My experience and tips for training a custom model using tensorflow 2 object detection api:

I. I took the dataset from Kaggle :
https://www.kaggle.com/ahemateja19bec1025/trafficsignlocalizationdetectionssdannotated
It consists of almost all traffic signs images and associated xml label file. I have attached data_segregation.py file to filter out all not required label data from xml files and also to filter out all not required images (images with no speed limit sign). Now create a label map file with class names and ids. An example pbtxt file (label_map.pbtxt) is also attached in the repo.

II. Below is the link for steps to be followed for setting up environment and installing required libraries:
https://tensorflow-object-detection-api-tutorial.readthedocs.io/en/latest/install.html
And below is the link for for steps to be followed for training custom model:
https://tensorflow-object-detection-api-tutorial.readthedocs.io/en/latest/training.html

III. My system basic details:
OS: windows 10.
Graphics card: Geforce-1080Ti
And 16 GB RAM.

IV. 1. Tensorflow version 2.5 is not suitable and it will lead to unnecessary errors. It is advisable to install version 2.7.
2. Install or upgrade graphics card driver version 460 (recommended) or above.
3. While verifying the installation of tensorflow (gpu test) cuda, cudnn using below command:
python -c "import tensorflow as tf;print(tf.reduce_sum(tf.random.normal([1000, 1000])))"
Incase if you encounter messages like:
successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero.
Don't worry about this you can still go ahead and proceed with next steps.

V. After performing COCO API installation make minor change in the file cocoeval.py at line number 507 and 508.To avoid errors like below:
TypeError: object of type <class 'numpy.float64'> cannot be safely interpreted as an integer.
Lines before changes:
self.iouThrs = np.linspace(.5, 0.95, int(np.round((0.95 - .5) / .05)) + 1, endpoint=True)
self.recThrs = np.linspace(.0, 1.00, int(np.round((1.00 - .0) / .01)) + 1, endpoint=True)
And lines after changes
self.iouThrs = np.linspace(.5, 0.95, int(np.round((0.95 - .5) / .05)) + 1, endpoint=True)
self.recThrs = np.linspace(.0, 1.00, int(np.round((1.00 - .0) / .01)) + 1, endpoint=True)

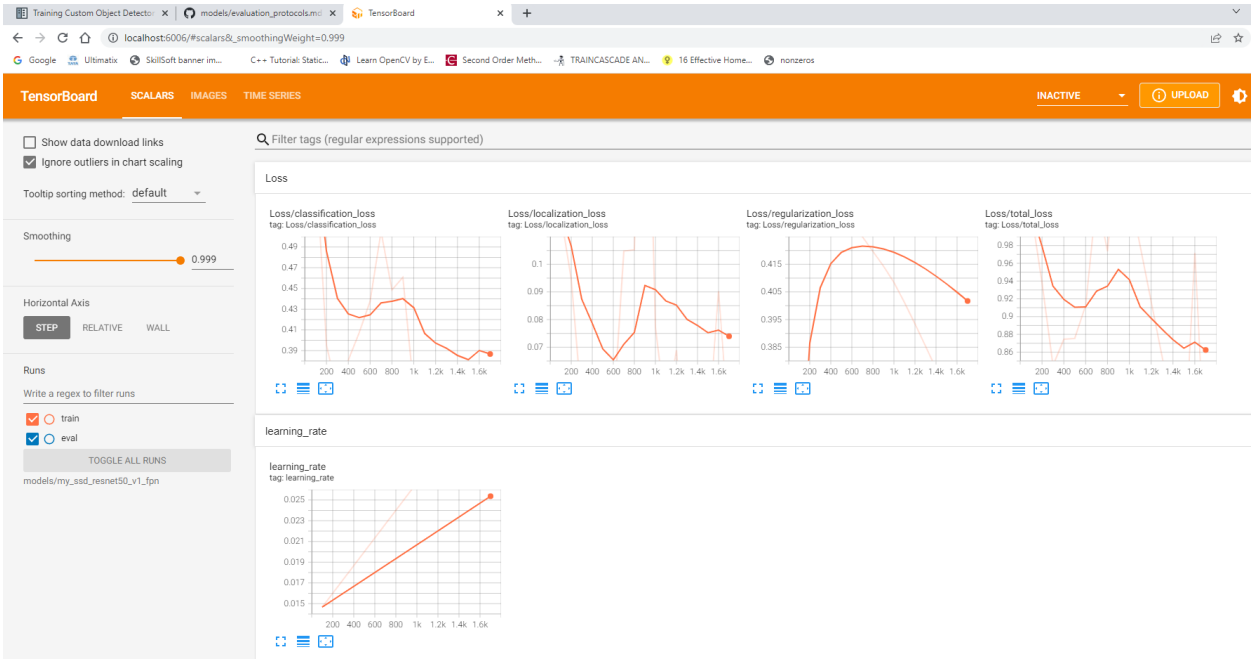
VI. 1. For partitioning data set it is advisable to perform carefully in such a way that all the required category of image should also be included in train, test and validation folder. If dataset is small may be manually splitting is also a good idea.
2. Create 3 TF record files train, test and validation instead of only train and test.

VII. While Configuring the Training Pipeline (pipeline.config file) initially for testing sake set 'total_steps' value to a lower number like 1500 at line number 152 and set 'num_steps' value 1500 at line number 162. The batch size also should be kept a lower number (example 8) if you have limited compute power.

VIII. Training and evaluating the model: if you don't have the validation data (tf record files) then perform training, evaluation and monitoring steps **sequentially** not parallel. It means first finish training phase then complete evaluating model and the use tensor board to monitor the performance of the model.

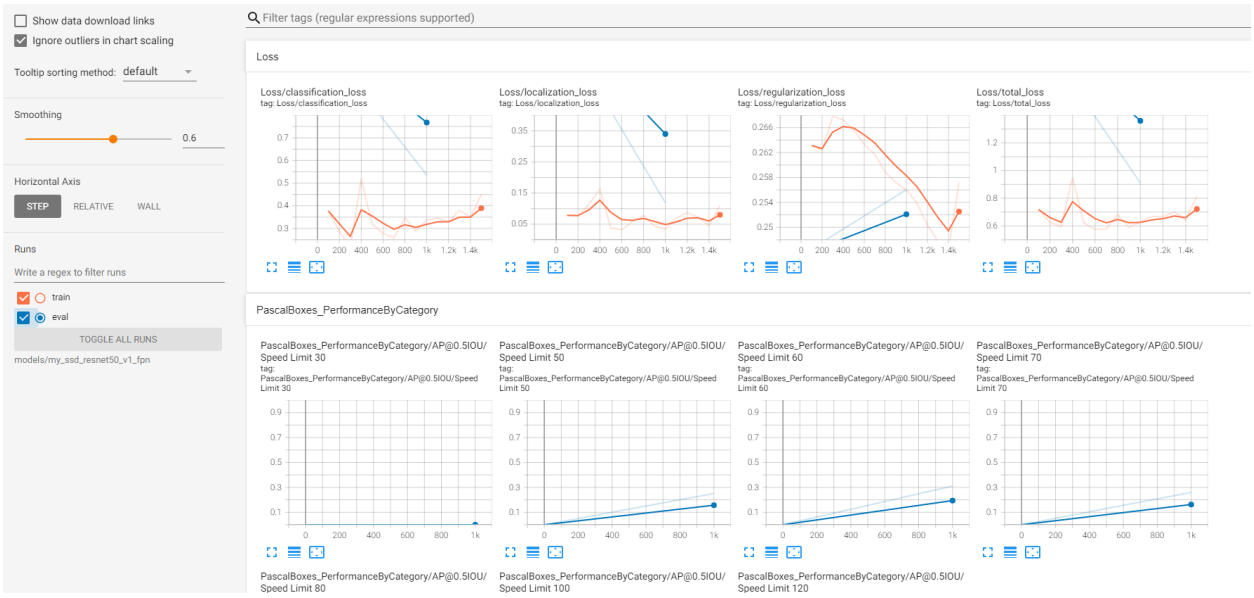
Impact of doing all above steps at a time: In the below figure you can see there is missing evaluation information.
Note: a. I have selected "pascal_voc_detection_metrics" (In pipeline.config file & line number 178).

b. Our intension here is not to train a perfect model but to show the steps required for training a custom model and evaluating it. Hence the performance of model as shown below may not be good.

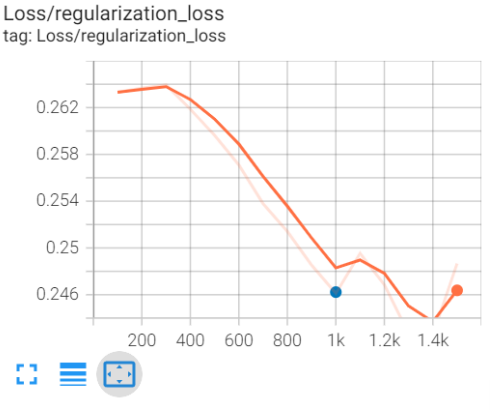


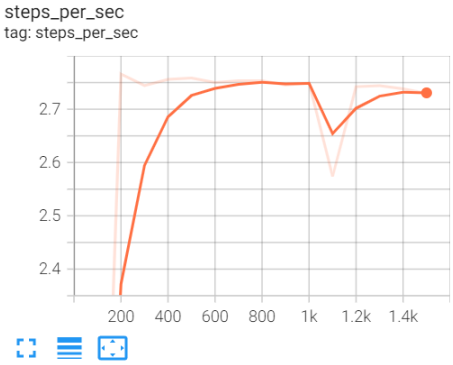
IX. There are multiple options available to evaluate the model. For more details follow the link: https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/evaluation_protocols.md

Result with when training, evaluating and monitoring perform sequentially:



Also using “coco_detection_metrics” result is as follows:





The performance of model can also be reviewed while training is in-progress using validation data set.

X. Results of data inference: