

Отчет

Задание заключается в реализации алгоритма обнаружения 68 особых точек на лице человека (face alignment), тестировании данного алгоритма на общедоступных датасетах и сравнении с аналогами.

Датасеты для тестирования были взяты следующие:

- 300W
- Менпо

В качестве базовой модели использовалась модель DLIB.

(<https://github.com/davisking/dlib/tree/master>)

Обработка датасетов.

Из всех датасетов удалялись изображения, где landmarks не равны 68. А так же те изображения, где детектор из библиотеки DLIB не смог задетектировать лицо. Можно было увеличить scale DLIB и тогда с мелкими изображениями детектор тоже справился бы. Но было принято решение не делать так, так как тогда приходилось обсчитывать детекцию слишком долго (отдельно для мелких и для крупных изображений в силу нехватки памяти и к тому же таких изображений было немного).

Все датасеты сохранялись в отдельные датафреймы, где так же были сохранены и координаты боксов детекций, кол-во точек, попавших в бокс детекций. На некоторых изображениях было несколько лиц и нужно было соотнести, координаты точек лица и бокс детекции для определённого изображения. Для этого рассчитывалось, сколько точек попадает в каждый бокс детекции и выбирался тот бокс, где больше точек.

	image_dir	real_landmarks	rect_x1	rect_y1	rect_x2	rect_y2	quantity_point	quantity_faces	n_landmarks
0	./data/Menpo/test/aflw_face_40656.jpg	[[373.862, 405.73], [361.104, 489.078], [356.4...	357.0	332.0	865.0	839.0	58.0	1	68
1	./data/Menpo/test/aflw_face_45431.jpg	[[97.429, 149.92], [103.294, 170.166], [110.29...	111.0	99.0	252.0	240.0	60.0	1	68
2	./data/Menpo/test/aflw_face_62296.jpg	[[43.365, 54.52], [43.65, 60.272], [44.221, 66...	36.0	27.0	93.0	83.0	57.0	1	68
3	./data/Menpo/test/aflw_face_46522.jpg	[[60.406, 79.666], [61.207, 88.885], [62.944, ...	53.0	45.0	135.0	127.0	61.0	1	68
4	./data/Menpo/test/aflw_face_64997.jpg	[[121.388, 339.202], [131.677, 371.279], [145....	91.0	192.0	384.0	485.0	57.0	1	68
...
893	./data/Menpo/test/aflw_face_62383.jpg	[[43.465, 57.268], [42.735, 65.527], [43.206, ...	30.0	36.0	98.0	104.0	56.0	1	68
894	./data/Menpo/test/aflw_face_63468.jpg	[[94.606, 123.316], [94.44, 136.701], [96.093,...	66.0	61.0	208.0	203.0	61.0	1	68
895	./data/Menpo/test/aflw_face_64435.jpg	[[92.942, 120.016], [91.463, 135.927], [93.51,...	61.0	45.0	202.0	187.0	52.0	1	68
896	./data/Menpo/test/fddb_image2646_1.jpg	[[105.538, 139.849], [106.785, 158.314], [106....	86.0	92.0	227.0	233.0	57.0	2	68
897	./data/Menpo/test/aflw_face_41554.jpg	[[72.546, 165.836], [74.784, 190.864], [78.853...	62.0	107.0	266.0	311.0	65.0	2	68

898 rows x 9 columns

Создание Dataset

При формировании train dataset нужно было произвести 2 этапа обработки как изображений, так и точек.

1. Крпать изображения по боксу детекции и соответственно пересчитывать координаты точек.
2. Resize изображений для подачи в модель и так же пересчитывать координаты точек.
3. Так же в ходе экспериментов были применены некоторые цветовые трансформации к изображениям.
4. К val и test dataset не применялись цветовые трансформации соответственно.

Метрика

Для оценки качества модели использовалась метрика AUC CED

В качестве лосса: MSE (расстояние между точками)

Модели

1. Модель Onet.

Num_epoch = 500,

Оптимайзер – AdamW(lr = 0.001),

Шедуллер – CosineAnnealingLR

Протестировано на обоих датасетах (помню, что DLIB тоже обучалась на этом датасете). Но чтобы можно было соотнести результаты – построила совместный график (рис.1, 2).

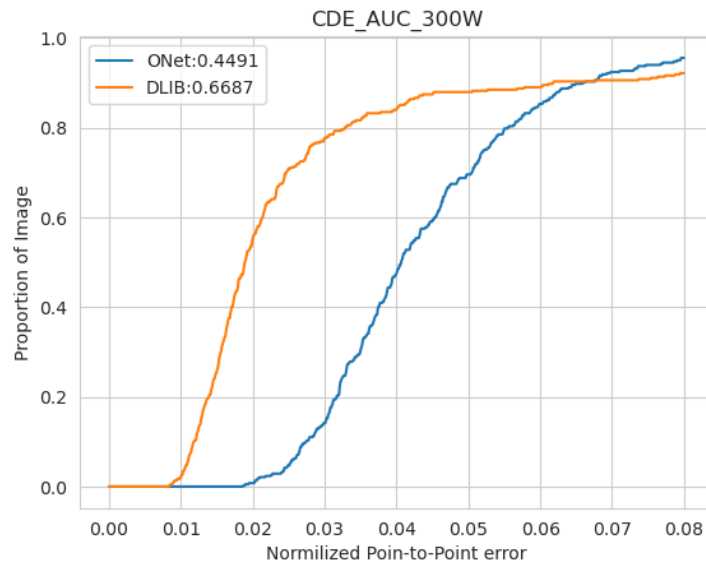


Рис.1

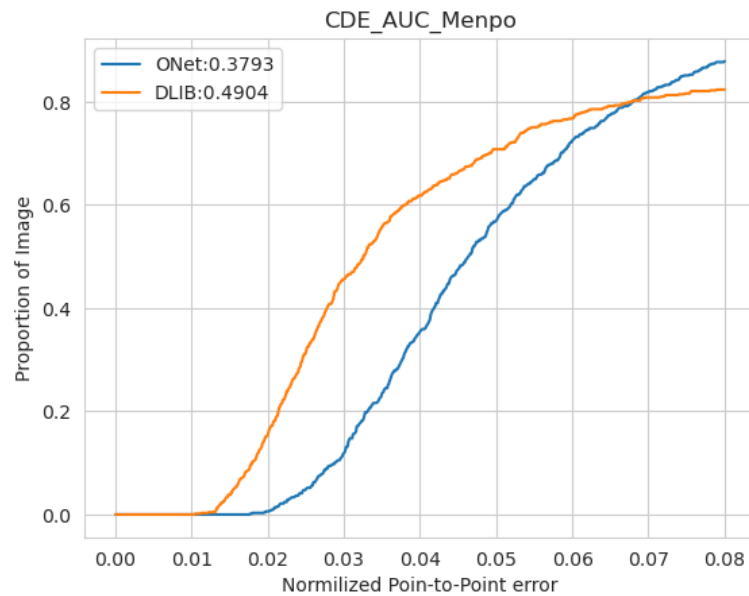


Рис.2

Как видно из графика ONet не показала хороших результатов (CED_AUC = 0.45 для 300W и CED_AUC = 0.38 для Менро).

Потому решила сразу попробовать более глубокую архитектуру сети.

2. Модель DAN

(https://www.researchgate.net/publication/319277818_Deep_Alignment_Network_A_Convolutional_Neural_Network_for_Robust_Face_Alignment)

Базовая архитектура была взята из данной статьи. И добавлены батчнорм слои.

Table 1. Structure of the feed-forward part of a Deep Alignment Network stage. The kernels are described as height \times width \times depth, stride.

Name	Shape-in	Shape-out	Kernel
conv1a	112 \times 112 \times 1	112 \times 112 \times 64	3 \times 3 \times 1,1
conv1b	112 \times 112 \times 64	112 \times 112 \times 64	3 \times 3 \times 64,1
pool1	112 \times 112 \times 64	56 \times 56 \times 64	2 \times 2 \times 1,2
conv2a	56 \times 56 \times 64	56 \times 56 \times 128	3 \times 3 \times 64,1
conv2b	56 \times 56 \times 128	56 \times 56 \times 128	3 \times 3 \times 128,1
pool2	56 \times 56 \times 128	28 \times 28 \times 128	2 \times 2 \times 1,2
conv3a	28 \times 28 \times 128	28 \times 28 \times 256	3 \times 3 \times 128,1
conv3b	28 \times 28 \times 256	28 \times 28 \times 256	3 \times 3 \times 256,1
pool3	28 \times 28 \times 256	14 \times 14 \times 256	2 \times 2 \times 1,2
conv4a	14 \times 14 \times 256	14 \times 14 \times 512	3 \times 3 \times 256,1
conv4b	14 \times 14 \times 512	14 \times 14 \times 512	3 \times 3 \times 512,1
pool4	14 \times 14 \times 512	7 \times 7 \times 512	2 \times 2 \times 1,2
fc1	7 \times 7 \times 512	1 \times 1 \times 256	-
fc2	1 \times 1 \times 256	1 \times 1 \times 136	-

Num_epoch = 300

Оптимайзер – AdamW(lr = 0.001),

Шедуллер – CosineAnnealingLR

В качестве дополнительных трансформаций были использованы цветные трансформации из Pytorch.

(v2.RandomSolarize(threshold=192.0, p=0.7),

v2.RandomAdjustSharpness(sharpness_factor=2),

v2.ColorJitter(brightness=.5, hue=.3)

Результаты DAN

Датасет 300W (CED_AUC = 0.571) (рис.3)

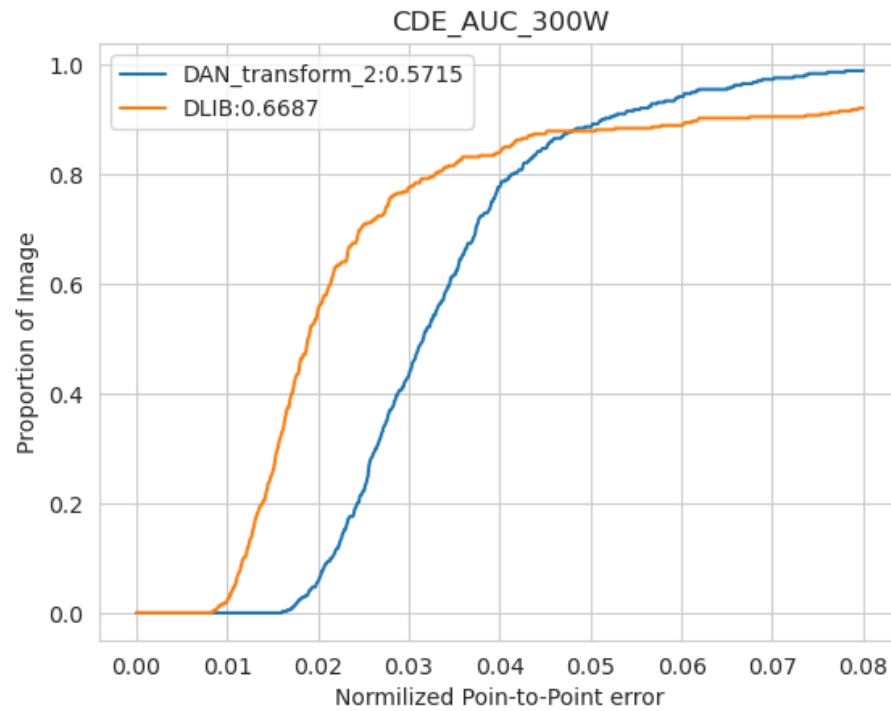


Рис.3

Датасет Менро (CED_AUC = 0.552) (рис.4)

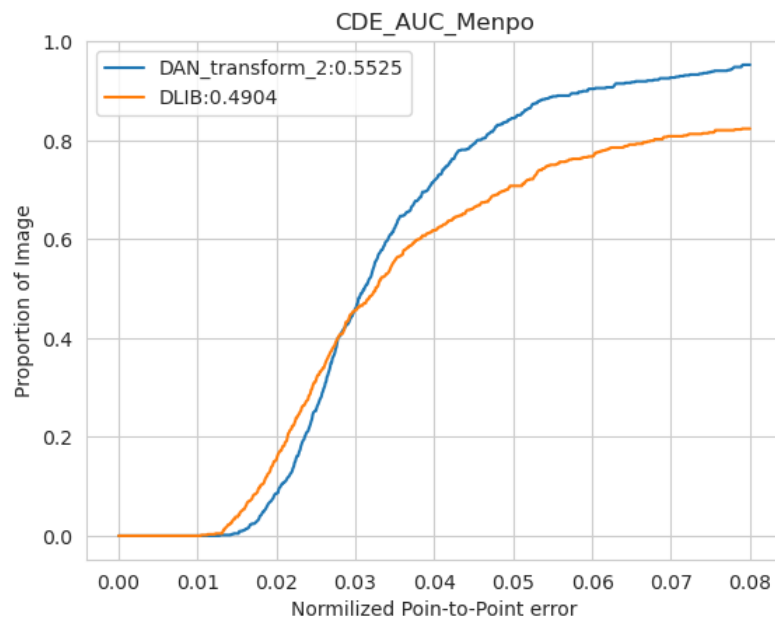


Рис.4

Выводы

Результаты моделей по метрике CED_AUC.

Датасет	DLIB	DAN	ONet
300W	0.66	0.57	0.45
Menpo	0.49	0.55	0.38

Как видно, на датасете Menpo – DLIB показывает хуже результат, что очевидно (на нем она НЕ обучалась). Модель DAN – достаточно стабильна. ONet показывает не очень хорошие результаты.

В качестве экспериментов по применениям трансформаций были опробовано меньшее кол-во цветовых трансформаций, но как видно это дает хуже результат - CED_AUC = 0.53 (рис. 5).

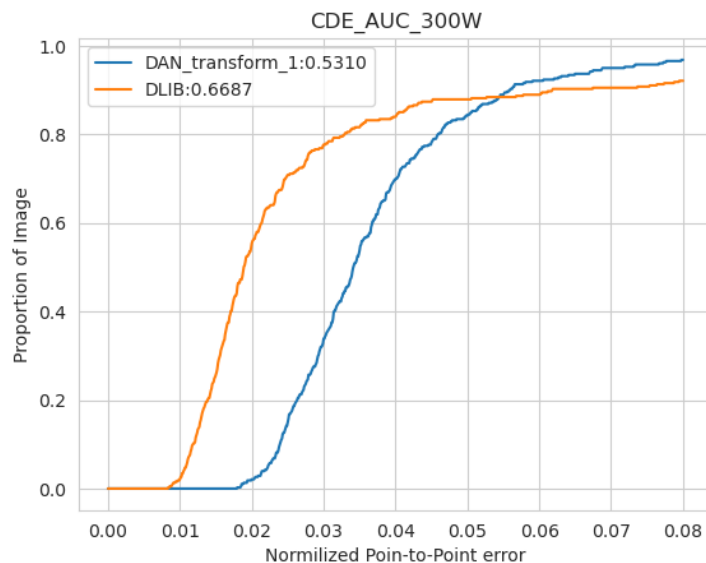


Рис.5

Код обеих моделей находится в репозитории

<https://github.com/anna244/Face-alignment.git>

Файл обученной модели ONet ([ONet_experiment_final.ipynb](#))

Файл обученной модели DAN ([DAN_experiment_final.ipynb](#))

Что можно было еще попробовать, но не хватило времени.

- Использовать геометрические трансформации (но нужно тогда учитывать изменение координат точек).

- Более интересные модели (<https://habr.com/ru/articles/661671/>).