The vote is over, but the fight for net neutrality isn't. Show your support for a free and open internet. **Learn more**

CMU-Perceptual-Computing-Lab / **openpose**

OpenPose: Real-time multi-person keypoint detection library for body, face, and hands estimation

#openpose #computer-vision #machine-learning #multi-threading #cpp #cpp11 #caffe #opencv #human-pose-estimation #real-time #deep-learning #human-behavior-understanding #cvpr-2017

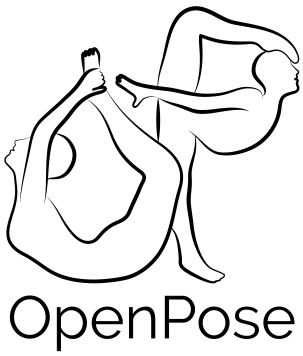
229 commits | **3** branches | **8** releases | **22** contributors

Branch: **master** | **New pull request** | **Create new file** | **Upload files** | **Find file** | **Clone or download**

gineshidalgo99 Changed body_59 colors & allowed COCO saving — Latest commit `b4a7a6c` a day ago

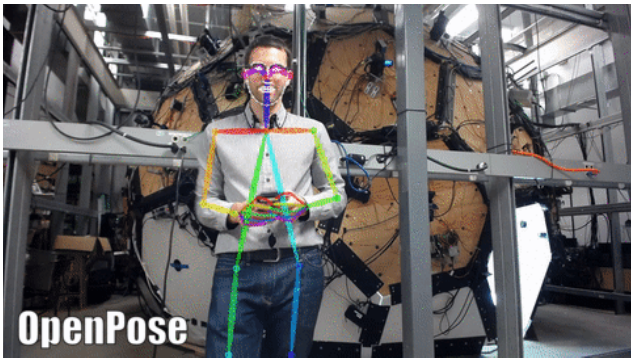
.github	Fixed cuda invalid configuration for >4k images (#324)	2 months ago
3rdparty	Updated Caffe	2 months ago
cmake	CMake generates libopenpose.so/.a	3 months ago
doc	Removed redundant function arguments	13 days ago
examples	PoseParameters splitted into 2 and its parameters into functions	13 days ago
examples_beta/openpose3d	PoseParameters splitted into 2 and its parameters into functions	13 days ago
include/openpose	Changed body_59 colors & allowed COCO saving	a day ago
models	Face Detection by OpenCV (#270)	3 months ago
src	Changed body_59 colors & allowed COCO saving	a day ago
ubuntu	Updated Caffe	2 months ago
windows	PoseParameters splitted into 2 and its parameters into functions	13 days ago
.gitignore	CMake support added (#234)	3 months ago
.travis.yml	CMake generates libopenpose.so/.a	3 months ago
CMakeLists.txt	CMake doesn't require numpy	2 months ago
LICENSE	Updated Caffe version license 1.0.0rc5 to 1.0.0	8 months ago
Makefile	OpenPose v1.2.0	2 months ago
README.md	Added floating heatmap saving	25 days ago

README.md



Linux
build passing

OpenPose represents the **first real-time multi-person system to jointly detect human body, hand, and facial keypoints (in total 130 keypoints) on single images**.



Functionality:

- **Real-time multi-person keypoint detection**.
  - 15 or **18-keypoint body estimation**. **Running time invariant to number of detected people**.
  - **2x21-keypoint hand** estimation. Currently, **running time depends** on **number of detected people**.
  - **70-keypoint face** estimation. Currently, **running time depends** on **number of detected people**.
- **Input**: Image, video, webcam, and IP camera. Included C++ demos to add your custom input.
- **Output**: Basic image + keypoint display/saving (PNG, JPG, AVI, ...), keypoint saving (JSON, XML, YML, ...), and/or keypoints as array class.
- Available: command-line demo, C++ wrapper, and C++ API.
- **OS**: Ubuntu (14, 16), Windows (8, 10), Nvidia TX2.

Latest News

- Sep 2017: **CMake** installer and **IP camera** support!
- Jul 2017: **Windows portable demo**!
- Jul 2017: **Hands** released!
- Jun 2017: **Face** released!
- May 2017: **Windows** version!

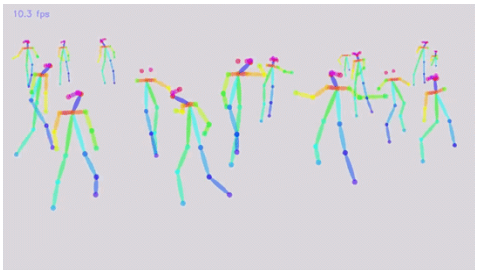
- Apr 2017: **Body** released!
- Check all the release notes.

Contents

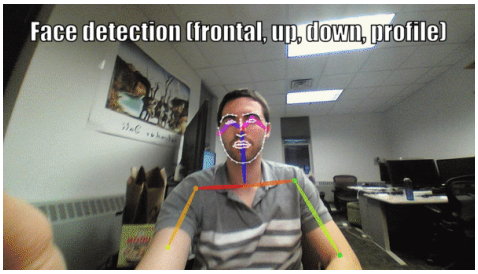
1. Latest News
2. Results
3. Installation, Reinstallation and Uninstallation
4. Quick Start
5. Output
6. Speeding Up OpenPose and Benchmark
7. Send Us Failure Cases and Feedback!
8. Authors and Contributors
9. Citation
10. License

Results

Body Estimation



Body, Face, and Hands Estimation



Body and Hands Estimation

Installation, Reinstallation and Uninstallation

See doc/installation.md for instructions on how to build from source or how to download our portable binaries.

Quick Start

Most users do not need the OpenPose C++ API, but they can simply use the basic Demo and/or OpenPose Wrapper.

- **Demo**: To easily process images/video/webcam and display/save the results. See doc/demo_overview.md. E.g. run OpenPose in a video with:

```
# Ubuntu
./build/examples/openpose/openpose.bin --video examples/media/video.avi
:: Windows - Portable Demo
bin\OpenPoseDemo.exe --video examples\media\video.avi
```

- **OpenPose Wrapper**: If you want to read a specific input, and/or add your custom post-processing function, and/or implement your own display/saving, check the `Wrapper` tutorial on examples/tutorial_wrapper/. You can create your custom code on examples/user_code/ and quickly compile it by using `make all` in the OpenPose folder (assuming Makefile installer).
- **OpenPose C++ API**: See doc/library_introduction.md.
- **Adding an extra module**: Check doc/library_add_new_module.md.
- **Standalone face or hand detector**:
  - **Face** keypoint detection **without body** keypoint detection: If you want to speed it up (but also reduce amount of detected faces), check the OpenCV-face-detector approach in doc/standalone_face_or_hand_keypoint_detector.md.
  - **Use your own face/hand detector**: You can use the hand and/or face keypoint detectors with your own face or hand detectors, rather than using the body detector. E.g. useful for camera views at which the hands are visible but not the body (OpenPose detector would fail). See doc/standalone_face_or_hand_keypoint_detector.md.
- **Library dependencies**: OpenPose uses default Caffe and OpenCV, as well as any Caffe dependency. The demos additionally use GFlags. It could easily be ported to other deep learning frameworks (Tensorflow, Torch, ...). Feel free to make a pull request if you implement any of those!

Output

Output (format, keypoint index ordering, etc.) in doc/output.md.

Speeding Up OpenPose and Benchmark

Check the OpenPose Benchmark and some hints to speed up OpenPose on doc/installation.md#faq.

Send Us Failure Cases and Feedback!

Our library is open source for research purposes, and we want to continuously improve it! So please, let us know if...

1. ... you find videos or images where OpenPose does not seems to work well. Feel free to send them to openposecmu@gmail.com (email only for failure cases!), we will use them to improve the quality of the algorithm!
2. ... you find any bug (in functionality or speed).
3. ... you added some functionality to some class or some new Worker subclass which we might potentially incorporate.
4. ... you know how to speed up or improve any part of the library.
5. ... you have a request about possible functionality.
6. ... etc.

Just comment on GitHub or make a pull request and we will answer as soon as possible! Send us an email if you use the library to make a cool demo or YouTube video!

Authors and Contributors

OpenPose is authored by Gines Hidalgo, Zhe Cao, Tomas Simon, Shih-En Wei, Hanbyul Joo, and Yaser Sheikh. Currently, it is being maintained by Gines Hidalgo and Bikramjot Hanzra. The original CVPR 2017 repo includes Matlab and Python versions, as well as the training code. The body pose estimation work is based on the original ECCV 2016 demo.

In addition, OpenPose would not be possible without the CMU Panoptic Studio dataset.

We would also like to thank all the people who helped OpenPose in any way. The main contributors are listed in doc/contributors.md.

Citation

Please cite these papers in your publications if it helps your research (the face keypoint detector was trained using the same procedure described in [Simon et al. 2017]):

```
@inproceedings{cao2017realtime,
  author = {Zhe Cao and Tomas Simon and Shih-En Wei and Yaser Sheikh},
  booktitle = {CVPR},
  title = {Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields},
  year = {2017}
}

@inproceedings{simon2017hand,
  author = {Tomas Simon and Hanbyul Joo and Iain Matthews and Yaser Sheikh},
  booktitle = {CVPR},
  title = {Hand Keypoint Detection in Single Images using Multiview Bootstrapping},
  year = {2017}
}

@inproceedings{wei2016cpm,
  author = {Shih-En Wei and Varun Ramakrishna and Takeo Kanade and Yaser Sheikh},
  booktitle = {CVPR},
  title = {Convolutional pose machines},
  year = {2016}
}
```

License

OpenPose is freely available for free non-commercial use, and may be redistributed under these conditions. Please, see the license for further details. Interested in a commercial license? Check this link. For commercial queries, contact Yaser Sheikh.