

---

# **FastDeploy**

*Release develop*

**PaddlePaddle**

**Sep 17, 2022**



<b>1</b>		<b>3</b>
1.1	CPU . . . . .	3
1.2	GPU . . . . .	5
1.3	Jetson . . . . .	8
<b>2</b>		<b>11</b>
2.1	. . . . .	11
2.2	Runtime . . . . .	14



- GitHub Repository: <https://github.com/PaddlePaddle/FastDeploy>
- Issue Feedback: <http://www.github.com/PaddlePaddle/FastDeploy/issues>
- Contact Us: [fastdeploy@baidu.com](mailto:fastdeploy@baidu.com)



FastDeploy

## 1.1 CPU

FastDeployCPU

### 1.1.1

FastDeployVisionText,

- ReleaseFastDeploy
- Nightly buildFastDeploy(Linux-x64Windows-x64)

## Python

Release

```
pip install fastdeploy-python -f https://www.paddlepaddle.org.cn/whl/fastdeploy.html
```

Nightly build

```
pip install fastdeploy-python -f https://www.paddlepaddle.org.cn/whl/fastdeploy_nightly_
↪build.html
```

## C++SDK

nightly build

### 1.1.2 C++ SDK

#### Linux & Mac

Linux

- gcc/g++ >= 5.4(8.2)
- cmake >= 3.18.0

```
git clone https://github.com/PaddlePaddle/FastDeploy.git
cd FastDeploy
mkdir build && cd build
cmake .. -DENABLE_ORT_BACKEND=ON \
        -DENABLE_PADDLE_BACKEND=ON \
        -DENABLE_OPENVINO_BACKEND=ON \
        -DCMAKE_INSTALL_PREFIX=${PWD}/compiled_fastdeploy_sdk \
        -DENABLE_VISION=ON
make -j12
make install
```

### Windows

Windows

- Windows 10/11 x64
- Visual Studio 2019

Windowsx64 Native Tools Command Prompt for VS 2019

```
git clone https://github.com/PaddlePaddle/FastDeploy.git
cd FastDeploy
mkdir build && cd build
cmake .. -G "Visual Studio 16 2019" -A x64 \
        -DENABLE_ORT_BACKEND=ON \
        -DENABLE_PADDLE_BACKEND=ON \
        -DENABLE_OPENVINO_BACKEND=ON \
        -DENABLE_VISION=ON \
        -DCMAKE_INSTALL_PREFIX="D:\Paddle\compiled_fastdeploy"
msbuild fastdeploy.sln /m /p:Configuration=Release /p:Platform=x64
msbuild INSTALL.vcxproj /m /p:Configuration=Release /p:Platform=x64
```

CMAKE\_INSTALL\_PREFIXC++

### 1.1.3 Python

- gcc/g++ >= 5.4(8.2)
- cmake >= 3.18.0
- python >= 3.6

## Linux & Mac

```
git clone https://github.com/PaddlePaddle/FastDeploy.git
cd FastDeploy/python
export ENABLE_ORT_BACKEND=ON
export ENABLE_PADDLE_BACKEND=ON
export ENABLE_OPENVINO_BACKEND=ON
export ENABLE_VISION=ON

python setup.py build
python setup.py bdist_wheel
```

## Windows

- Windows 10/11 x64
- Visual Studio 2019
- python >= 3.6

Windowsx64 Native Tools Command Prompt for VS 2019

```
git clone https://github.com/PaddlePaddle/FastDeploy.git
cd FastDeploy/python
set ENABLE_ORT_BACKEND=ON
set ENABLE_PADDLE_BACKEND=ON
set ENABLE_OPENVINO_BACKEND=ON
set ENABLE_VISION=ON

python setup.py build
python setup.py bdist_wheel
```

FastDeploy/python/distwheel pip install

FastDeploy/python build.setuptools-cmake-build

## 1.2 GPU

FastDeployGPU Paddle Inference ONNX Runtime TensorRT Linux & Windows GPU CPU OpenVINO

GPUWITH\_GPU ON CUDA\_DIRECTORY TensorRT TRT\_DIRECTORY

### 1.2.1

FastDeployVisionText,

- ReleaseFastDeploy
- Nightly buildFastDeploy(Linux-x64Windows-x64)

### Python

Release

```
pip install fastdeploy-gpu-python -f https://www.paddlepaddle.org.cn/whl/fastdeploy.html
```

Nightly build

```
pip install fastdeploy-gpu-python -f https://www.paddlepaddle.org.cn/whl/fastdeploy_
↪nightly_build.html
```

### C++SDK

nightly build

### 1.2.2 C++ SDK

#### Linux

Linux

- gcc/g++ >= 5.4(8.2)
- cmake >= 3.18.0
- cuda >= 11.2
- cudnn >= 8.2

```
git clone https://github.com/PaddlePaddle/FastDeploy.git
cd FastDeploy
mkdir build && cd build
cmake .. -DENABLE_ORT_BACKEND=ON \
        -DENABLE_PADDLE_BACKEND=ON \
        -DENABLE_OPENVINO_BACKEND=ON \
        -DENABLE_TRT_BACKEND=ON \
        -DWITH_GPU=ON \
        -DTRT_DIRECTORY=/Paddle/TensorRT-8.4.1.5 \
        -DCUDA_DIRECTORY=/usr/local/cuda \
        -DCMAKE_INSTALL_PREFIX=${PWD}/compiled_fastdeploy_sdk \
        -DENABLE_VISION=ON
make -j12
make install
```

## Windows

### Windows

- Windows 10/11 x64
- Visual Studio 2019
- cuda >= 11.2
- cudnn >= 8.2

### Windowsx64 Native Tools Command Prompt for VS 2019

```
git clone https://github.com/PaddlePaddle/FastDeploy.git
cd FastDeploy
mkdir build && cd build
cmake .. -G "Visual Studio 16 2019" -A x64 \
  -DENABLE_OR_T_BACKEND=ON \
  -DENABLE_PADDLE_BACKEND=ON \
  -DENABLE_OPENVINO_BACKEND=ON \
  -DENABLE_TRT_BACKEND=ON
  -DENABLE_VISION=ON \
  -DWITH_GPU=ON \
  -DTRT_DIRECTORY="D:\Paddle\TensorRT-8.4.1.5" \
  -DCUDA_DIRECTORY="C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.2" \
  -DCMAKE_INSTALL_PREFIX="D:\Paddle\compiled_fastdeploy"
msbuild fastdeploy.sln /m /p:Configuration=Release /p:Platform=x64
msbuild INSTALL.vcxproj /m /p:Configuration=Release /p:Platform=x64
```

CMAKE\_INSTALL\_PREFIX++

## 1.2.3 Python

### Linux

- gcc/g++ >= 5.4(8.2)
- cmake >= 3.18.0
- python >= 3.6
- cuda >= 11.2
- cudnn >= 8.2

```
git clone https://github.com/PaddlePaddle/FastDeploy.git
cd FastDeploy/python
export ENABLE_OR_T_BACKEND=ON
export ENABLE_PADDLE_BACKEND=ON
export ENABLE_OPENVINO_BACKEND=ON
export ENABLE_VISION=ON
export ENABLE_TRT_BACKEND=ON
export WITH_GPU=ON
```

(continues on next page)

(continued from previous page)

```
export TRT_DIRECTORY=/Paddle/TensorRT-8.4.1.5
export CUDA_DIRECTORY=/usr/local/cuda

python setup.py build
python setup.py bdist_wheel
```

## Windows

- Windows 10/11 x64
- Visual Studio 2019
- python >= 3.6
- cuda >= 11.2
- cudnn >= 8.2

Windowsx64 Native Tools Command Prompt for VS 2019

```
git clone https://github.com/PaddlePaddle/FastDeploy.git
cd FastDeploy/python
export ENABLE_ORT_BACKEND=ON
export ENABLE_PADDLE_BACKEND=ON
export ENABLE_OPENVINO_BACKEND=ON
export ENABLE_VISION=ON
export ENABLE_TRT_BACKEND=ON
export WITH_GPU=ON
export TRT_DIRECTORY="D:\Paddle\TensorRT-8.4.1.5"
export CUDA_DIRECTORY="C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.2"

python setup.py build
python setup.py bdist_wheel
```

FastDeploy/python/distwheelpip install

FastDeploy/pythonbuild.setuptools-cmake-build

## 1.3 Jetson

FastDeployJetsonONNX Runtime CPU TensorRT GPU

### 1.3.1 C++ SDK

- gcc/g++ >= 5.4(8.2)
- cmake >= 3.18.0
- jetpack >= 4.6

```
git clone https://github.com/PaddlePaddle/FastDeploy.git
cd FastDeploy
mkdir build && cd build
cmake .. -DENABLE_ORT_BACKEND=ON \
        -DENABLE_TRT_BACKEND=ON \
        -DWITH_GPU=ON \
        -DBUILD_ON_JETSON=ON \
        -DCMAKE_INSTALL_PREFIX=${PWD}/compiled_fastdeploy_sdk \
        -DENABLE_VISION=ON
make -j8
make install
```

CMAKE\_INSTALL\_PREFIX++

### 1.3.2 Python

- gcc/g++ >= 5.4(8.2)
- cmake >= 3.18.0
- jetpack >= 4.6
- python >= 3.6

```
git clone https://github.com/PaddlePaddle/FastDeploy.git
cd FastDeploy/python
export ENABLE_ORT_BACKEND=ON
export ENABLE_TRT_BACKEND=ON
export WITH_GPU=ON
export ENABLE_VISION=ON
export BUILD_ON_JETSON=ON

python setup.py build
python setup.py bdist_wheel
```

FastDeploy/python/distwheelpip install

FastDeploy/pythonbuild.setuptools-cmake-build



FastDeployNLPRuntimeAPI

## 2.1

FastDeployNLP100+

### 2.1.1 Python

FastDeployFastDeployFastDeploy

PaddleDetectionPPYOLOECPU

1.

```
import fastdeploy as fd

model_url = "https://bj.bcebos.com/paddlehub/fastdeploy/pyyoloe_crn_l_300e_coco.tgz"
image_url = "https://gitee.com/paddlepaddle/PaddleDetection/raw/release/2.4/demo/
->000000014439.jpg"
fd.download_and_decompress(model_url, path=".")
fd.download(image_url, path=".")
```

2.

- FastDeploy/examples
- APIAPIFastDeployPython API

```
model_file = "pyyoloe_crn_l_300e_coco/model.pdmodel"
params_file = "pyyoloe_crn_l_300e_coco/model.pdiparams"
infer_cfg_file = "pyyoloe_crn_l_300e_coco/infer_cfg.yml"
model = fd.vision.detection.PPYOLOE(model_file, params_file, infer_cfg_file)
```

3.

```
import cv2
im = cv2.imread("0000000014439.jpg")

result = model.predict(im)
print(result)
```

4.

```
vis_im = fd.vision.visualize.vis_detection(im, result, score_threshold=0.5)
cv2.imwrite("vis_image.jpg", vis_im)
```

### 2.1.2 C++

FastDeploy C++FastDeployFastDeploy

PaddleDetectionPPYOLOECPU

1.

```
wget https://bj.bcebos.com/paddlehub/fastdeploy/ppyoloe_crn_l_300e_coco.tgz
wget https://gitee.com/paddlepaddle/PaddleDetection/raw/release/2.4/demo/0000000014439.jpg
tar xvf ppyoloe_crn_l_300e_coco.tgz
```

2. C++

C++infer\_demo.cc

```
#include "fastdeploy/vision.h"
int main() {
    std::string model_file = "ppyoloe_crn_l_300e_coco/model.pdmodel";
    std::string params_file = "ppyoloe_crn_l_300e_coco/model.pdiparams";
    std::string infer_cfg_file = "ppyoloe_crn_l_300e_coco/infer_cfg.yml";
    auto model = fastdeploy::vision::detection::PPYOLOE(model_file, params_file, infer_cfg_
↪file);

    assert(model.Initialized()); //

    cv::mat im = cv::imread("0000000014439.jpg");
    fastdeploy::vision::DetectionResult result;

    assert(model.Predict(im)); //

    std::cout << result << std::endl;

    cv::mat vis_im = fastdeploy::vision::Visualize::VisDetection(im, result, 0.5);
    //
```

(continues on next page)

(continued from previous page)

```

cv::imwrite("vis_result.jpg", vis_im);

return 0;
}

```

### 3. CMakeList.txt

FastDeployg++cmake

FastDeploy C++ SDK/Paddle/Downloadfastdeploy\_cpp\_sdkFASTDEPLOY\_INCSFASTDEPLOY\_LIBS

```
include(/Paddle/Download/fastdeploy_cpp_sdk/FastDeploy.cmake)
```

```

PROJECT(infer_demo C CXX)
CMAKE_MINIMUM_REQUIRED (VERSION 3.12)

include(/Paddle/Download/fastdeploy_cpp_sdk/FastDeploy.cmake)

# FastDeploy
include_directories(${FASTDEPLOY_INCS})

add_executable(infer_demo ${PROJECT_SOURCE_DIR}/infer_demo.cc)
target_link_libraries(infer_demo ${FASTDEPLOY_LIBS})

```

### 4.

infer\_demo.ccCMakeLists.txt

#### Linux & Mac

infer\_demo.ccCmakeLists.txt

```

mkdir build & cd build
cmake ..
make -j

```

cmakeFastDeployNoticeFastDeploycommand

```

===== Notice =====
After compiled binary executable file, please add the following path to environment,
↳ execute the following command,

export LD_LIBRARY_PATH=/Paddle/Download/fastdeploy_cpp_sdk/third_libs/install/
↳paddle2onnx/lib:/Paddle/Download/fastdeploy_cpp_sdk/third_libs/install/opencv/lib:/
↳Paddle/Download/fastdeploy_cpp_sdk/third_libs/install/onnxruntime/lib:/Paddle/Download/
↳fastdeploy_cpp_sdk/lib:${LD_LIBRARY_PATH}
=====

```

```
./infer_demo
```

### Windows

Windowsx64 Native Tools Command Prompt for VS 2019infer\_demo.ccCMakeLists.txt

```
mkdir build
cd build
cmake .. -G "Visual Studio 16 2019" -A x64
msbuild infer_demo.sln /m /p:Configuration=Release /p:Platform=x64
```

```
cmakeFastDeployNoticeFastDeployexecommand
```

```
===== Notice =====
```

```
build/Releaseinfer_demo.exe
```

```
Release\infer_demo.exe
```

## 2.2 Runtime

FastDeploy RuntimePaddle/ONNX .. toctree:

```
:caption:
:maxdepth: 2
:titlesonly:

python.md
cpp.md
```