Detekcija lica

HOW DOES FACE DETECTION WORK WITH DEEP LEARNING?

As we mentioned earlier, deep learning is a subset of machine learning in which large neural networks process huge amounts of data and make complex predictions. One of the most popular deep learning approaches is the Multi-Task Cascaded Convolutional Neural Network – or, MTCNN. This approach is popular because it achieved cutting-edge results (for the time) on a variety of benchmark datasets – plus, it is able to use landmark detection to recognize the eyes, mouth, and other facial features. MTCNN uses a cascade structure that contains three networks: P-net, R-Net, and O-Net. The image is first rescaled to different sizes (or an image period). P-Net proposes facial regions, R-Net filters the bounding boxes, and O-Net proposes facial landmarks

**Face detection tools**

There are dozens of face detection solutions, both proprietary and open-source, that offer various features, from simple face detection to emotion detection and face recognition.

*Proprietary face detection software*

-Amazon Recognition, Face++, Face Recognition and Face Detection API (Lambda Labs), Kairos, Microsoft Azure Cognitive Services Face API, Paravision, Trueface

**Open-source face detection solutions**

-Ageitgey/face\_recognition is one of the most extensive face recognition libraries. The contributors also claim it to be the “simplest facial recognition API for Python and the command line.”

-Deepface is a framework for Python providing facial attribute analysis like age, gender, race, and emotion. It also provides REST API.

-FaceNet developed by Google uses the Python library for implementation. The accuracy of recognition is 99,65%, and it does not have REST API.

-InsightFace is another Python library. The recognition accuracy is 99,86%..

-InsightFace-REST is an actively updating repository that “aims to provide convenient, easily deployable and scalable REST API for InsightFace face detection and recognition pipeline using FastAPI for serving and NVIDIA TensorRT for optimized inference.”

-OpenCV isn’t an API, but it is a valuable tool with over 3,000 optimized computer vision algorithms. It offers many options for developers, including Eigenfacerecognizer, LBPHFacerecognizer, or lpbhfacerecognition face recognition modules.

-OpenFace is a Python and Torch implementation of face recognition with deep neural networks.

Object detectors are not perfect so you are bound to see some false-positives. The SSD algorithm works (at a very simplistic level) by dividing your image into boxes and classifying each of them, class-wise. Since your face most of the frame being close up to the camera, there are likely a large number of boxes that contain face-like regions. This would imply why you may see a detection adjacent to the real one.

“The model was trained in Caffe framework on some huge and available online dataset.” I asked him and it’s the WIDER face dataset. He blurred small <30px faces.

Does this algorithm do non-max suppression as well? Yes, the algorithm is internally doing NMS.

Face detection

<https://towardsdatascience.com/face-detection-models-which-to-use-and-why-d263e82c302c>

<https://github.com/opencv/opencv/tree/master/samples/dnn>

<https://pyimagesearch.com/2018/02/26/face-detection-with-opencv-and-deep-learning/>

<https://pyimagesearch.com/2021/04/26/face-detection-tips-suggestions-and-best-practices/>

<https://pyimagesearch.com/2017/11/06/deep-learning-opencvs-blobfromimage-works/>

<https://github.com/opencv/opencv/blob/master/samples/dnn/face_detector/how_to_train_face_detector.txt>

Face recognition

<https://pyimagesearch.com/2018/06/18/face-recognition-with-opencv-python-and-deep-learning/>

<https://pyimagesearch.com/2018/09/24/opencv-face-recognition/>

<https://www.cv-foundation.org/openaccess/content_cvpr_2015/app/1A_089.pdf>

<https://cmusatyalab.github.io/openface/models-and-accuracies/>

<https://pyimagesearch.com/2017/05/22/face-alignment-with-opencv-and-python/>

Face landmarks

<https://yinguobing.com/facial-landmark-localization-by-deep-learning-background/>

<https://github.com/yinguobing/cnn-facial-landmark>

<https://towardsdatascience.com/robust-facial-landmarks-for-occluded-angled-faces-925e465cbf2e>

<https://github.com/yinguobing/head-pose-estimation/tree/master/assets/pose_model>

<https://github.com/faust690226/cnn-facial-landmark-tutorial>

<https://www.academia.edu/35372127/Approaching_human_level_facial_landmark_localization_by_deep_learning>

<https://pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/>

Head pose

<https://github.com/yinguobing/head-pose-estimation>

<https://medium.com/analytics-vidhya/real-time-head-pose-estimation-with-opencv-and-dlib-e8dc10d62078>

<https://learnopencv.com/head-pose-estimation-using-opencv-and-dlib/#disqus_thread>

<https://docs.opencv.org/2.4/doc/tutorials/calib3d/camera_calibration/camera_calibration.html>

<https://github.com/niconielsen32/ComputerVision/blob/master/headPoseEstimation.py>

<https://www.youtube.com/watch?v=-toNMaS4SeQ>

People detector

<https://pyimagesearch.com/2018/08/13/opencv-people-counter/>

<https://viso.ai/deep-learning/object-detection/>

<https://towardsdatascience.com/picking-fast-people-detector-working-with-opencv-on-cpu-in-2021-ff8d752088af>

<https://vidishmehta204.medium.com/object-detection-using-ssd-mobilenet-v2-7ff3543d738d>

<https://arxiv.org/abs/1512.02325>

<https://blog.roboflow.com/training-a-tensorflow-object-detection-model-with-a-custom-dataset/>

<https://github.com/chuanqi305/MobileNet-SSD>

<https://github.com/weiliu89/caffe/tree/ssd>

<https://pyimagesearch.com/2017/09/11/object-detection-with-deep-learning-and-opencv/>

Eyes tracking

<https://medium.com/@stepanfilonov/tracking-your-eyes-with-python-3952e66194a6>

<https://stackoverflow.com/questions/61016954/controlling-contrast-and-brightness-of-video-stream-in-opencv-and-python>

<https://github.com/antoinelame/GazeTracking>

<https://stackoverflow.com/questions/3490727/what-are-some-methods-to-analyze-image-brightness-using-python>

Liveness

<https://pyimagesearch.com/2019/03/11/liveness-detection-with-opencv/>

<https://medium.com/codex/fake-webcam-for-your-online-meetings-with-python-755556d7667b>

<https://pyimagesearch.com/2017/04/24/eye-blink-detection-opencv-python-dlib/>