

MNN quick start guide

1.introduction

MNN is a highly efficient and lightweight deep learning framework. It supports inference and training of deep learning models and has industry-leading performance for inference and training on-device. At present, MNN has been integrated into more than 30 apps of Alibaba Inc, such as Taobao, Tmall, Youku, DingTalk, Xianyu, etc., covering more than 70 usage scenarios such as live broadcast, short video capture, search recommendation, product searching by image, interactive marketing, equity distribution, security risk control. In addition, MNN is also used on embedded devices, such as IoT.

MNN is a completely open-source framework that can be obtained from GitHub <https://github.com/alibaba/MNN>.

2. quick start

```
# 2.1 Amlogic MNN file list
├── DOC
│   ├── MNN_quick_start_guid.md
├── MNN_DEMO
│   ├── mnn_test          # object detection demo
│   ├── multiPose.out     # pose detection demo
│   └── resource
│       ├── mobilenetv2-12.mnn
│       ├── model-mobilenet_v1_075_fixed.mnn
│       ├── pose_input1.jpeg
│       ├── pose_input2.jpeg
│       ├── test_input.bin
│       └── test_input.jpeg
└── MNN_lib
    └── libMNN.so          # MNN library

# 2.1 prepare MNN library & demo
cd C:\path\of\MNN
adb push MNN_lib\libMNN.so /usr/lib/
adb push MNN_DEMO /data

# 2.2 run demo
adb shell "chmod +x /data/MNN_DEMO/mnn_test; chmod +x
/data/MNN_DEMO/multiPose.out"

adb shell "cd /data/MNN_DEMO; ./mnn_test resource/mobilenetv2-12.mnn
resource/test_input.jpeg 1"
adb shell "cd /data/MNN_DEMO;export NN_USE_OPENCL=1; ./mnn_test
resource/mobilenetv2-12.mnn resource/test_input.jpeg 1"
adb shell "cd /data/MNN_DEMO;export NN_USE_VULKAN=1; ./mnn_test
resource/mobilenetv2-12.mnn resource/test_input.jpeg 1"

adb shell "cd /data/MNN_DEMO;./multiPose.out resource/model-
mobilenet_v1_075_fixed.mnn resource/pose_input1.jpeg pose_output1.jpg"
```

3.MNN_DEMO

3.1 mnn_test

a. object detection demo

b. usage

```
Usage(use input file): ./mnn_test [model path] [input0] [input1]...[inputn]
[loop time]
Usage(no input file): ./mnn_test [model path] [loop time]
```

c. log

```
# default use CPU backend
C:\path\of\MNN>adb shell "cd /data/MNN_DEMO; ./mnn_test resource/mobilenetv2-12.mnn resource/test_input.jpeg 1"
    SET NN_SAVE_OUTPUT = 0
    SET NN_NO_INPUT = 0
    SET NN_USE_VULKAN = 0
    SET NN_USE_OPENCL = 0
    SET NN_THREAD_NUM = 2
    Set precision type: 0
    Open Model resource/mobilenetv2-12.mnn
    createFromBuffer time - 86.6915ms,
    CPU Group: [ 0  1  2  3  4 ], 100000 - 1800000
    The device supports: i8sdot:0, fp16:0, i8mm: 0, sve2: 0, sme2: 0
    createSession time - 48.8535ms,
    input name : resource/test_input.jpeg
    model input data from picture
    Image preprocessed: 3x224x224
    Float buffer size: 150528 elements (602112 bytes)
    model input[0] :602112
    set   input[0] :602112
    set input time - 23.1208ms,
    create output Host Tensor time - 0.02375ms,
    execution time CPU - 86.7128ms, loop time = 1
    top 0:score--19.153988,class--1
    top 1:score--10.447388,class--115
    top 2:score--9.974771,class--27
    top 3:score--9.305929,class--139
    top 4:score--8.774748,class--124
    get output time - 0.088625ms

#use GPU OpenCL backend
C:\path\of\MNN>adb shell "cd /data/MNN_DEMO;export NN_USE_OPENCL=1;
./mnn_test resource/mobilenetv2-12.mnn resource/test_input.jpeg 1"
    SET NN_SAVE_OUTPUT = 0
    SET NN_NO_INPUT = 0
    SET NN_USE_VULKAN = 0
    SET NN_USE_OPENCL = 1
    SET NN_THREAD_NUM = 2
    Set precision type: 0
```

```

Open Model resource/mobilenetv2-12.mnn
createFromBuffer time - 85.9218ms,
CPU Group: [ 0 1 2 3 4 ], 100000 - 1800000
The device supports: i8sdot:0, fp16:0, i8mm: 0, sve2: 0, sme2: 0
createSession time - 21910.6ms,
input name : resource/test_input.jpeg
model input data from picture
Image preprocessed: 3x224x224
Float buffer size: 150528 elements (602112 bytes)
model input[0] :602112
set input[0] :602112
set input time - 274.872ms,
create output Host Tensor time - 0.036458ms,
execution time GPU(OpenCL) - 3.47083ms, loop time = 1
top 0:score--19.156250,class--1
top 1:score--10.375000,class--115
top 2:score--9.929688,class--27
top 3:score--9.359375,class--139
top 4:score--8.796875,class--124
get output time - 259.305ms

```

#use GPU Vulkan backend

```

C:\path\of\MNN>adb shell "cd /data/MNN_DEMO;export NN_USE_VULKAN=1; ./mnn_test
resource/mobilenetv2-12.mnn resource/test_input.jpeg 1"

```

```

SET NN_SAVE_OUTPUT = 0
SET NN_NO_INPUT = 0
SET NN_USE_VULKAN = 1
SET NN_USE_OPENCL = 0
SET NN_THREAD_NUM = 2
Set precision type: 0
Open Model resource/mobilenetv2-12.mnn
createFromBuffer time - 84.7033ms,
CPU Group: [ 0 1 2 3 4 ], 100000 - 1800000
The device supports: i8sdot:0, fp16:0, i8mm: 0, sve2: 0, sme2: 0
createSession time - 11937.8ms,
input name : resource/test_input.jpeg
model input data from picture
Image preprocessed: 3x224x224
Float buffer size: 150528 elements (602112 bytes)
model input[0] :602112
set input[0] :602112
set input time - 22.4397ms,
create output Host Tensor time - 0.029666ms,
execution time GPU(Vulkan) - 72.6858ms, loop time = 1
top 0:score--19.187500,class--1
top 1:score--10.406250,class--115
top 2:score--9.976562,class--27
top 3:score--9.359375,class--139
top 4:score--8.835938,class--124
get output time - 1.47467ms

```

3.2 multiPose.out

a. pose detection demo

b. usage:

```
./multiPose.out [model path] [input path] [output path]
```

c. log

```
C:\path\of\MNN>adb shell "cd /data/MNN_DEMO;./multiPose.out resource/model-  
mobilenet_v1_075_fixed.mnn resource/pose_input1.jpeg pose_output1.jpg"  
CPU Group: [ 0 1 2 3 4 ], 100000 - 1800000  
The device supports: i8sdot:0, fp16:0, i8mm: 0, sve2: 0, sme2: 0  
main, 381, cost time: 481.804993 ms  
main, 405, cost time: 1.133000 ms
```

d. result show



4.MNN API

please refers: <https://mnn-docs.readthedocs.io/en/latest/#>