

# Обнаружение особых точек на лице человека

Декабрь 2018

## 1 Подготовка данных

Для увеличения количества обучающих и тестовых данных, train и test части датасетов Menpo и 300W были объединены между собой соответственно.

Далее, из объединенного датасета были удалены примеры, аннотированные 39 точками, а также те картинки, на которых детектор DLIB нашел 0 лиц или больше 1го. Выяснилось, что детектор не нашел ни одного лица примерно на 920 изображениях, а больше 1 лица - примерно на 1тыс.

Затем вырезанные лица были сохранены отдельно и при обучении модели использовались именно они

## 2 Модель

Модель была взята из предложенной [статьи](#) с единственным изменением: последний слой был заменен с

$$Linear(in\_channels = 256, out\_channels = 16)$$

на

$$Linear(in\_channels = 1152, out\_channels = 136)$$

## 3 Обучение

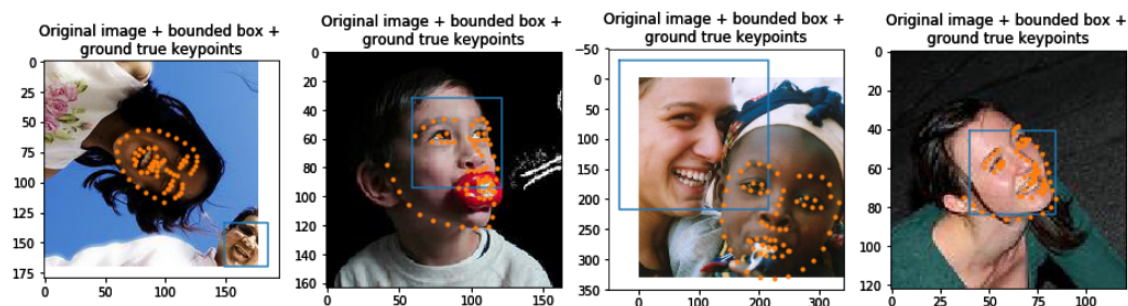
Сначала модель обучалась в течение 5 эпох для построения общей картины.

При обучении было замечено, что лосс на некоторых батчах остается очень высоким при том, что на других батчах заметно падает. Был проведен анализ, в ходе которого были рассмотрены изображения, на которых модель имела большую функцию ошибки. Некоторые примеры представлены на рисунке ниже

Видно, что в большинстве случаев ошибка связана с неверной работой детектора: на картинках с несколькими лицами он находит только одно, причем ground true разметка указана для другого лица.

Эти картинки были удалены из датасета, и модель была поставлена обучаться на 50 эпох. Кривые обучения представлены ниже на рисунке 2

TRAIN:



TEST:

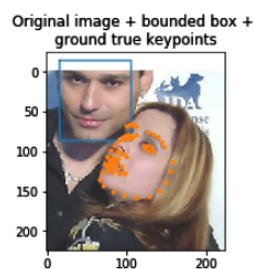


Рис. 1: Картинки, на которых модель имеет большую функцию потерь

## 4 Подсчет метрик и оценка качества

Далее были подсчитаны метрики  $ced_{auc}$  для обученной модели и для детектора. Кривые представлены на рисунке 3.

Подсчет показал, что значение метрики у модели выше, чем у детектора.

Стоит отметить, что функции потерь, используемая при обучении, и среднеквадратичная ошибка, используемая для оценки метрики – разные величины

Работу модели на некоторых тестовых картинках можно увидеть на рисунке 4

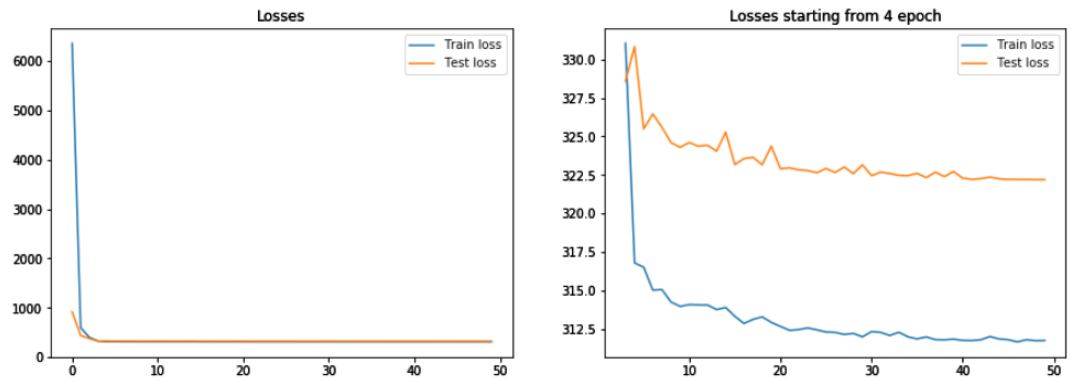
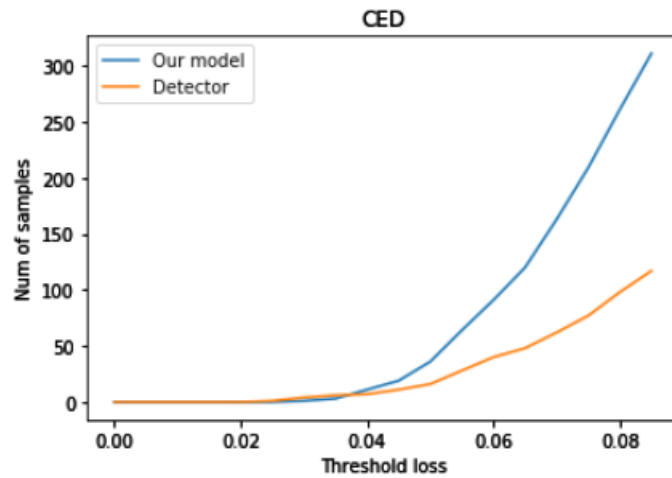


Рис. 2: Функции потерь при обучении



```
1 print("ced_auc для полученной модели при пороге 0.08: %.2f" % ced_auc)
2 print("ced_auc для детектора при пороге 0.08: %.2f" % ced_auc_detector)
```

```
ced_auc для полученной модели при пороге 0.08: 6.44
ced_auc для детектора при пороге 0.08: 2.58
```

Рис. 3: *ced* и *ced\_auc* для обученной модели и детектора

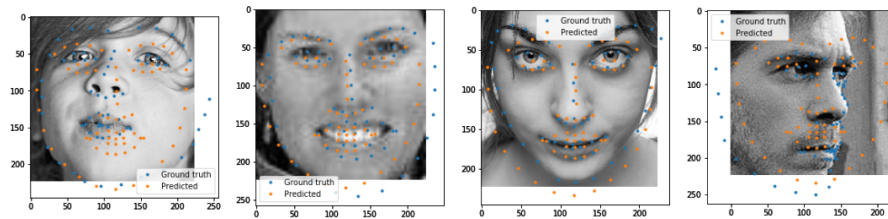


Рис. 4: Некоторые изображения с ground true и предсказанных разметками