

# DEPLOY KERAS MODEL WITH TENSORFLOW SERVING

Các bước triển khai Tensorflow Serving:

- Convert tensorflow/keras model (.h5, .ckpt) về định dạng **saved\_model** (.pb).
- Kiểm tra convert model.
- Khởi chạy **Tensorflow model server**.
- Tiền xử lý dữ liệu và request tới **Tensorflow model server**.
- Giao tiếp qua RESTful API & gRPC

## 1) Load và Export model

Load pre trained model (sử dụng mobilenet\_v2.h5) và export thành model định dạng saved\_model:

```
model.save(export_path)
```

Kết quả: thu được model theo định dạng **saved\_model** cùng các file thành phần.



NOTE: Đây là cách đơn giản để convert. Tuy nhiên, cách này đôi khi gặp sự cố. Để khắc phục "sự cố" này, có thể thực hiện convert file model h5 của keras sang dạng **frozen** của tensorflow, rồi sau đó convert tiếp 1 lần nữa frozen model sang định dạng saved\_model của tf-serving (đang tìm hiểu)

## 2) Check export model

Kiểm tra lại các thông tin meta-data của file **saved\_model.pb**

```
!saved_model_cli show \
--dir serving_models/mobilenet_v2/1 \
--tag_set serve \
--signature_def serving_default
```

Kết quả: Các thông tin về tag-set, signature\_def, inputs, outputs kèm kích thước và kiểu tương ứng như đã được khai báo lúc convert sang **saved\_model.pb** model.

```

2022-03-18 03:49:54.203184: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic
library 'libcudart.so.11.0'; dlderror: libcudart.so.11.0: cannot open shared object file: No such file or director
y; LD_LIBRARY_PATH: /usr/local/lib/python3.6/dist-packages/cv2/../../lib64:/usr/local/cuda/extras/CUPTI/lib64:/us
r/local/cuda/lib64:/usr/local/nvidia/lib:/usr/local/nvidia/lib64
2022-03-18 03:49:54.203212: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if yo
u do not have a GPU set up on your machine.
The given SavedModel SignatureDef contains the following input(s):
  inputs['input_1'] tensor_info:
    dtype: DT_FLOAT
    shape: (-1, 224, 224, 3)
    name: serving_default_input_1:0
The given SavedModel SignatureDef contains the following output(s):
  outputs['dense_1'] tensor_info:
    dtype: DT_FLOAT
    shape: (-1, 2)
    name: StatefulPartitionedCall:0
Method name is: tensorflow/serving/predict

```

### 3) gRPC (Google Remote Procedures Calls) vs RESTful (Representational State Transfer)

Thực hiện install **tensorflow\_model\_server** và **tensorflow-serving-api**

```

!echo "deb [arch=amd64]
http://storage.googleapis.com/tensorflow-serving-apt stable
tensorflow-model-server tensorflow-model-server-universal" |
tee /etc/apt/sources.list.d/tensorflow-serving.list && \
curl
https://storage.googleapis.com/tensorflow-serving-apt/tensor
flow-serving.release.pub.gpg | apt-key add -

```

```

!apt-get update && apt-get install tensorflow-model-server

```

#### Kết quả:

```

deb [arch=amd64] http://storage.googleapis.com/tensorflow-serving-apt stable tensorflow-model-server tensorflow-m
del-server-universal
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
100 2943 100 2943 0 0 18055 0 --:--:-- --:--:-- --:--:-- 18055
OK
Hit:1 http://storage.googleapis.com/tensorflow-serving-apt stable InRelease
Ign:2 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64 InRelease
Hit:3 http://archive.ubuntu.com/ubuntu bionic InRelease
Ign:4 https://developer.download.nvidia.com/compute/machine-learning/repos/ubuntu1804/x86_64 InRelease
Hit:5 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64 Release
Hit:6 https://developer.download.nvidia.com/compute/machine-learning/repos/ubuntu1804/x86_64 Release
Get:7 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Get:8 http://archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:11 http://archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Fetched 252 kB in 2s (160 kB/s)
Reading package lists... Done
Reading package lists... Done
Building dependency tree
Reading state information... Done
tensorflow-model-server is already the newest version (2.8.0).
0 upgraded, 0 newly installed, 0 to remove and 50 not upgraded.

```

### 4) Thực hiện khởi chạy Tensorflow model server

#### Câu lệnh:

```

!tensorflow_model_server \
--port=8500 \
--rest_api_port=8501 \
--model_name=face-mask-detection-serving \
--model_base_path=/home/dev-fti/thuc/Face-Mask-Detection/
serving_models/mobilenet_v2

```

#### Trong đó:

- **port:** gRPC port, mặc định là cổng 8500
- **rest\_api\_port:** http port (RESTful API), mặc định là cổng 8501
- **model\_name:** tên của model, các bạn đặt thế nào cũng được nhưng sẽ sử dụng để định danh chính xác model cần request

- **model\_base\_path:** đường dẫn tuyệt đối tới thư mục chứa các version của model

Kết quả:

```
2022-03-18 03:50:03.702780: I tensorflow_serving/core/loader_harness.cc:87] Successfully loaded servable version
{name: face-mask-detection-serving version: 1}
2022-03-18 03:50:03.712307: I tensorflow_serving/model_servers/server_core.cc:486] Finished adding/updating models
2022-03-18 03:50:03.712352: I tensorflow_serving/model_servers/server.cc:133] Using InsecureServerCredentials
2022-03-18 03:50:03.712362: I tensorflow_serving/model_servers/server.cc:391] Profiler service is enabled
2022-03-18 03:50:03.713493: I tensorflow_serving/model_servers/server.cc:417] Running gRPC ModelServer at 0.0.0.0:
8500 ...
[warn] getaddrinfo: address family for nodename not supported
2022-03-18 03:50:03.716822: I tensorflow_serving/model_servers/server.cc:438] Exporting HTTP/REST API at:localhost:
8501 ...
[evhttp_server.cc : 245] NET_LOG: Entering the event loop ...
```

Kiểm tra bằng curl:

```
!curl localhost:8501/v1/models/face-mask-detection-serving
```

```
{
  "model_version_status": [
    {
      "version": "1",
      "state": "AVAILABLE",
      "status": {
        "error_code": "OK",
        "error_message": ""
      }
    }
  ]
}
```

## 5) Tiền xử lý

Thực hiện đọc và tiền xử lý ảnh đầu vào.

```
# Apply the same preprocessing as during training (resize and rescale)

image = tf.io.decode_image \
(open('testset/NoMask/frame_2021_12_21_16_10_05_0_1.jpg',
'rb').read(), channels=3)

image = tf.image.resize(image, [224, 224])


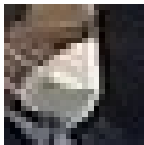
image = image/255.

# Convert the Tensor to a batch of Tensors and then to a list

image_tensor = tf.expand_dims(image, 0)
image_tensor = image_tensor.numpy().tolist()
```

6) Giao tiếp qua RESTful API & gRPC

Kết quả:

	RESTful API	gRPC
Port	8501	8500
URL	<a href="http://localhost:8501/v1/models/face-mask-detection-serving:predict">http://localhost:8501/v1/models/face-mask-detection-serving:predict</a>	<a href="http://localhost:8500/v1/models/face-mask-detection-serving:predict">http://localhost:8500/v1/models/face-mask-detection-serving:predict</a>
	<pre>In [94]: result Out[94]: 'No Mask'</pre>	<pre>In [97]: print(MAP_CHARACTERS[result]) No Mask</pre>
	<pre>In [101]: result Out[101]: 'Mask'</pre>	<pre>In [105]: print(MAP_CHARACTERS[result]) Mask</pre>