



Анализ решений задачи детекции лиц на изображениях в сфере киберкриминалистики

Автор: Федор Игоревич Жилкин, группа 17.Б10-мм

Научный руководитель: к. т. н., доцент Ю.В. Литвинов

Консультант: рук. отд. раз. ПО, ООО “Белкасофт”
Н.М.Тимофеев

Санкт-Петербургский государственный университет
Кафедра системного программирования

14 мая 2020г.

Введение

- Изображения, полученные с помощью камер видеонаблюдения или благодаря мобильным телефонам, активно используются службами безопасности по всему миру для поиска нарушителей или установления личности
- Необходимо автоматизировать процесс нахождения лиц на изображениях
- Практически нет работ, сравнивающих решения задачи детекции лиц в сфере киберкриминалистики

Постановка задачи

Целью работы является сравнение существующих решений задачи детекции лиц людей на изображениях на основе трех ключевых факторов:

- качество решения обнаружения лиц
- быстродействие при работе на CPU
- работа под платформой .NET

Задачи:

- выбрать датасет, который будем использовать для тестирования решений
- рассмотреть существующие решения детекции лиц
- отдельно рассмотреть решения, работающие под платформой .NET
- изучить применимые для работы алгоритмы детектирования лиц
- создать тестирующую систему для сравнения решений
- провести полное сравнение всех решений
- выбрать лучшее решение для платформы .NET

Эксперимент

Исследовательский вопрос: какое решение покажет наилучшие результаты, если сравнивать по быстродействию и качеству обнаружения лиц?

Критерии отбора решений: Результаты соревнования на выбранном датасете; решения, работающие под .NET

Датасет для тестирования: Wider Faces

Измеряемые метрики: Recall, Precision, F-мера, время работы

Набор данных WIDER FACES



Рис. 1: Размеченный датасет WIDER FACES

Тестируемые решения

Решения, имеющие лучшие результаты на датасете WIDER FACES:

- AlInnoFace
- DSFD
- SRN

Решения, работающие под платформой .NET:

- Haar's and LBP's cascades
- HOG (Histogram of Oriented Gradients)
- SSD (Single-Shot-Multibox detector)
- MMOD (Maximum-Margin Object Detector)

IoU (Intersection over Union)

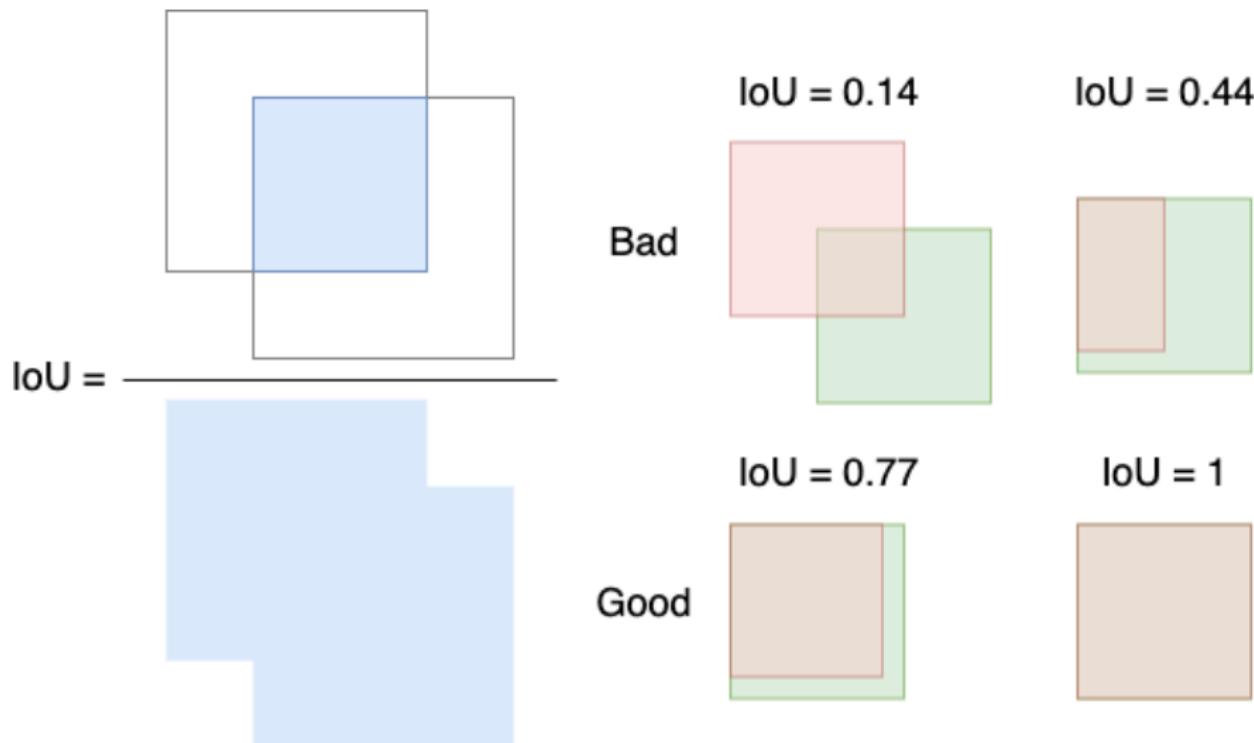


Рис. 2: IoU (Intersection-over-Union) – метрика степени пересечения между двумя ограничивающими рамками

Классы TP, FP, FN



Рис. 3: True-Positive, False-Positive, False-Negative метки

Алгоритм тестирования

```
for image in {Images}
    for ground-truth_b in {Ground-Truth vectors}:
        for predicted_a in {Predicted vectors}:
            if IoU(predicted_a, ground-truth_b) > 0.5 then
                True-Positive (TP) += 1
            ...
            False-Positive = {Predicted vectors}.Count - TP
            False-Negative = {Ground-Truth vectors}.Count - TP
```

Рис. 4: Алгоритм заполнения классов TP, FP, FN

Результаты лучших решений

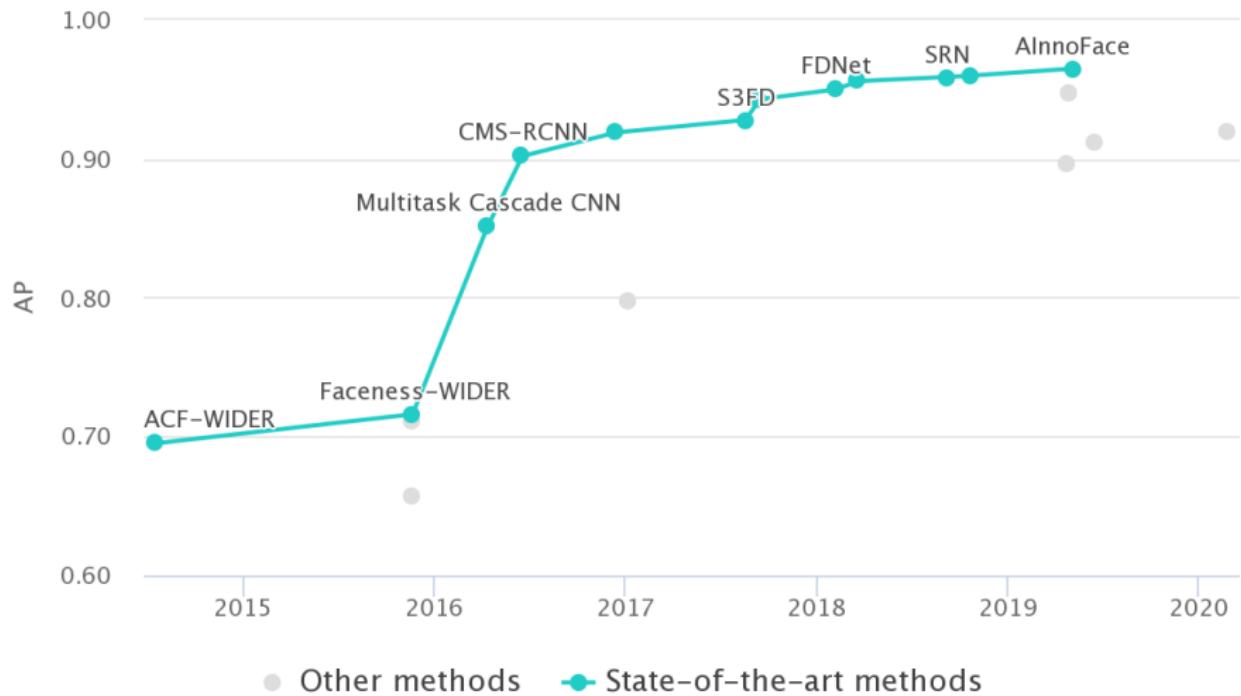


Рис. 5: График Face Detection on WIDER Face

Результаты лучших решений

	Recall
AInnoFace	0.965
DSFD	0.960
SRN	0.959

Таблица 1: Результаты лучших решений

.NET Решения

	Recall	Precision	F-Score
LBP	0.29	0.86	0.43
HOG	0.38	0.95	0.54
Haar	0.45	0.91	0.60
MMOD	0.52	0.86	0.65
SSD	0.82	0.66	0.73
MTCNN	0.86	0.89	0.87

Таблица 2: Результаты существующих решений под .NET

Скорость будем тестировать на изображении размером 300x300.

Каждый алгоритм обработает изображение 10000 раз.

Hardware:

- Processor : Intel Core i7 6850K – 6 Core
- RAM : 32 GB
- GPU : NVIDIA GTX 1080 Ti with 11 GB RAM
- OS : Linux 16.04 LTS
- Programming Language : Python

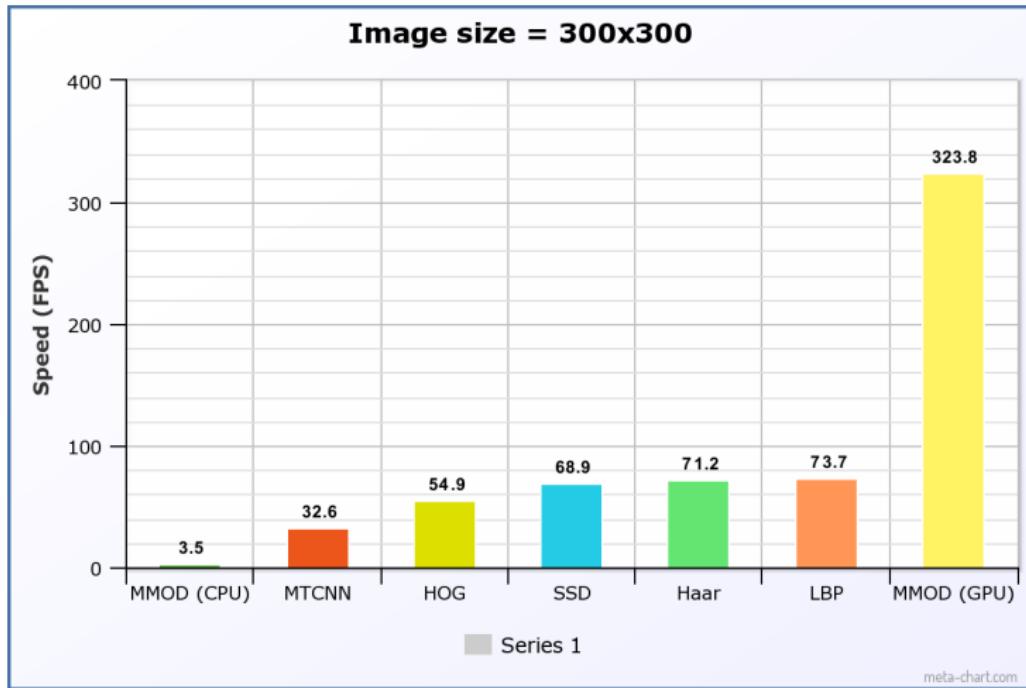


Рис. 6: Средняя скорость работы (прогон изображения 10000 раз)

.NET Решения



Рис. 7: Тест на Occlusion

Результаты

- Сделана система тестирования решений распознавания лиц
- Рассмотрены решения, показавшие лучшие результаты на выбранном датасете
- Реализованы или интегрированы решения, работающие под платформой .NET
- Все решения протестированы по выбранным параметрам
- Выбрано лучшее решение (*Single-Shot-Multibox-Detector*), работающее под платформой .NET
- Изучены алгоритмы и архитектуры глубоких нейронных сетей распознавания лиц