Step 1: Prepare Repo source

```
!git clone --recursive --depth=1 https://github.com/Linzaer/Ultra-Light-Fast-Generic-Face-Det
```

```
Cloning into 'Ultra-Light-Fast-Generic-Face-Detector-1MB'...
remote: Enumerating objects: 276, done.
remote: Counting objects: 100% (276/276), done.
remote: Compressing objects: 100% (244/244), done.
remote: Total 276 (delta 41), reused 220 (delta 22), pack-reused 0
Receiving objects: 100% (276/276), 37.04 MiB | 24.16 MiB/s, done.
Resolving deltas: 100% (41/41), done.
Submodule 'ncnn/3rdparty/ncnn' (https://github.com/Tencent/ncnn) registered for path 'n
Cloning into '/content/Ultra-Light-Fast-Generic-Face-Detector-1MB/ncnn/3rdparty/ncnn'..
remote: Enumerating objects: 25478, done.
remote: Counting objects: 100% (641/641), done.
remote: Compressing objects: 100% (302/302), done.
remote: Total 25478 (delta 465), reused 443 (delta 339), pack-reused 24837
Receiving objects: 100% (25478/25478), 17.57 MiB | 22.77 MiB/s, done.
Resolving deltas: 100% (21303/21303), done.
Submodule path 'ncnn/3rdparty/ncnn': checked out '22a2be4e6cb9cc6ef596d5bb801923135c82a
```

Step 2: Prepare input image for inference

```
%cd /content/Ultra-Light-Fast-Generic-Face-Detector-1MB/MNN/imgs
!wget https://raw.githubusercontent.com/ultralytics/yolov5/master/data/images/zidane.jpg
```

```
/content/Ultra-Light-Fast-Generic-Face-Detector-1MB/MNN/imgs
--2022-04-09 10:33:54--  https://raw.githubusercontent.com/ultralytics/yolov5/master/da
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.110.133, 185
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.110.133|:44
HTTP request sent, awaiting response... 200 OK
Length: 168949 (165K) [image/jpeg]
Saving to: ‘zidane.jpg’

zidane.jpg          100%[===================>] 164.99K  --.-KB/s    in 0.02s

2022-04-09 10:33:54 (7.17 MB/s) - ‘zidane.jpg’ saved [168949/168949]
```

Step 3: Cd to MNN folder to build MNN lib

```
%cd /content/Ultra-Light-Fast-Generic-Face-Detector-1MB/MNN
```

```
/content/Ultra-Light-Fast-Generic-Face-Detector-1MB/MNN
```

Step 4: Check environment and dependency libs to make sure suitable version

- cmake (version >=3.10 is recommended)
- protobuf (version >= 3.0 is required)
- gcc (version >= 4.9 is required)

```
!cmake --version
```

```
cmake version 3.12.0

CMake suite maintained and supported by Kitware (kitware.com/cmake).
```

```
!protoc --version
```

```
libprotoc 3.0.0
```

```
!gcc --version
```

```
gcc (Ubuntu 7.5.0-3ubuntu1~18.04) 7.5.0
Copyright (C) 2017 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

Step 5: Comment line 37-38 in `./Ultra-Light-Fast-Generic-Face-Detector-1MB/MNN/src/main.cpp` to do not show output image when inference. Because of colab limitation, it does not create server to show output

Step 6: Run the following command to build MNN lib

```
!mkdir build && cd build && cmake .. && make -j4
```

```
-- The C compiler identification is GNU 7.5.0
-- The CXX compiler identification is GNU 7.5.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Found OpenCV: /usr (found version "3.2.0")
-- Configuring done
-- Generating done
-- Build files have been written to: /content/Ultra-Light-Fast-Generic-Face-Detector-1M
Scanning dependencies of target Ultra-face-mnn
[ 33%] Building CXX object CMakeFiles/Ultra-face-mnn.dir/src/main.cpp.o
[ 66%] Building CXX object CMakeFiles/Ultra-face-mnn.dir/src/UltraFace.cpp.o
[100%] Linking CXX executable Ultra-face-mnn
[100%] Built target Ultra-face-mnn
```

Step 7: Check to ensure build MNN evironment successfully

```
%cd build
```

```
/content/Ultra-Light-Fast-Generic-Face-Detector-1MB/MNN/build
```

```
!make
```

```
[100%] Built target Ultra-face-mnn
```

```
!./Ultra-face-mnn
```

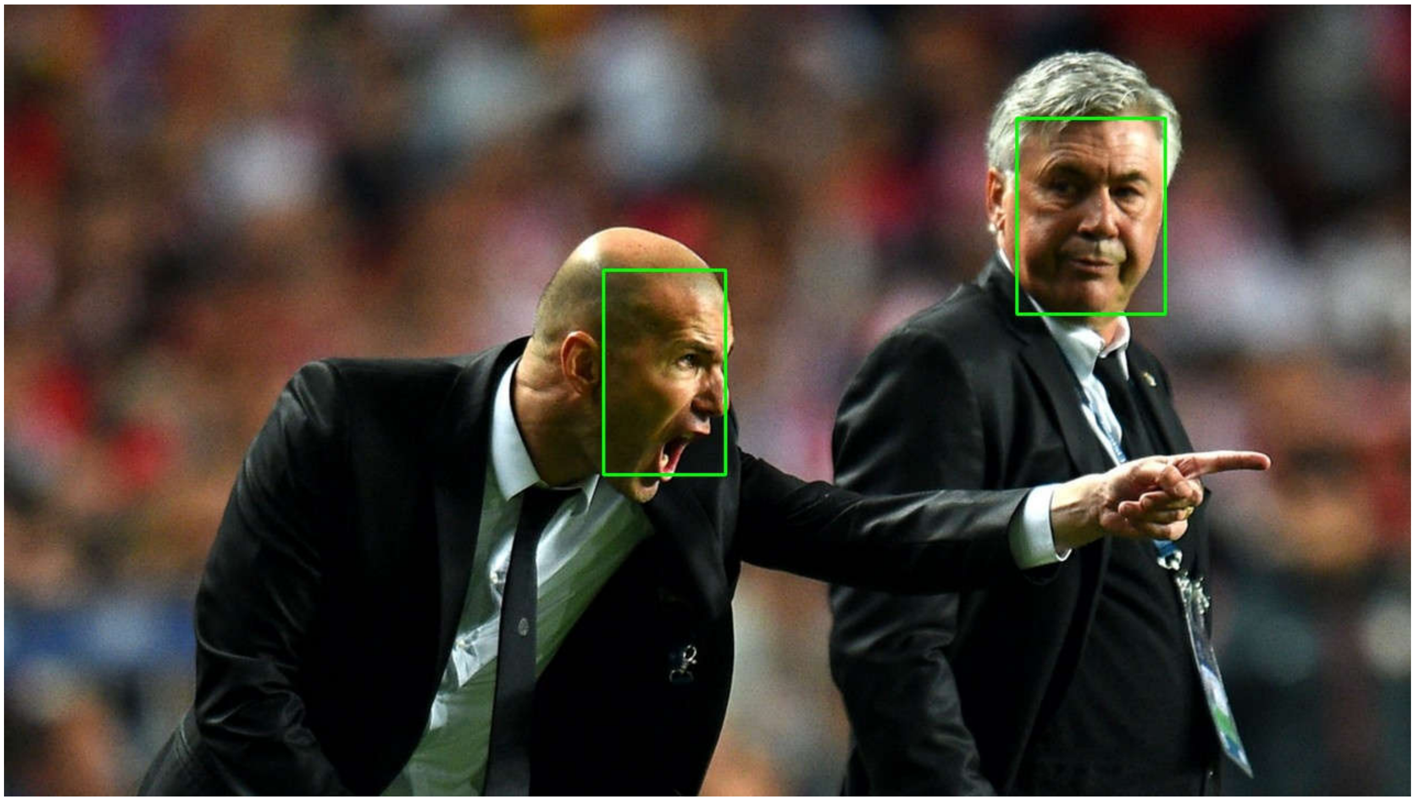
```
Usage: ./Ultra-face-mnn <mnn .mnn> [image files...]
```

Step 8: Use FP32 model and run in FP16 mode to inference input image. Output image will be saved at `./build/result*.jpg`

```
!./Ultra-face-mnn ../model/version-RFB/RFB-320.mnn  ../imgs/zidane.jpg
```

```
Processing ../imgs/zidane.jpg
inference time:0.0181551 s
all time: 0.385009 s
```

Output image



✓ 0s completed at 5:35 PM