

Ссылка на репозиторий

<https://github.com/PKuznetsov-vl/DlibModel.git> - там же инструкция по запуску и настройке скрипта

Ссылка на модель

https://drive.google.com/drive/folders/1InYhb6IOOXMs1cBs9Nx9KBFr7s_V6m9O?usp=sharing

В качестве алгоритма, построения контрольных точек лица, был выбран алгоритм **shape_predictor**, входящий в состав библиотеки **dlib**.

Работа алгоритма основана на ансамбле деревьев регрессии, который оптимизирует сумму квадратичных потерь и, обрабатывает отсутствующие или частично потерянные данные.

Принцип создания модели можно разделить на несколько этапов:

1) Создание файла с обучающими данными использовался датасет mepro

shape_predictor – в качестве обучающих данных принимает xml документ который содержит в себе

- Путь к изображению
- Bbox лиц на изображении, координаты которого я получил используя детектор из библиотеки dlib. Результаты выдаваемые данным детектором позволили отфильтровать входные данные, на наличие лиц которые не могут быть обнаружены детектором – следовательно обучающая выборка стала точнее
- Координаты контрольных точек – которые так же были отфильтрованы на наличие только 68 точек в файле .pts

2) Создание алгоритма обучения модели

Для корректного обучения алгоритма я установил значения для нескольких параметров

`options.tree_depth = 4` отвечает за глубину каждого дерева регрессии в Ансамбле деревьев регрессии каждое дерево 2 листа ^ глубину

Чем больше значение, тем меньше скорость работы модели, но модель становится более точной

`options.nu = 0.15` используется как параметр регуляризации, значение лежит в диапазоне от 0 до 1, значение близкое к 1 требует меньше обучающих данных, но может переобучить модель

`options.cascade_depth = 25` – глубина каскадов

Чем больше каскадов, тем больше выходная модель, однако она становится более точной

`options.num_test_splits = 50` – Количество разделенных объектов на каждом этапе для выборки. Выбирается тот, который дает наилучший результат

`options.oversampling_amount = 8` - количество искажений изображения, по сути увеличения входных данных)

`options.oversampling_translation_jitter = 0.1` коэффициент искажений изображения лежит в диапазоне от 0 до 0.5

`options.feature_pool_size = 400` - количество пикселей, используемых для создания объектов для деревьев в каждом каскаде.

чем больше пикселей выбрано, тем медленнее модель

Мною был использован сравнительно небольшой объем обучающих данных(изображений),так как при количестве изображений большим 700,я испытывал проблемы с производительностью а именно потеря ОЗУ

Поэтому выбранные параметры, сделать итоговую модель более точной, и затратить не так много ресурсов

Тестирование модели:

Тестирование модели так же проходило в несколько этапов

1)Сначала был запущен алгоритм shape_predictor,работающий на моей модели

При работе алгоритма на тестовой выборке создаются .pts файлы ,которые в дальнейшем будут сравниваться с эталонными значениями находящимися в директории test

2) Результатом сравнения являются графики метрике CED (cumulative error distribution), по оси X в которых идёт нормированная среднеквадратичная ошибка, а по оси Y - процент изображений в тестовой выборке

было проведено сравнение со стандартной моделью shape_predictor_68_face_landmarks_GTX.dat

Тестирование моей модели,а так же модель dlib проводилось на датасете Menpo раздел test

Так же моя модель была протестирована на датасете 300W,результаты можно увидеть ниже

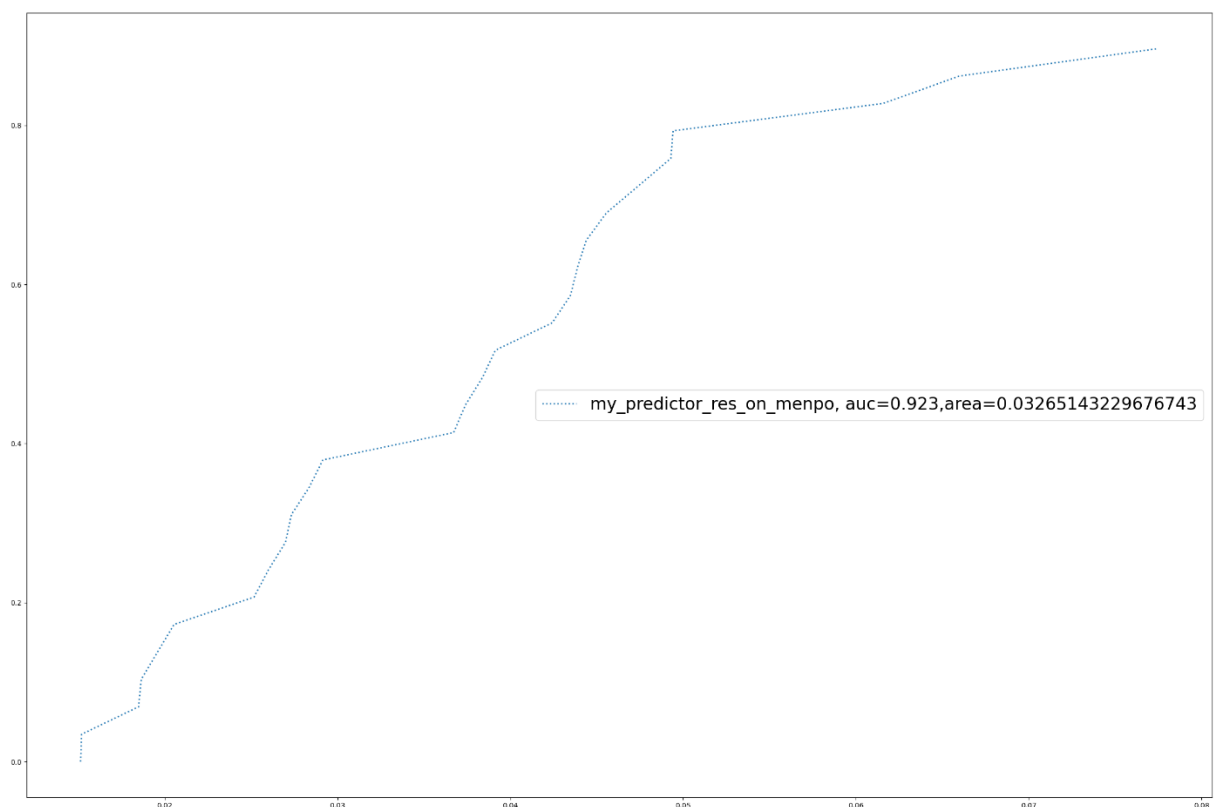


Рис 1. Результат работы моей модели на Menpo/test, точность – 0.923, площадь под графиком - 0.327

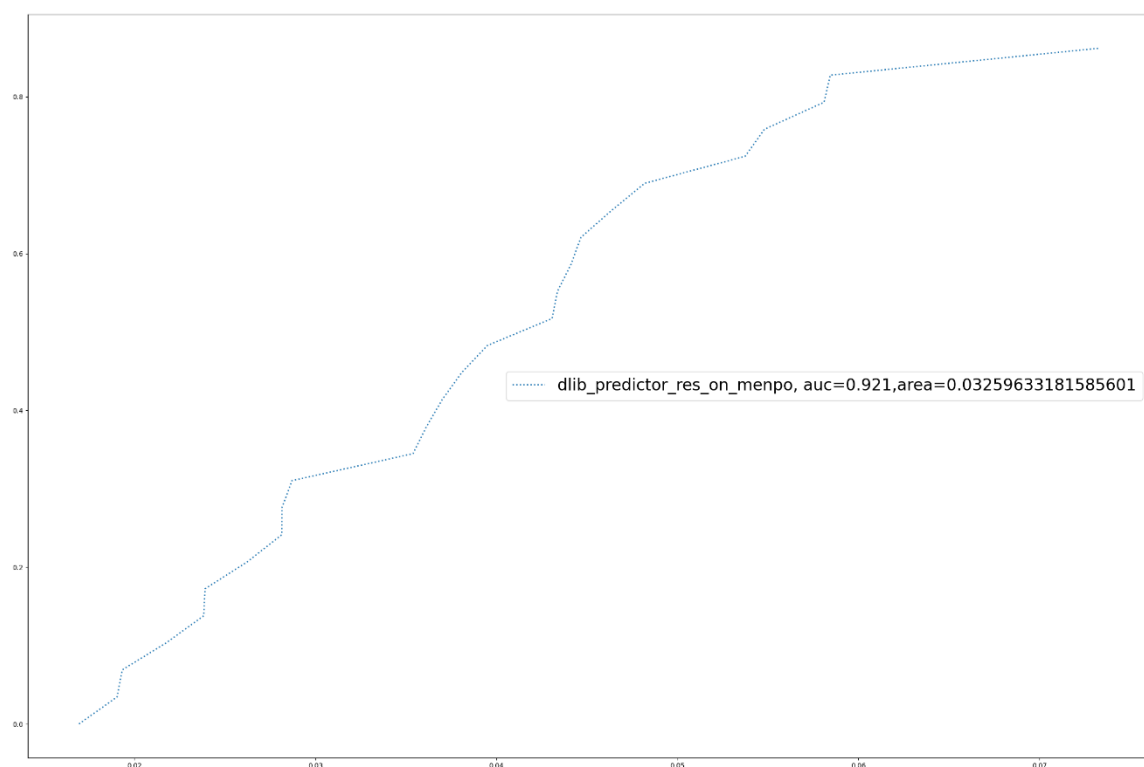


Рис 2. Результат работы dlib модели на Menpo/test, точность – 0.921, площадь под графиком - 0.326

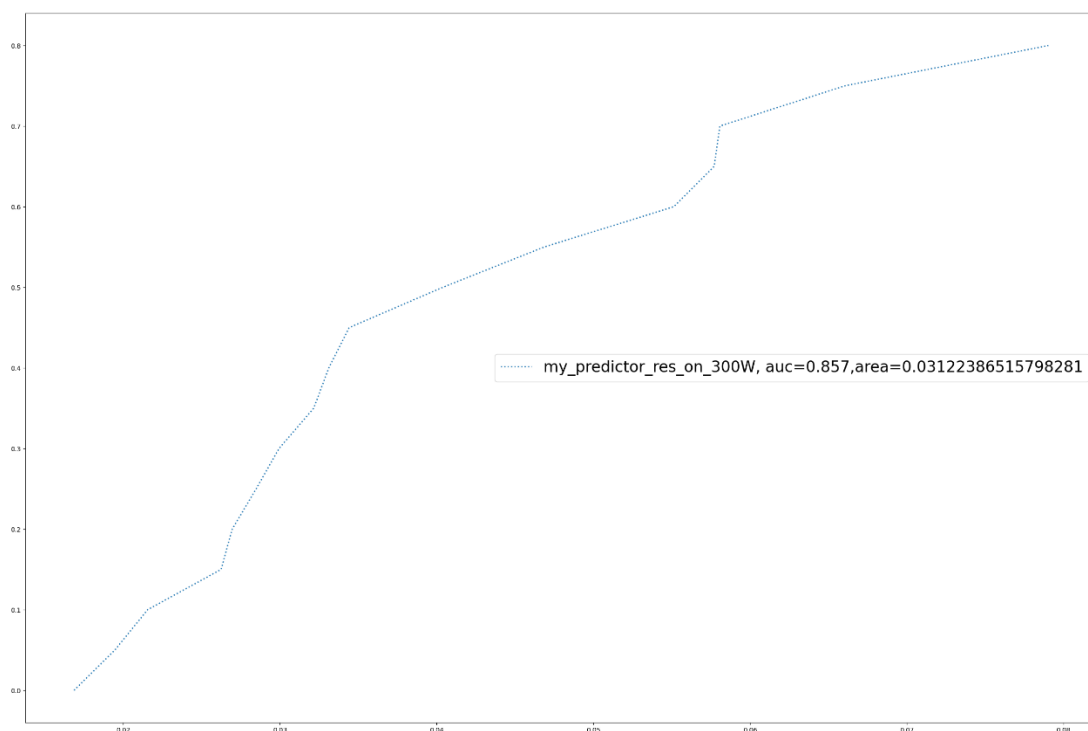


Рис 1. Результат работы моей модели на 300W, точность – 0.857, площадь под графиком - 0.312

Результаты, а так же построенную модель можно найти в папке outputs моего проекта