~



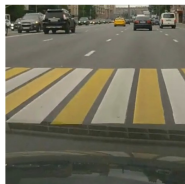
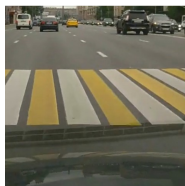
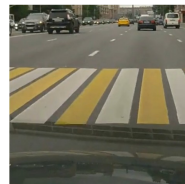
Оценка качества

- $100 / (\text{реальное количество объектов}) * TP -$
 $-1000 / (\text{реальное количество объектов}) * FP$
- Ложное срабатывание сильно штрафуетя!
- За каждую задачу можно было получить от -1000 до 100 баллов
- Итоговый результат — сумма баллов за все задачи

Зебра

- Fine - tune Resnet50
- Разметили моменты начала и конца видимости зебры
- Random crop из нижней половины картинки
- Resize до 224 x 224
- Аугментация горизонтальным отражением

Обучение

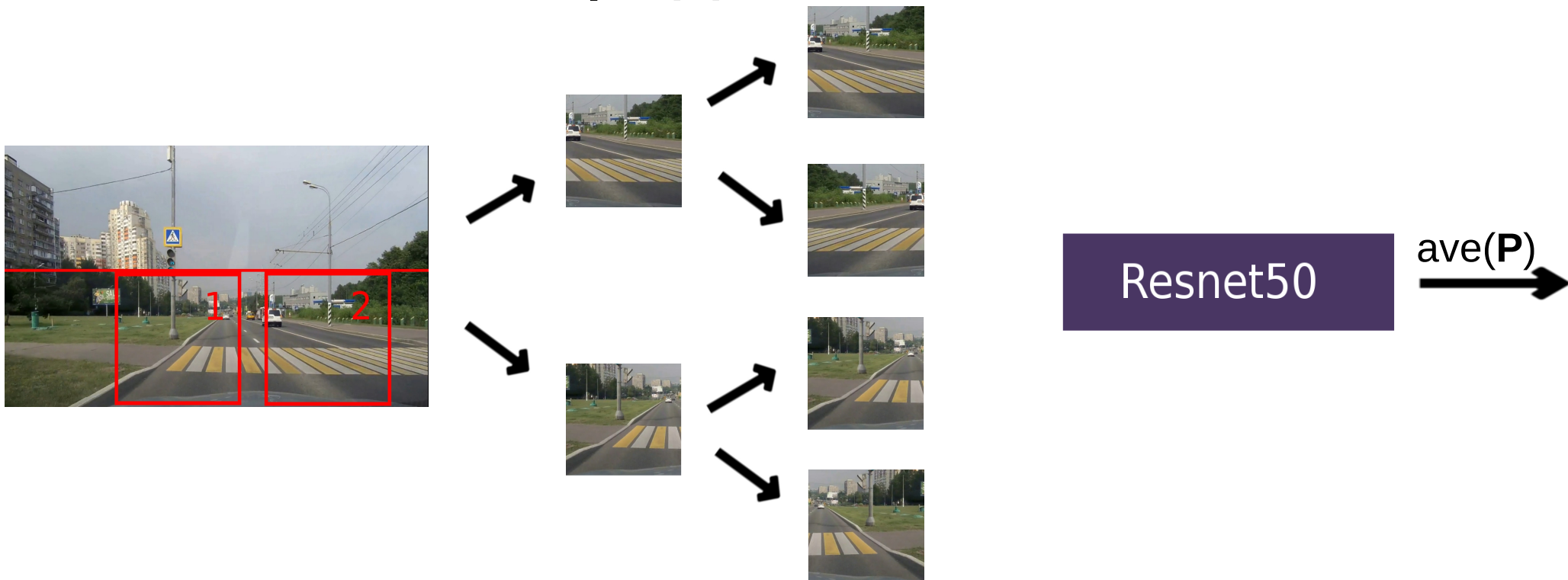


Resnet50



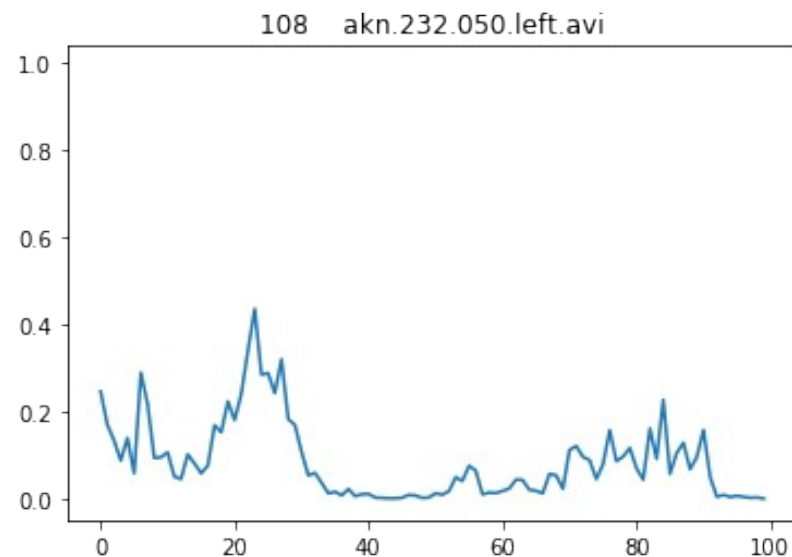
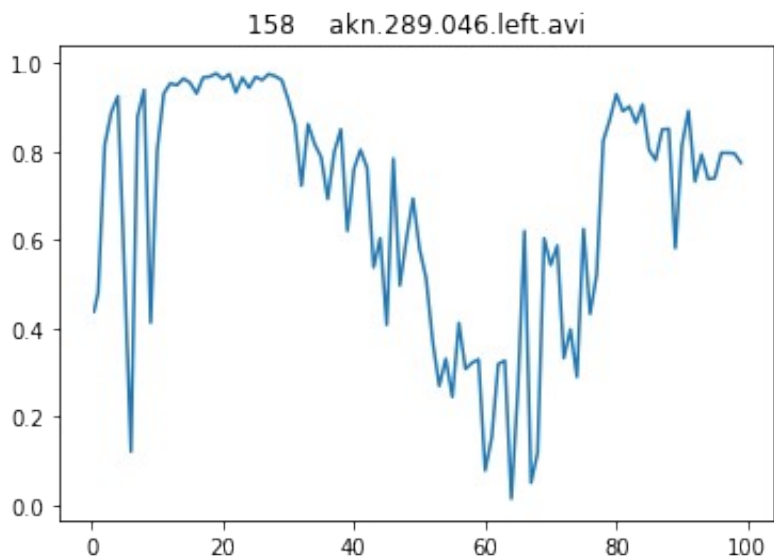
$P(\text{есть зебра})$

Предсказание



ТТА — усреднение предсказаний с четырех картинок, полученных случайными кропами и флипами

Предсказание



По оси x откладывается каждый третий кадр, по оси y — вероятность того, что на этом кадре есть зебра

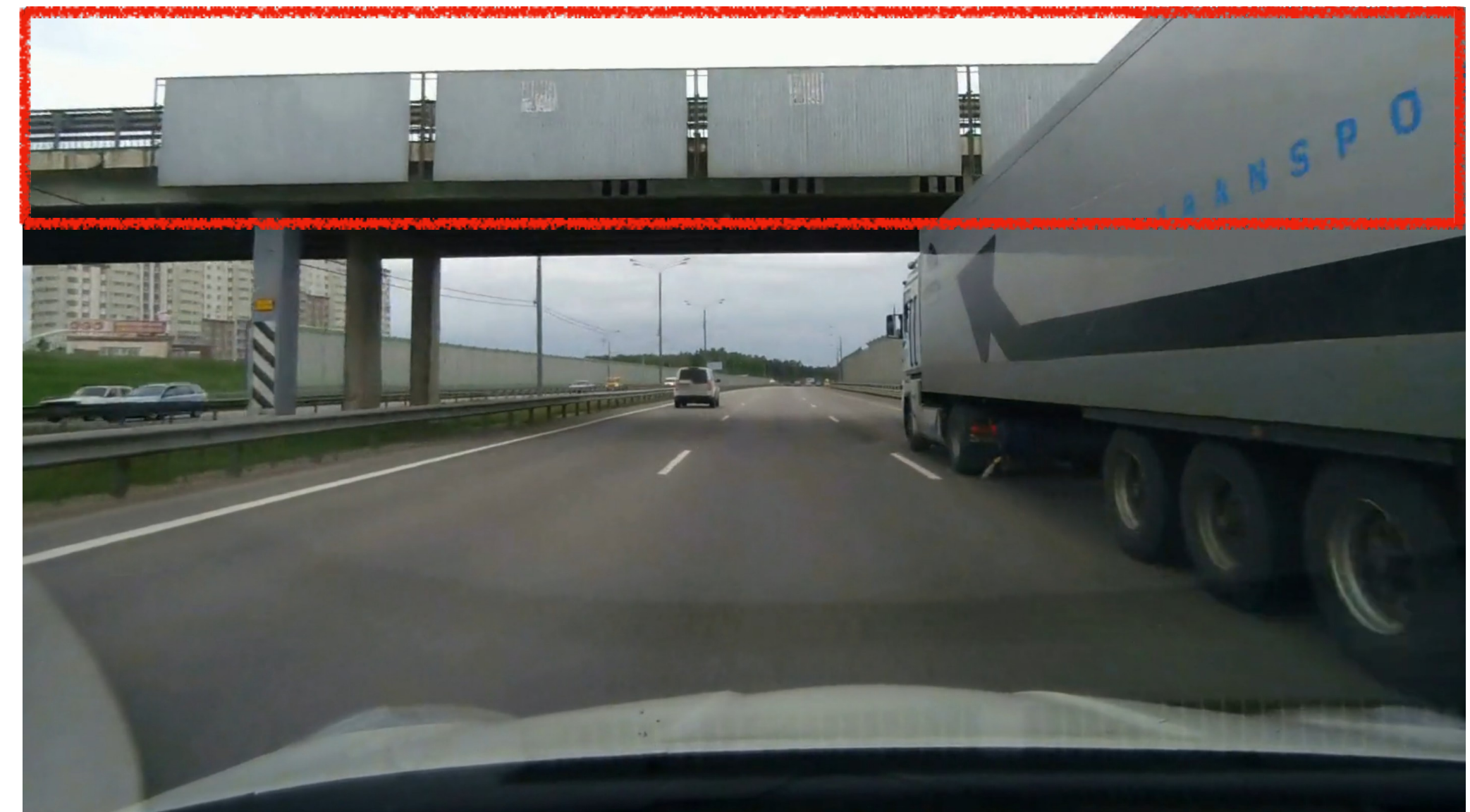
Что делать с вероятностями?

- Отсекать по порогу: максимум или среднее
- Посчитать статистики: max, min, mean, std, квантили
и подать их на вход другой модели машинного обучения

Детекция заезда в туннель

Начало:

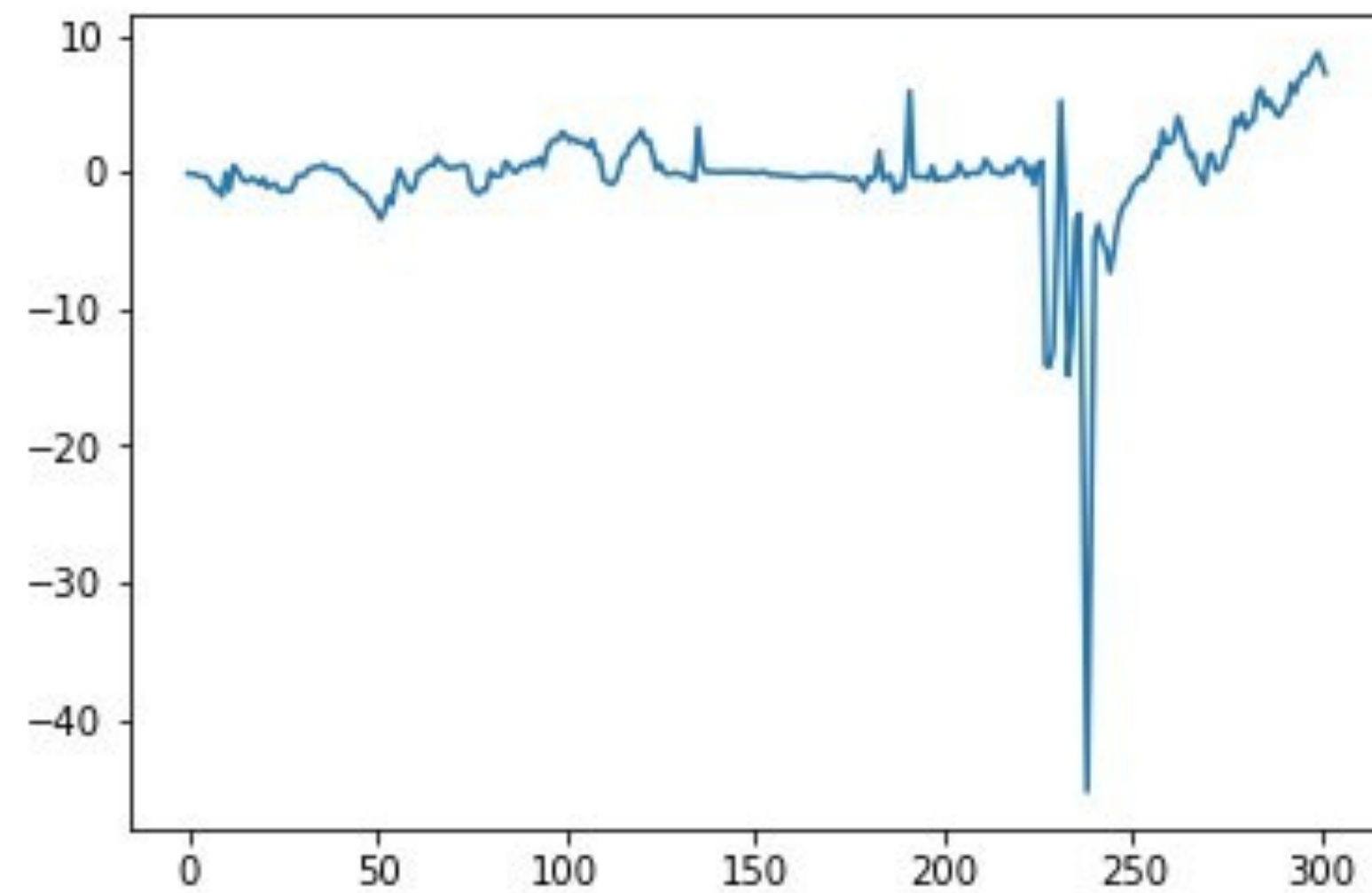
- Смотрели только на верхнюю часть экрана
- На каждом кадре считали яркость
- По временному ряду брали производные 1-го, 2-го порядка, $\text{mean} - \text{min}$, $\text{max} - \text{mean}$, $\text{max} - \text{min}$...
- Обучили catboost, смотрели auc (0.96)



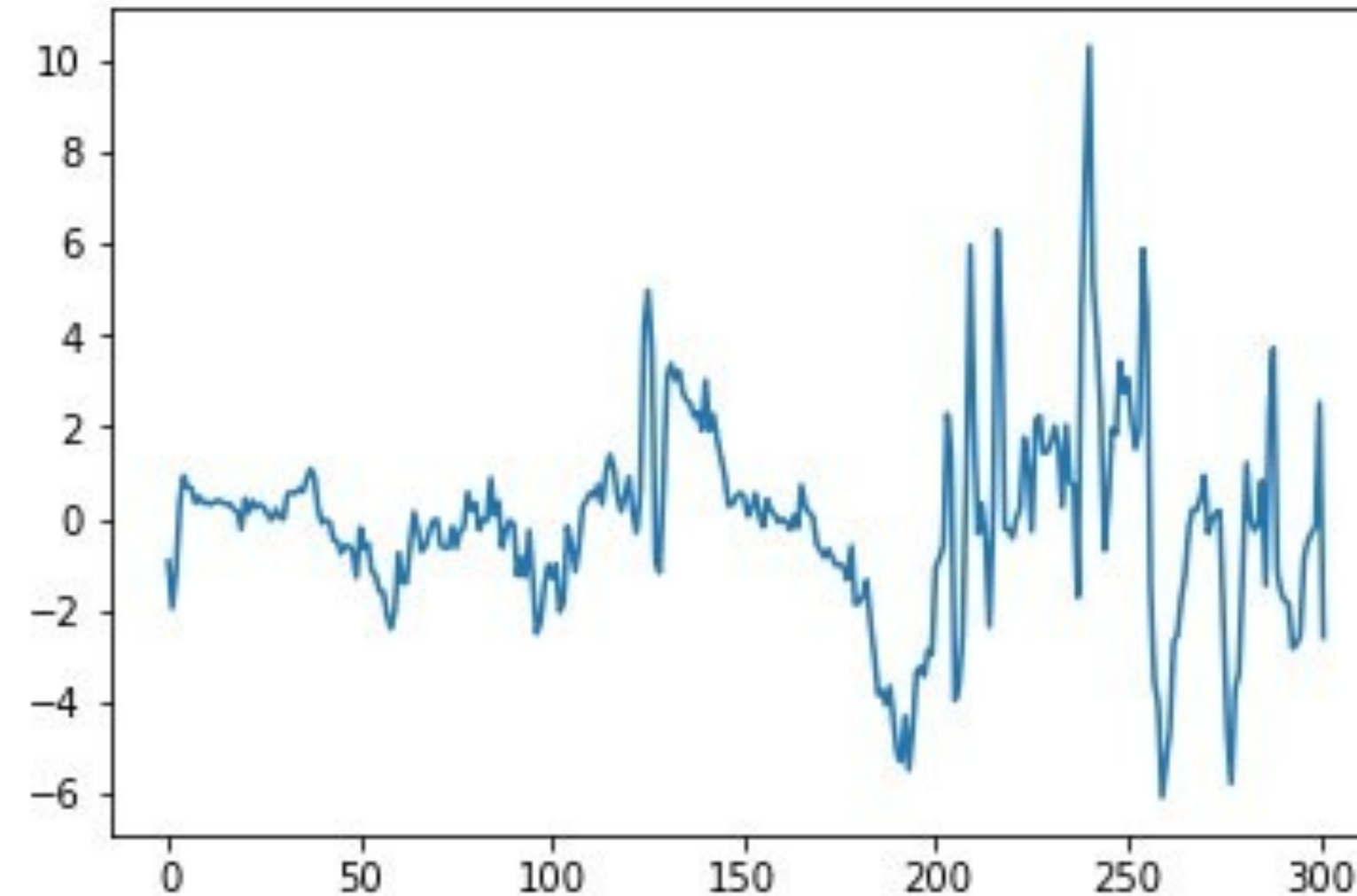
Детекция заезда в туннель

Графики производных от яркости

Въезд под мост



Нет въезда под мост



Детекция дворников

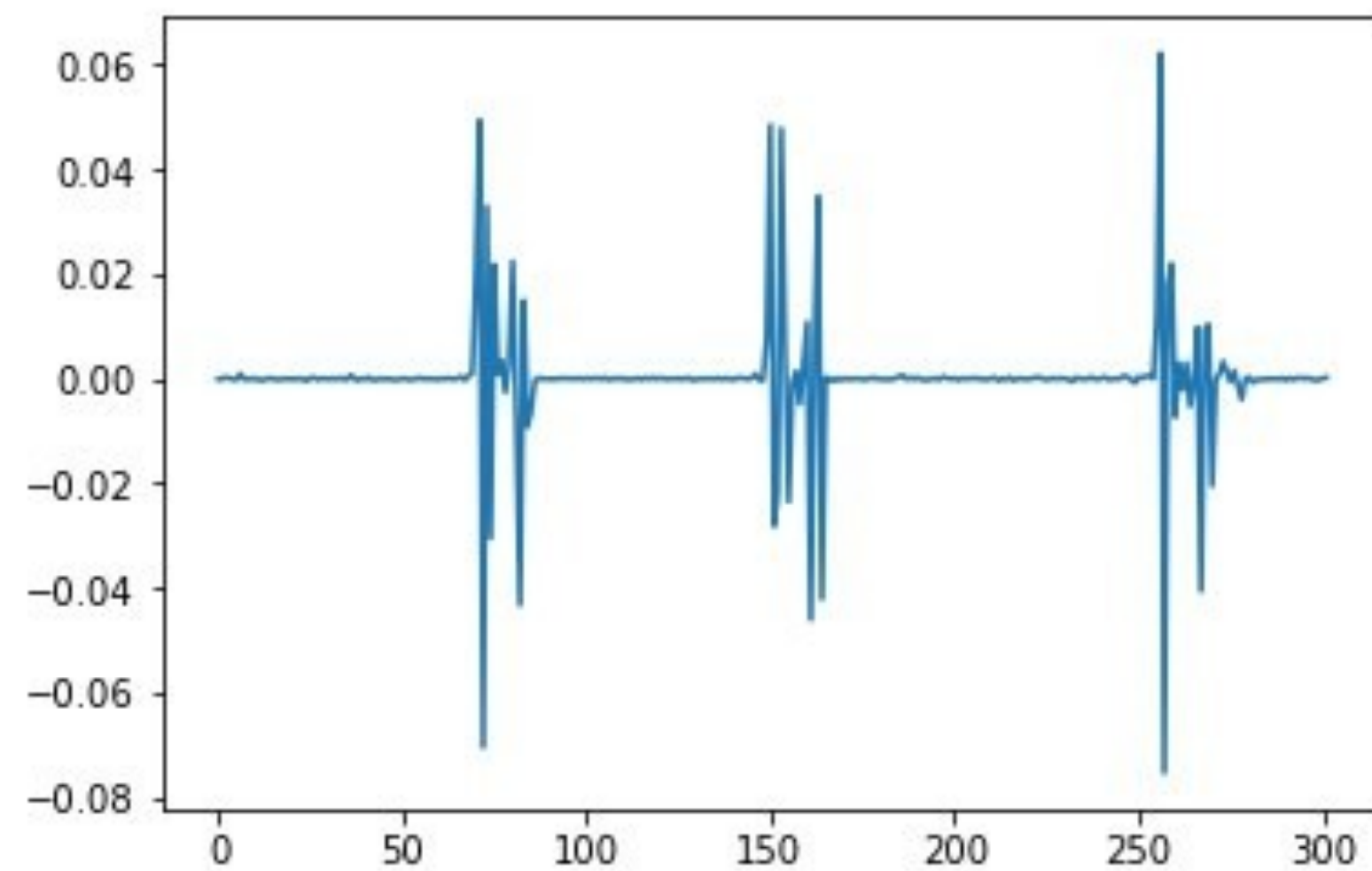
- На каждом кадре видео считали долю черных пикселей в верхней части экрана
- Брали производные 1-го, 2-го порядка, std, mean - min, max - mean, max-min ...
- Обучили catboost, смотрели auc (0.91)



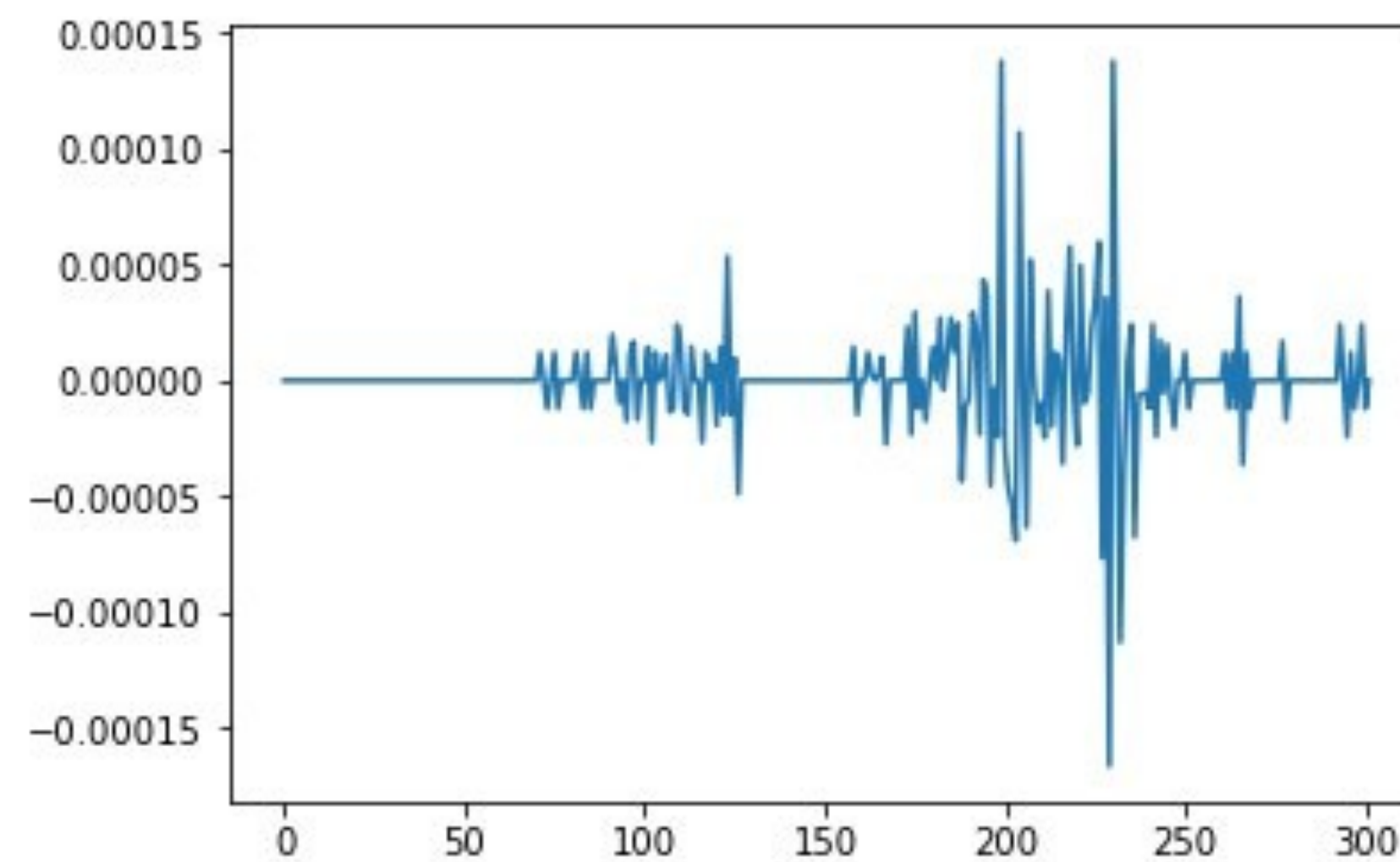
Детекция дворников

Графики производных от доли черного цвета

Включение дворников



Нет дворников

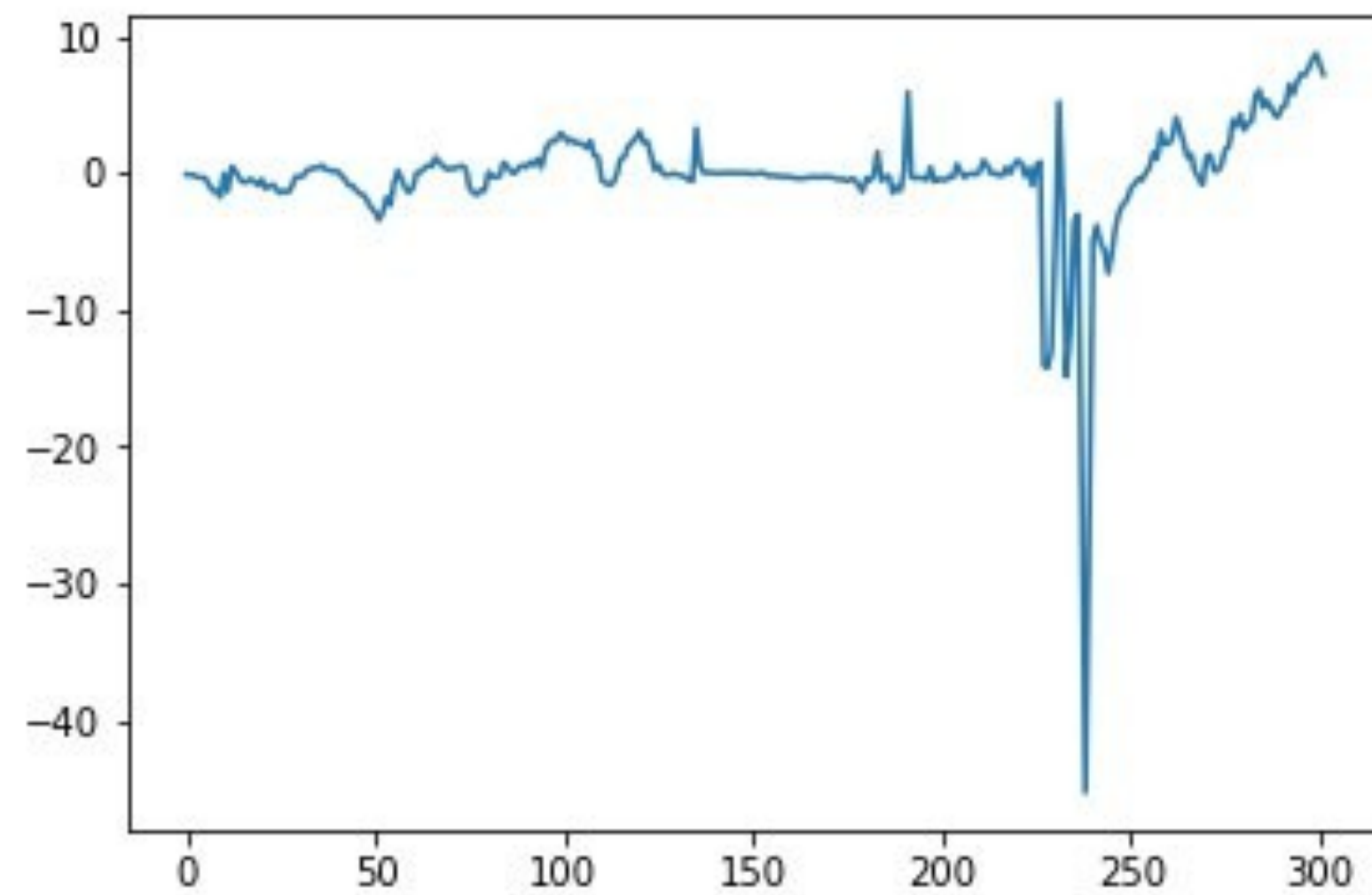


Объединение фичей

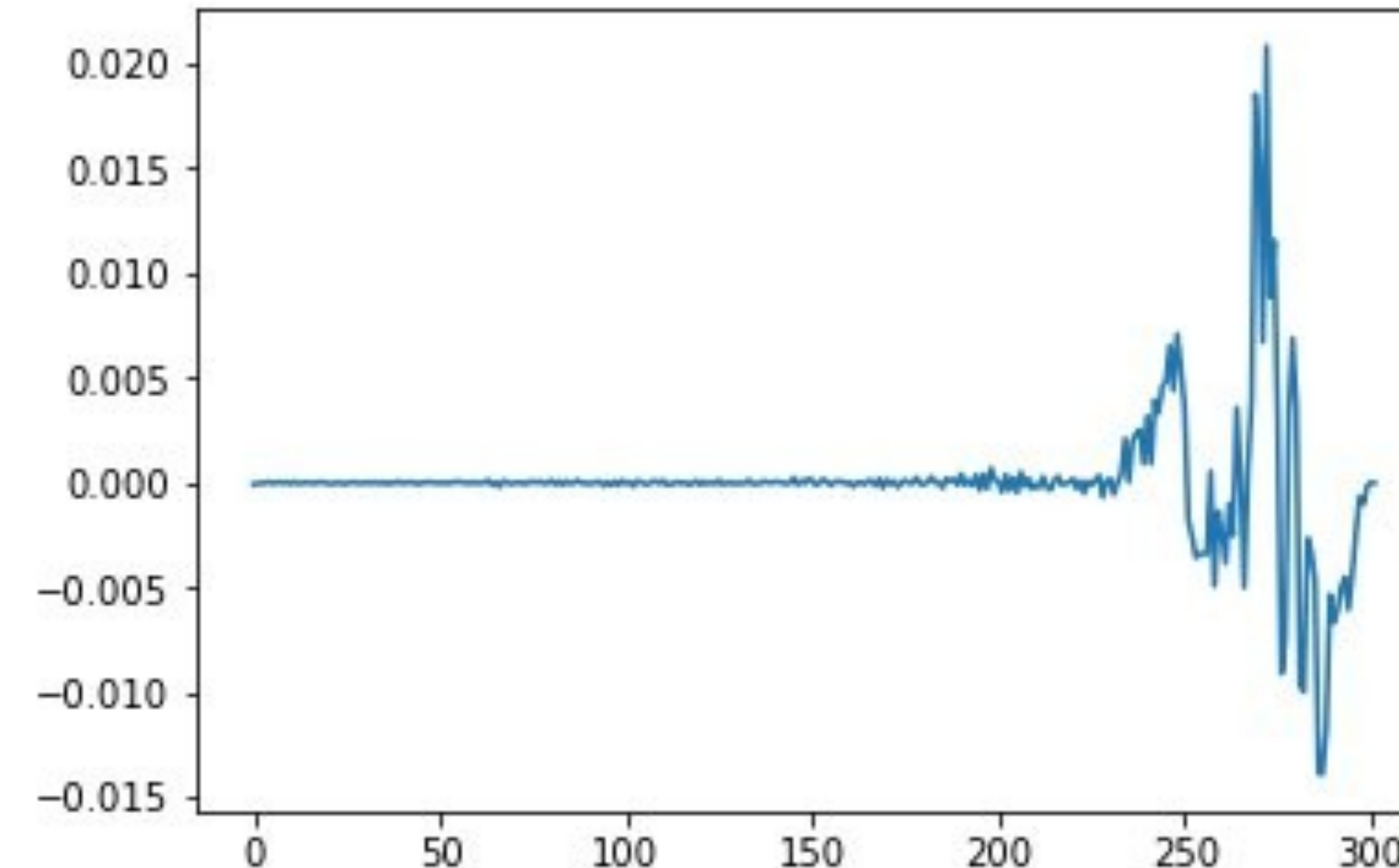
- Перепутали целевую переменную
- Объединение 2-х наборов фичей
- Улучшение сора (туннель: auc 0.99, дворники: auc 0.96)

Въезд под мост

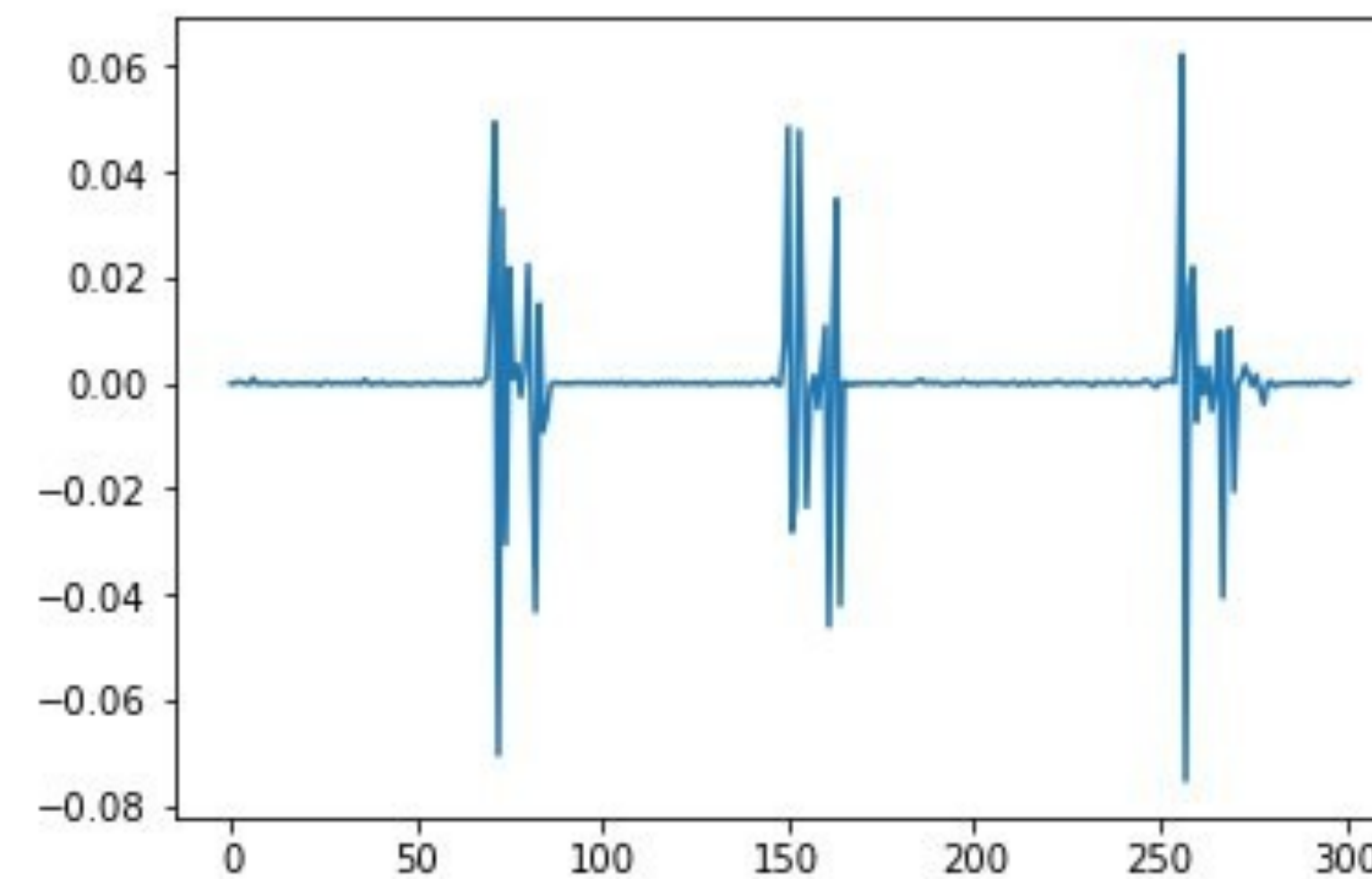
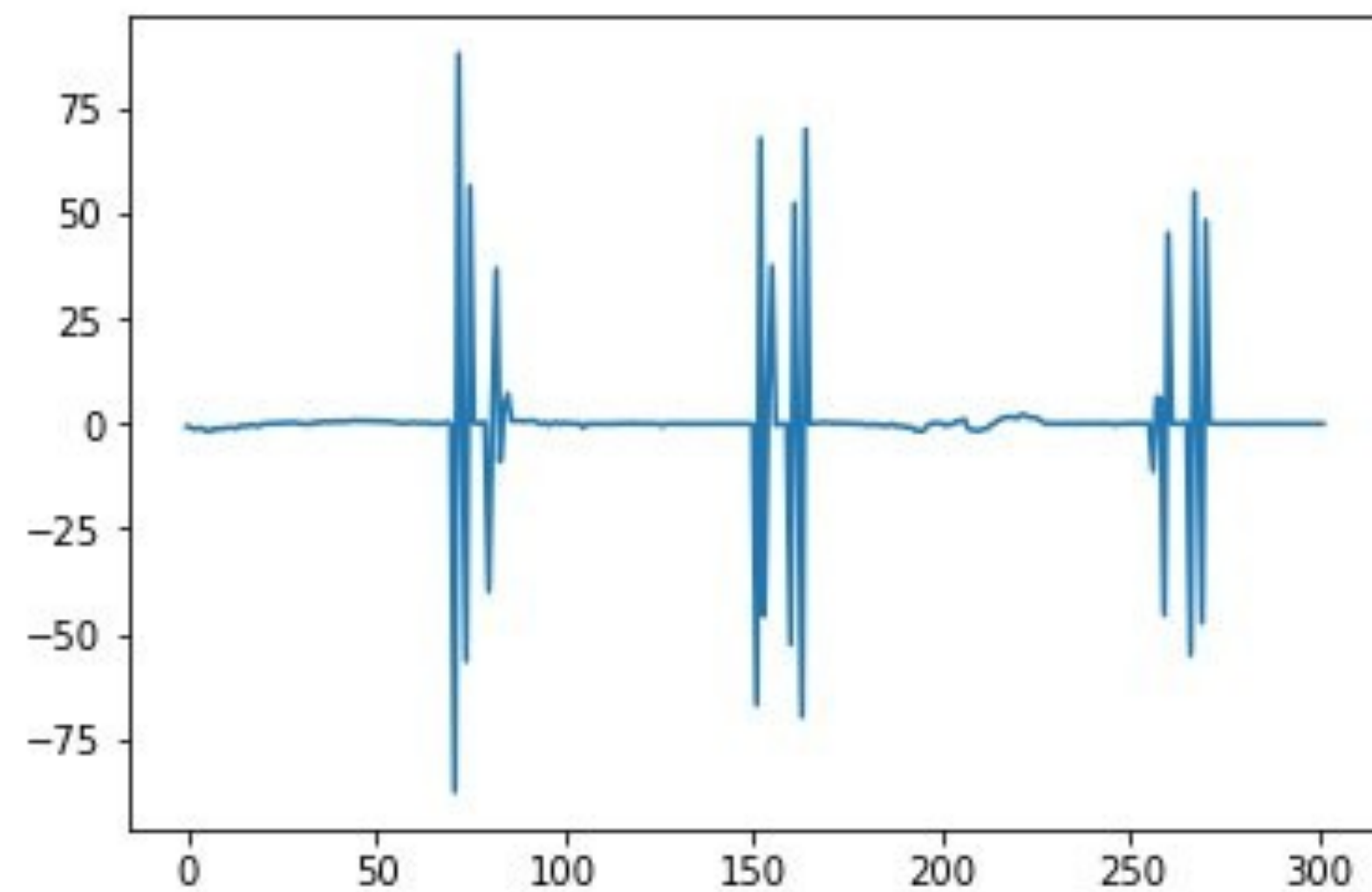
Производная яркости



Производная доли черного цвета



Срабатывание дворников



Детекция выезда/въезда в город

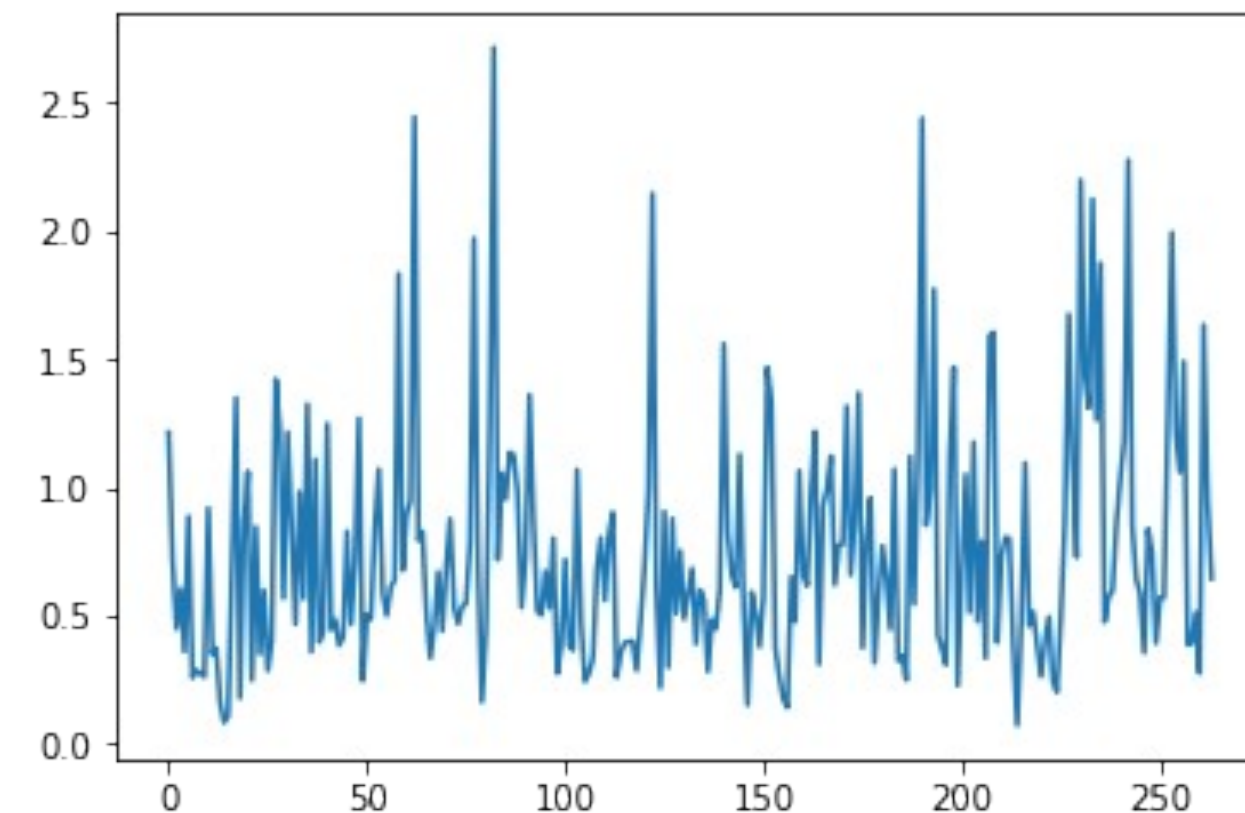
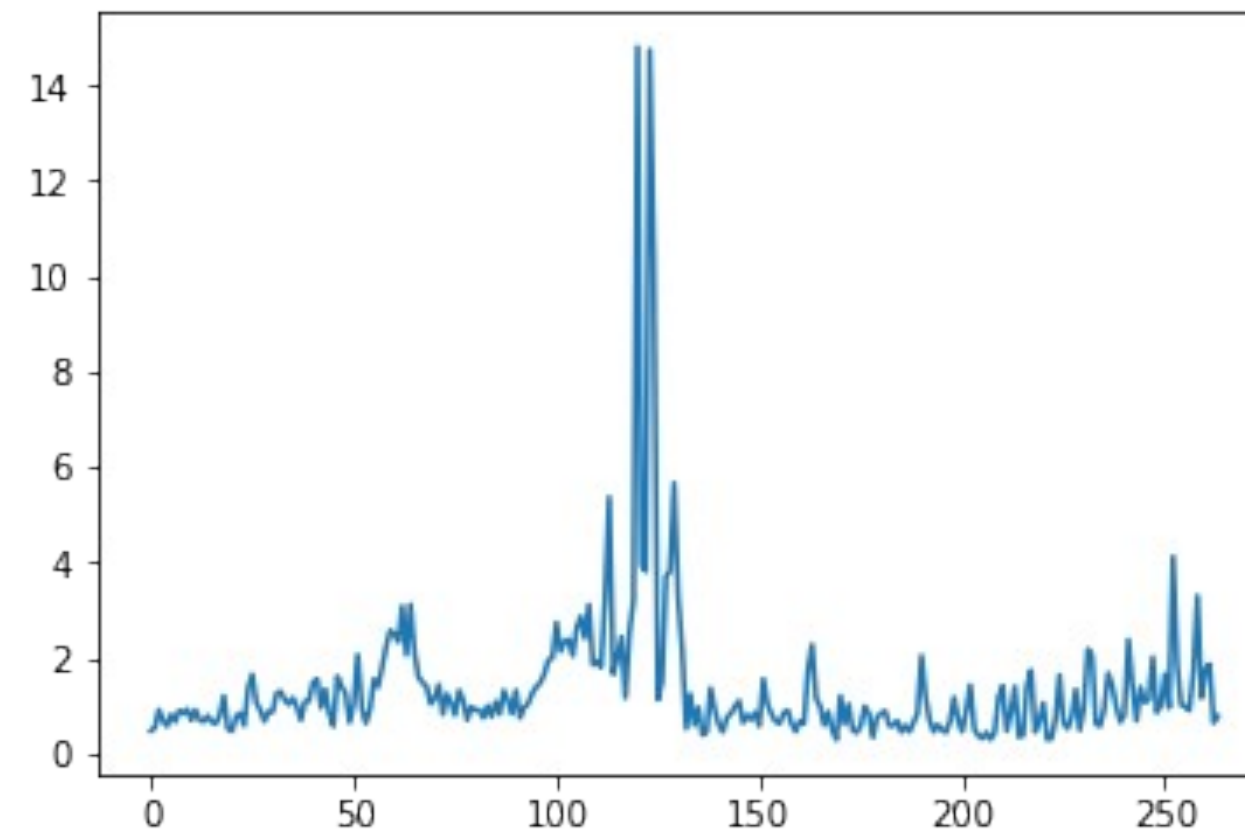
404
NOT FOUND



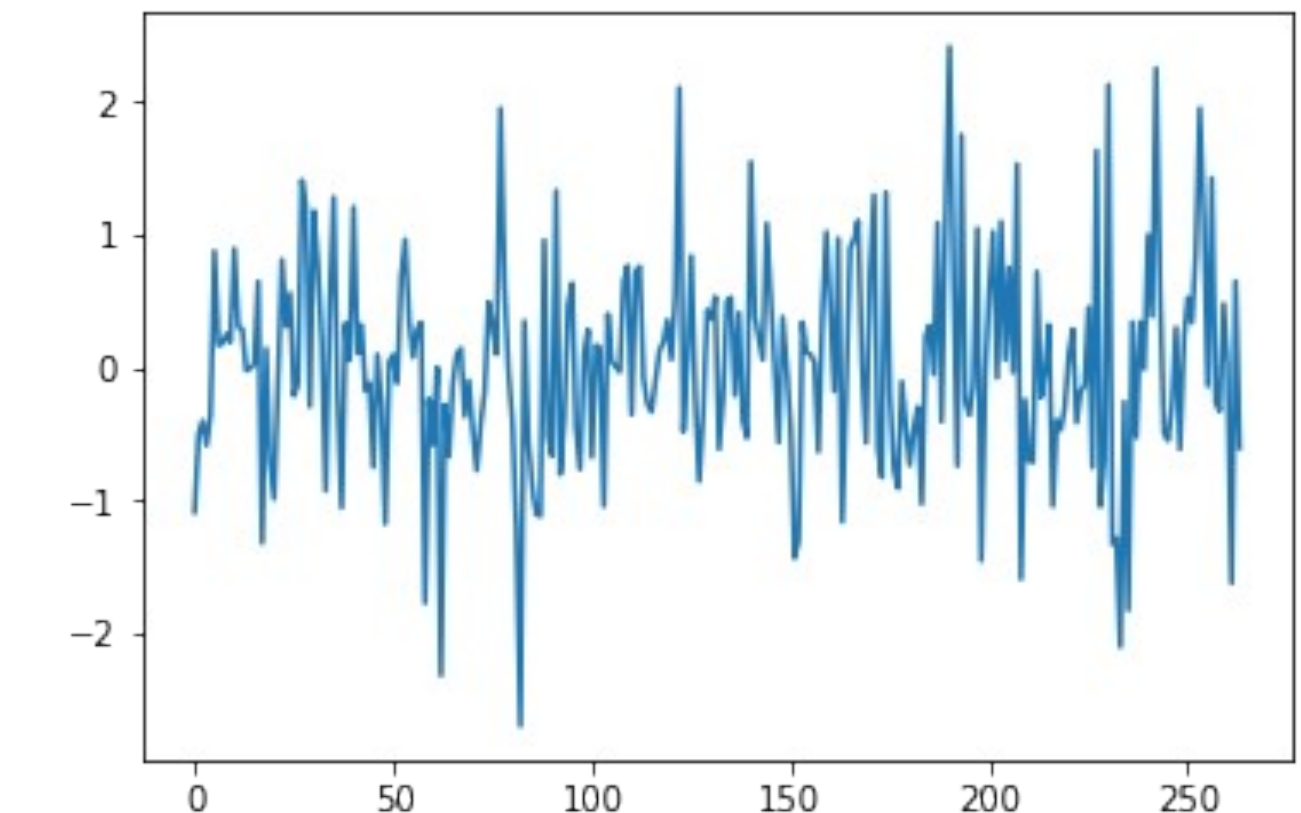
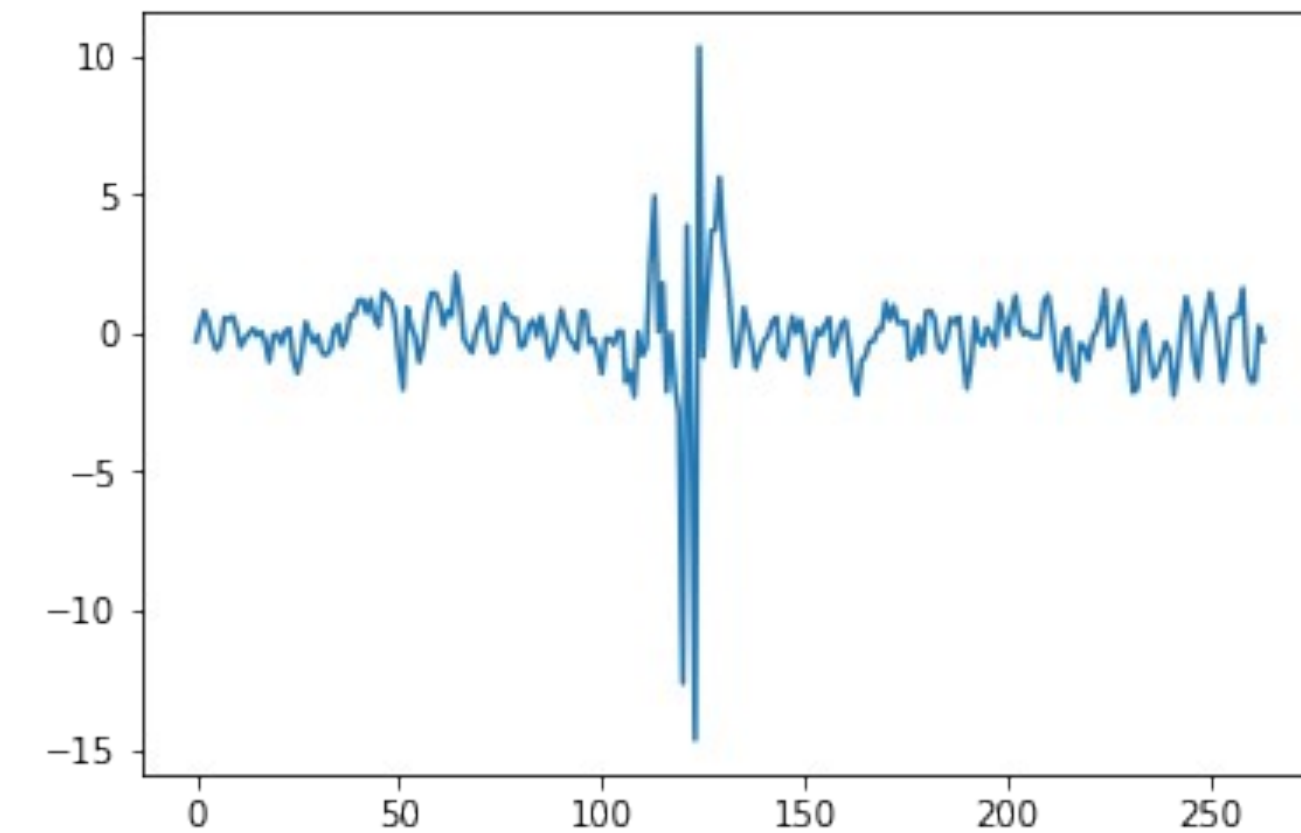
Детекция искусственной неровности

- Lucas-Kanade Optical Flow in OpenCV

Производная нормы отклонения



Производная отклонения по оу



- [Video_1](#)

- [Video_2](#)

Отборочный этап

- Задача: определить кадр переключения светофора с красного на зеленый
 - Обучение: 100 роликов
 - Валидация: 100 роликов
 - Тест: 300 роликов
- Метрика качества:
 - TP - 2
 - TN - 1
 - FN - 1
 - FP - 0

Далее начисленные баллы нормируются на 100

Отборочный этап

- Использовалась tensorflow object detection api
- На каждом кадре в зоне светофора считали долю зеленого цвета
- Брали производную и смотрели на первый максимум, который больше какого-то порога



<https://github.com/KirillTushin/VisionHack>