

Exporting wheat detection model to edge TPU – Documentation

Dataset

Original dataset file = <https://www.kaggle.com/c/global-wheat-detection>

Dataset record file obtained using RoboFlow =

https://drive.google.com/drive/folders/105Qx6m9OK_A-4-RpIYYb3Z31VAmFq8s-?usp=sharing

Training code

```
!pip install -q tensorflow==2.5.0
!pip install -q tflite-model-maker
import numpy as np
import os

from tflite_model_maker.config import ExportFormat
from tflite_model_maker import model_spec
from tflite_model_maker import object_detector

import tensorflow as tf
assert tf.__version__.startswith('2')

tf.get_logger().setLevel('ERROR')
from absl import logging
logging.set_verbosity(logging.ERROR)
from google.colab import drive
drive.mount('/content/drive')
train_data = object_detector.DataLoader('/content/drive/MyDrive/Agrobot/Dataset-record/train/wheat.tfrecord', 2403, {1: 'wheat'})
test_data = object_detector.DataLoader('/content/drive/MyDrive/Agrobot/Dataset-record/test/wheat.tfrecord', 343, {1: 'wheat'})
valid_data = object_detector.DataLoader('/content/drive/MyDrive/Agrobot/Dataset-record/valid/wheat.tfrecord', 686, {1: 'wheat'})
spec = object_detector.EfficientDetLite0Spec(hparams="max_instances_per_image=200")
model = object_detector.create(train_data=train_data,
                              model_spec=spec,
                              validation_data=valid_data,
                              epochs=1,
                              batch_size=32,
```

```

train_whole_model=True)
model.evaluate(test_data)
model.export(export_dir='.',
             tflite_filename='efficientdet-lite-wheat.tflite',
             label_filename='wheat-labels.txt',
             export_format=[ExportFormat.TFLITE, ExportFormat.LABEL])

```

** The above code should output a tflite file. However, this tflite is not yet compatible with the edge tpu. Use https://colab.research.google.com/github/google-coral/tutorials/blob/master/compile_for_edgetpu.ipynb#scrollTo=jcrDngo2WUBW to convert the above tflite file to a edgetpu tflite file that can be used with the coral board.

Running inference

*It is important to update the google coral board's firmware to the whatever latest version is available at the time. issues regarding dependencies will be faced if not done so.

Steps from <https://coral.ai/docs/dev-board/get-started/#flash-the-board> were followed to flash and update the firmware. Continue following the steps until the demo app is able to be run.

The trained wheat detection edge tpu tflite file was moved to a directory in the edge tpu. The below steps were then followed:

```

mkdir coral && cd coral
git clone https://github.com/google-coral/pycoral.git
cd pycoral

```

The pycoral repository contains the scripts and dependencies required to run inference.

```
bash examples/install_requirements.sh classify_image.py
```

The classify_image.py should be replaced with detect_image.py since our model is a detector and not a classifier.

```

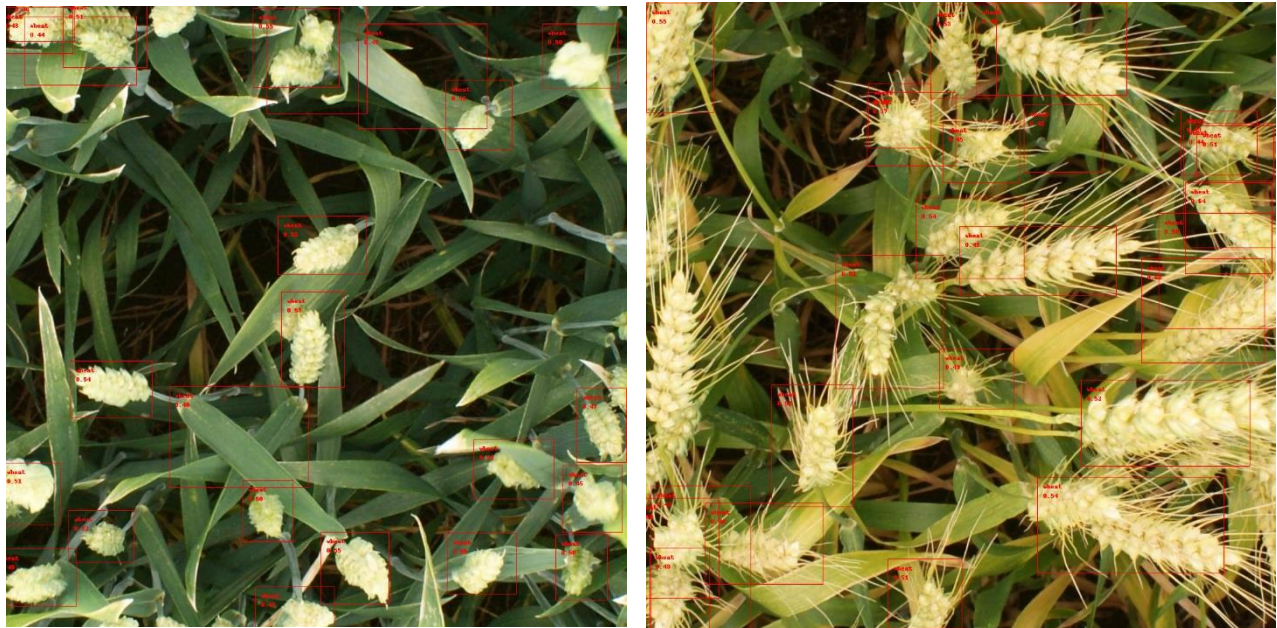
python3 coral/pycoral/examples/detect_image.py
--model wheat_detection_edgetpu/efficientdet-lite-wheat_edgetpu.tflite
--labels wheat_detection_edgetpu/wheat-labels.txt
--input wheat_detection_edgetpu/51b3e36ab.jpg

```

```
--output ${HOME}/wheat_processed.bmp
```

Note that 51b3e36ab.jpg is an image from the test subset of the dataset. This is the input we want to run the inference on. The model is the edge tpu tflite file. The labels wheat-labels.txt is an output of the google colab export performed when training the model. This file should have been exported along with the initial tflite file. The output wheat_processed.bmp is an image file that contains the bounding boxes for the detected wheats.

Examples of outputs (in progress):



Inference times:

```
----INFERENCE TIME----
```

Note: The first inference is slow because it includes loading the model into Edge TPU memory.

```
86.22 ms
```

```
68.66 ms
```

```
69.49 ms
```

```
70.29 ms
```

```
68.85 ms
```

Hence, average inference time is approximately **72ms**. Excluding the initial inference time that includes loading model onto the hardware, average is **69ms**.

Example of running inference (full output):

```
mendel@zippy-orange:~$ python3 coral/pycoral/examples/detect_image.py \
ection_edgetpu/efficientdet-lite-wheat_edgetpu.tflite \
--labels wheat_detection_edgetpu/wheat-labels.txt \
--input wheat_detection_edgetpu/51b3e36ab.jpg \
--output ${HOME}/wheat_processed.bmp> --model wheat_detection_edgetpu/efficientd
et-lite-wheat_edgetpu.tflite \
> --labels wheat_detection_edgetpu/wheat-labels.txt \
> --input wheat_detection_edgetpu/51b3e36ab.jpg \
> --output ${HOME}/wheat_processed.bmp
----INFERENCE TIME----
Note: The first inference is slow because it includes loading the model into Edg
e TPU memory.
86.22 ms
68.66 ms
69.49 ms
70.29 ms
68.85 ms
-----RESULTS-----
wheat
  id:      0
  score:   0.640625
  bbox:    BBox(xmin=876, ymin=293, xmax=1017, ymax=441)
wheat
  id:      0
  score:   0.58203125
  bbox:    BBox(xmin=805, ymin=417, xmax=1005, ymax=586)
wheat
  id:      0
  score:   0.5703125
  bbox:    BBox(xmin=359, ymin=147, xmax=503, ymax=266)
wheat
  id:      0
  score:   0.546875
  bbox:    BBox(xmin=0, ymin=1, xmax=110, ymax=179)
wheat
  id:      0
  score:   0.5390625
  bbox:    BBox(xmin=636, ymin=772, xmax=939, ymax=928)
wheat
  id:      0
  score:   0.5390625
  bbox:    BBox(xmin=438, ymin=318, xmax=614, ymax=449)
wheat
  id:      0
  score:   0.53125
  bbox:    BBox(xmin=462, ymin=5, xmax=569, ymax=156)
```