Dataloader

> For now, we only use fake data for training. Let's start to use real data, and build a data loader to load raw data. To speed up data loading process in tensorflow, converting all your data into *tfrecords* are recommended.

First, let's learn to convert image data into tfrecords. In *4.1_tfrecords.py*, we try to convert some images into tfrecords. Then, we will write a data loader to load these files in *4.2_loader.py*

Since the codes are becoming long, we will no longer show them here. But some important codes and explanation will still be listed below.

1. multi-thread

   The codes in *4.1_tfrecords.py* use multi-thread mechanism (coordinator) to generate *tfrecords*.

2. multi-tfrecords

   All image data were divided into *chunks* of data stored in *tfrecord_dir*

3. images SHOULD be stored in *uint8* if the RGB values are ranged between 0 and 255.

Let's have a look at *4.2_loader.py*. It used *tf.dataset* to implement the data loader, which is the recommended way for tensorflow.

1. a parser should be defined to extract the tfrecords data
2. *global_step* is defined to record the current training step
3. a *try except* mechanism is used to train the model for exact *epoch*