

## Установка и эксплуатация ПО

Для запуска и эксплуатации ПО необходимо установить на рабочий компьютер интерпретатор Python. Для его установки необходимо перейти на страницу <https://www.python.org/downloads/> и найдем ссылку на загрузку последней версии языка.

Кроме того, необходимо установить PIP (Это утилита командной строки, которая позволяет устанавливать, переустанавливать и деинсталлировать PyPI пакеты простой командой `pip`):

### Как установить PIP на Windows

Следующие инструкции подойдут для Windows 7, Windows 8.1 и Windows 10.

1. Скачайте установочный скрипт `get-pip.py`. Если у вас Python 3.2, версия `get-pip.py` должны быть такой же. В любом случае щелкайте правой кнопкой мыши на ссылке и нажмите “Сохранить как...” и сохраните скрипт в любую безопасную папку, например в “Загрузки”.
2. Откройте командную строку и перейдите к каталогу с файлом `get-pip.py`.
3. Запустите следующую команду: `python get-pip.py`

### Как установить PIP на Mac

Современные версии Mac идут с установленными Python и PIP. Так или иначе версия Python устаревает, а это не лучший вариант для серьёзного разработчика. Так что рекомендуется установить актуальные версии Python и PIP.

Если вы хотите использовать родную систему Python, но у вас нет доступного PIP, его можно установить следующей командой через терминал:

```
sudo easy_install pip
```

Если вы предпочитаете более свежие версии Python, используйте Homebrew. Следующие инструкции предполагают, что Homebrew уже установлен и готов к работе.

Установка Python с помощью Homebrew производится посредством одной команды:

```
brew install python
```

Будет установлена последняя версия Python, в которую может входить PIP. Если после успешной установки пакет недоступен, необходимо выполнить перелинковку Python следующей командой:

```
brew unlink python && brew link python
```

### Как установить PIP на Linux

Если у вас дистрибутив Linux с уже установленным на нем Python, то скорее всего возможно установить PIP, используя системный пакетный менеджер. Это более удачный способ, потому что системные версии Python не слишком хорошо работают со скриптом `get-pip.py`, используемым в Windows и Mac.

```
Advanced Package Tool (Python 2.x)
sudo apt-get install python-pip
Advanced Package Tool (Python 3.x)
sudo apt-get install python3-pip
pacman Package Manager (Python 2.x)
sudo pacman -S python2-pip
pacman Package Manager (Python 3.x)
sudo pacman -S python-pip
Yum Package Manager (Python 2.x)
sudo yum upgrade python-setuptools
sudo yum install python-pip python-wheel
Yum Package Manager (Python 3.x)
sudo yum install python3 python3-wheel
Dandified Yum (Python 2.x)
sudo dnf upgrade python-setuptools
sudo dnf install python-pip python-wheel
Dandified Yum (Python 3.x)
sudo dnf install python3 python3-wheel
Zypper Package Manager (Python 2.x)
sudo zypper install python-pip python-setuptools python-wheel
Zypper Package Manager (Python 3.x)
sudo zypper install python3-pip python3-setuptools python3-wheel
```

---

После установки необходимо запустить файл *requirements.txt* находящийся в корне проекта. Он содержит все необходимые пакеты:

```
numpy~=1.21.5
pandas~=1.3.5
torch==1.11.0
torchvision==0.12.0
opencv-python~=4.5.5.64
dlib~=19.24.0
```

*requirements.txt* устанавливается командой `pip install -r </path/to/requirements.txt>`

Примеры в *examples* надо запускать из самой папки *examples*

Для перехода по директориям используется команда `cd <название директории/путь к директории>`

## Examples

This is an example of cvflow working with detection/segmentation/classification models.

Example of running a script:

Usbcam:

```
python3 [script name] --usbcam /dev/video0 --font ../fonts/OpenSans-VariableFont_wdth,wght.ttf --fsz 1280,1920
```

Videofile:

```
python3 [script name] --videofile {file_path},{file_path} --font ../fonts/OpenSans-VariableFont_wdth,wght.ttf --fsz 1280,1920
```

## Commands list

- --usbcam <path\_to\_device>,<path\_to\_device>,... - list of sources for reading the video stream from the camera
- --videofile <path\_to\_file>,<path\_to\_file>,... - list of sources for reading a video stream from a video file
- -c or --confidence <threshold\_value> - threshold value for the model
- -f or --font <path\_to\_font> - the path to the font file. File format "\*.ttf"
- --tsize - <n, k> - the format of the video stream grid on the output.
- --fsize <H, W> - the resolution of the image to output.
- -d or --device <device\_type> - The device on which the conveyor will work. Available: cpu, cuda.
- -l or --line - coordinates of the line intersecting the objects to be counted.

## Stereo distance tracker example

### Creating ROI-bounding boxes

1. Run `giu/roi_getter.py` with parameters: -n to set number of objects and -f to set filepath.
2. Press 'd' to skip 10 frames
3. Press 's' to start selection
4. Press 'n' to add each bbox. When you will select n objects, `conf.txt` will be written automatically.

### Running

5. Download the [weights](#) and put it to `tests/test_data` folder.
6. Run `stereo_example.py` for demonstration. You can use data from `tests/test_data` for example. NB! Model works only with cuda. Please provide at least two data sources.

## Objects counter example

### Create configuration of counting lines

1. Run command:

```
python3 config_gui.py -v <video_file> -n <path_to_config_file>
```

2. Press save and close the window.

### Running

```
python3 objects_counter_example.py --videofile <path_to_video> -f <path_to_fonts> -c <path_to_config_file>
```