python3 model\_main\_tf2.py --pipeline\_config\_path=./models/EfficientDet/V1/pipeline.config --model\_dir= ./models/EfficientDet/V1 --checkpoint\_every\_n=1 --num\_workers=4 –alsologtostderr

python3 model\_main\_tf2.py --pipeline\_config\_path=./models/EfficientDet/V1/pipeline.config --model\_dir= ./models/EfficientDet/V1 --checkpoint\_every\_n=1 –alsologtostderr

python3 model\_main\_tf2.py --logtostderr --train\_dir== ./models/EfficientDet/V1 --pipeline\_config\_path==./models/EfficientDet/V1/pipeline.config

model {

ssd {

num\_classes: 90

image\_resizer {

keep\_aspect\_ratio\_resizer {

min\_dimension: 768

max\_dimension: 768

pad\_to\_max\_dimension: true

}

}

feature\_extractor {

type: "ssd\_efficientnet-b2\_bifpn\_keras"

conv\_hyperparams {

regularizer {

l2\_regularizer {

weight: 3.9999998989515007e-05

}

}

initializer {

truncated\_normal\_initializer {

mean: 0.0

stddev: 0.029999999329447746

}

}

activation: SWISH

batch\_norm {

decay: 0.9900000095367432

scale: true

epsilon: 0.0010000000474974513

}

force\_use\_bias: true

}

bifpn {

min\_level: 3

max\_level: 7

num\_iterations: 5

num\_filters: 112

}

}

box\_coder {

faster\_rcnn\_box\_coder {

y\_scale: 1.0

x\_scale: 1.0

height\_scale: 1.0

width\_scale: 1.0

}

}

matcher {

argmax\_matcher {

matched\_threshold: 0.5

unmatched\_threshold: 0.5

ignore\_thresholds: false

negatives\_lower\_than\_unmatched: true

force\_match\_for\_each\_row: true

use\_matmul\_gather: true

}

}

similarity\_calculator {

iou\_similarity {

}

}

box\_predictor {

weight\_shared\_convolutional\_box\_predictor {

conv\_hyperparams {

regularizer {

l2\_regularizer {

weight: 3.9999998989515007e-05

}

}

initializer {

random\_normal\_initializer {

mean: 0.0

stddev: 0.009999999776482582

}

}

activation: SWISH

batch\_norm {

decay: 0.9900000095367432

scale: true

epsilon: 0.0010000000474974513

}

force\_use\_bias: true

}

depth: 112

num\_layers\_before\_predictor: 3

kernel\_size: 3

class\_prediction\_bias\_init: -4.599999904632568

use\_depthwise: true

}

}

anchor\_generator {

multiscale\_anchor\_generator {

min\_level: 3

max\_level: 7

anchor\_scale: 4.0

aspect\_ratios: 1.0

aspect\_ratios: 2.0

aspect\_ratios: 0.5

scales\_per\_octave: 3

}

}

post\_processing {

batch\_non\_max\_suppression {

score\_threshold: 9.99999993922529e-09

iou\_threshold: 0.5

max\_detections\_per\_class: 100

max\_total\_detections: 100

}

score\_converter: SIGMOID

}

normalize\_loss\_by\_num\_matches: true

loss {

localization\_loss {

weighted\_smooth\_l1 {

}

}

classification\_loss {

weighted\_sigmoid\_focal {

gamma: 1.5

alpha: 0.25

}

}

classification\_weight: 1.0

localization\_weight: 1.0

}

encode\_background\_as\_zeros: true

normalize\_loc\_loss\_by\_codesize: true

inplace\_batchnorm\_update: true

freeze\_batchnorm: false

add\_background\_class: false

}

}

train\_config {

batch\_size: 8

data\_augmentation\_options {

random\_horizontal\_flip {

}

}

data\_augmentation\_options {

random\_scale\_crop\_and\_pad\_to\_square {

output\_size: 768

scale\_min: 0.10000000149011612

scale\_max: 2.0

}

}

sync\_replicas: true

optimizer {

momentum\_optimizer {

learning\_rate {

cosine\_decay\_learning\_rate {

learning\_rate\_base: 0.07999999821186066

total\_steps: 300000

warmup\_learning\_rate: 0.0010000000474974513

warmup\_steps: 2500

}

}

momentum\_optimizer\_value: 0.8999999761581421

}

use\_moving\_average: false

}

fine\_tune\_checkpoint: "models/workspace/pre\_trained\_models/efficientdet\_d2\_coco17\_tpu-32/checkpoint/ckpt-0"

num\_steps: 300000

startup\_delay\_steps: 0.0

replicas\_to\_aggregate: 8

max\_number\_of\_boxes: 100

unpad\_groundtruth\_tensors: false

fine\_tune\_checkpoint\_type: "detection"

use\_bfloat16: false

fine\_tune\_checkpoint\_version: V2

}

train\_input\_reader: {

label\_map\_path: "models/workspace/data/label\_map.txt"

tf\_record\_input\_reader {

input\_path: "models/workspace/data/train2017-?????-of-00256.tfrecord"

}

}

eval\_config: {

metrics\_set: "coco\_detection\_metrics"

use\_moving\_averages: false

batch\_size: 1;

}

eval\_input\_reader: {

label\_map\_path: "models/workspace/data/label\_map.txt"

shuffle: false

num\_epochs: 1

tf\_record\_input\_reader {

input\_path: "models/workspace/data/val2017-?????-of-00032.tfrecord"

}

}