

---

# **openpose-plus**

**tensorlayer**

**Jun 07, 2020**



**CONTENTS**

<b>Index</b>	<b>3</b>
--------------	----------



## interface paf\_processor

### Public Functions

std::vector<human\_t> **operator()** (const float\*, const float\*, bool) = 0

## interface pose\_detection\_runner

### Public Functions

void **operator()** (const std::vector<void\*> &inputs, const std::vector<void\*> &outputs, int batch-Size = 1) = 0

*paf\_processor* \***create\_paf\_processor** (int input\_height, int input\_width, int height, int width, int n\_joins, int n\_connections, int gauss\_kernel\_size)

Create a *paf\_processor*.

### Parameters

- input\_height: height of feature maps
- input\_width: width of feature maps
- height: height of output, usually equal to input image height
- width: width of output, usually equal to input image width
- n\_joins: must be 19 for now
- n\_connections: must be 19 for now
- gauss\_kernel\_size: gauss kernel size for smooth the feature maps after resize

*pose\_detection\_runner* \***create\_pose\_detection\_runner** (const std::string &model\_file, int input\_height, int input\_width, int max\_batch\_size, bool use\_f16)

Creates a *pose\_detection\_runner*.

### Parameters

- model\_file: path to the exported uff model file
- input\_height: height of the input image
- input\_width: width of the input image
- max\_batch\_size: max batch size
- use\_f16: if use float 16



## INDEX

### C

`create_paf_processor (C++ function), 1`  
`create_pose_detection_runner (C++ function), 1`

### P

`paf_processor (C++ class), 1`  
`paf_processor::operator() (C++ function), 1`  
`pose_detection_runner (C++ class), 1`  
`pose_detection_runner::operator() (C++ function), 1`