Метод поворота лица с использованием оценки ориентиров лица (face landmark estimation). Предлагается использовать библиотеки dlib и openface (https://github.com/cmusatyalab/openface)

Пример кода и использования: https://gist.github.com/ageitgey/82d0ea0fdb56dc93cb9b716e7ceb364b

**Входные данные:** Процесс происходит в 2 этапа, в первом на вход подаются 2 параметра, собственно само изображение и так называемый face\_rect (рамка, окружающая детектированное лицо, dlib.rectangle).

pose\_landmarks = face\_pose\_predictor(image, face\_rect)

Здесь также приведен пример нахождения 68 меток

<https://stackoverflow.com/questions/42941079/how-do-i-align-a-face-for-preprocessing>

Для лучшего понимания можно посмотреть вот этот файл

(https://github.com/cmusatyalab/openface/blob/master/openface/align\_dlib.py)

Далее, зная эти метки, выравниваем лицо:

alignedFace = face\_aligner.align(534, image, face\_rect, landmarkIndices=openface.AlignDlib.OUTER\_EYES\_AND\_NOSE)

где первый параметр - Длина края в пикселях квадрата , к которому изменяется изображение, второй – само изображение, третий – его рамка, параметр landmarkIndices – индексы меток для преобразования

**Выходные данные:** Можно сохранить выходное изображение в простом формате, например, jpg

cv2.imwrite("aligned\_face\_{}.jpg".format(i), alignedFace)

В принципе на выходе мы получаем The aligned RGB image. Shape: (imgDim, imgDim, 3)

**На чем основано:** основная идея в том, что мы отмечаем на лице 68 точек-ориентиров и после этого используем базовые афинные преобразования (поворот и масштабирование), так как они не искажают параллельные линии, трехмерные преобразования в свою очередь портят изображение и могут осложнить последующую работу с ними