This document demonstrates how to run the ShapeMask model using Cloud TPU with the COCO dataset.

The instructions below assume you are already familiar with running a model on Cloud TPU. If you are new to Cloud TPU, you can refer to the Quickstart (/tpu/docs/quickstart) for a basic introduction.

If you plan to train on a TPU Pod slice, review Training on TPU Pods (/tpu/docs/training-on-tpu-pods) to understand parameter changes required for Pod slices.

Warning: This tutorial uses a third-party dataset. Google provides no representation, warranty, or other guarantees about the validity, or any other aspects of, this dataset.

- Create a Cloud Storage bucket to hold your dataset and model output
- Prepare the COCO dataset
- Set up a Compute Engine VM and Cloud TPU node for training and evaluation
- Run training and evaluation on a single Cloud TPU or a Cloud TPU Pod

This tutorial uses billable components of Google Cloud, including:

- Compute Engine
- Cloud TPU
- Cloud Storage

Use the pricing calculator (/products/calculator/) to generate a cost estimate based on your projected usage. New Google Cloud users might be eligible for a free trial (/free/).

This section provides information on setting up Cloud Storage storage and a Compute Engine VM.

Important: Set up your Compute Engine VM, your Cloud TPU node and your Cloud Storage bucket in the same region/zone to reduce network latency and network costs.

1. Open a Cloud Shell window.

   Open Cloud Shell (https://console.cloud.google.com/?cloudshell=true)

2. Create a variable for your project's name.

3. Configure `gcloud` command-line tool to use the project where you want to create Cloud TPU.

4. Create a Cloud Storage bucket using the following command:

   ★ **Note:** In the following command, replace ***your-bucket-name*** with the name you want to assign to your bucket.

   This Cloud Storage bucket stores the data you use to train your model and the training results.

5. Launch a Compute Engine VM instance.

6. Connect to the VM instance.

As you continue these instructions, run each command that begins with `(vm)$` in your VM session window.

1. Create a variable to store your Cloud Storage bucket location.

2. Clone the `tpu` repository.

3. Install the packages needed to pre-process the data.

4. Create a directory to store the COCO data and navigate to it.

5. Run the `download_and_preprocess_coco.sh` script to convert the COCO dataset into a set of TFRecords (`*.tfrecord`) that the training application expects.

   This installs the required libraries and then runs the preprocessing script. It outputs a number of `*.tfrecord` files in your local data directory. The COCO download and conversion script takes approximately 1 hour to complete.

6. After you convert the data into TFRecords, copy them from local storage to your Cloud Storage bucket using the `gsutil` command. You must also copy the annotation files. These files help validate the model's performance:

Note: If you want to monitor the model's output and performance, follow the guide to [setting up TensorBoard][tensorboard-setup].

1. Launch a Cloud TPU resource.

2. Set up the following environment variables:

3. Run the training script:

To avoid incurring charges to your Google Cloud Platform account for the resources used in this tutorial:

1. Disconnect from the Compute Engine instance, if you have not already done so:

Your prompt should now be `user@projectname`, showing you are in the Cloud Shell.

2. In your Cloud Shell, run `ctpu delete` with the –zone flag you used when you set up the Cloud TPU to delete your Compute Engine VM and your Cloud TPU:

Important: If you set the TPU resources name when you ran `ctpu up`, you must specify that name with the `--name` flag when you run `ctpu delete` in order to shut down your TPU resources.

3. Run the following command to verify the Compute Engine VM and Cloud TPU have been shut down:

   The deletion might take several minutes. A response like the one below indicates there are no more allocated instances:

Caution: All training data will be lost when you delete your bucket, so only do this step when you are finished running the tutorial.

4. Run `gsutil` as shown, replacing ***your-bucket-name*** with the name of the Cloud Storage bucket you created for this tutorial:

You can explore using a larger backbone network (for example, ResNet-101 instead of ResNet-50). A larger input image and a more powerful backbone will yield a slower but more precise model.

Alternatively, you can explore pre-training a ResNet model on your own dataset and using it as a basis for your ShapeMask model. With some more work, you can also swap in an alternative *backbone* network in place of ResNet. Finally, if you are interested in implementing your own object detection models, this network may be a good basis for further experimentation.