

```

1  import os
2  import tensorflow as tf
3  os.environ["CUDA_VISIBLE_DEVICES"] = "1"
4  os.environ['TF_CPP_MIN_LOG_LEVEL'] = '2'
5  import random,math,sys
6  from PIL import Image
7  import numpy as np
8
9  gpu_options =
  tf.compat.v1.GPUOptions(per_process_gpu_memory_fraction=0.333)
10
11  _NUM_TEST = 500
12  _RANDOM_SEED = 0
13  _NUM_SHARDS = 5
14
15  #dataset path
16  DATASET_DIR = r'captcha/images/'
17  #tfrecord where to save
18  TFRECORD_DIR = 'captcha/'
19
20  #判断 tfrecord is exists
21  def _dataset_exists(dataset_dir):
22      for split_name in ['train', 'test']:
23          output_filename = os.path.join(dataset_dir, split_name,
24          '.tfrecords')
25          if not tf.gfile.Exists(output_filename):
26              return False
27          return True
28
29  #get all v_code's path
30  def _get_filename_and_classes(dataset_dir):
31      photo_filenames = []
32      for filename in os.listdir(dataset_dir):
33          path = os.path.join(dataset_dir, filename)
34          photo_filenames.append(path)
35      return photo_filenames
36
37  def int64_feature(values):
38      if not isinstance(values, (tuple, list)):
39          values = [values]
40      return tf.train.Feature(int64_list = tf.train.Int64List(value=values))
41
42  def bytes_feature(values):
43      return tf.train.Feature(bytes_list = tf.train.BytesList(value=
44      [values]))
45
46  def image_to_tfexample(image_data, label0, label1, label2, label3, label4):
47      return tf.train.Example(features=tf.train.Features(feature={
48          'image':bytes_feature(image_data), #bytes类型 int bytes float 可以
49          有三种类型
50          'label0':int64_feature(label0),
51          'label1': int64_feature(label1),
52          'label2': int64_feature(label2),

```

```

50         'label13': int64_feature(label13),
51         'label14': int64_feature(label14),
52     })
53     #为什么要拆成5位呢? 而不是1位呢? 是为了多任务的方式。
54
55     #把数据转换为TFRecord格式
56     def _covert_dataset(split_name, filenames, dataset_dir):
57         assert split_name in ['train', 'test']
58         #计算每个数据块有多少数据(数据量比较大的时候才需要切分)
59         with
tf.compat.v1.Session(config=tf.compat.v1.ConfigProto(gpu_options=gpu_option
s)) as sess:
60             output_filename =
os.path.join(TFRECORD_DIR,split_name+'.tfrecords')
61             with tf.python_io.TFRecordWriter(output_filename) as
tfrecord_writer:
62                 for i, filename in enumerate(filenames):
63                     try:
64                         sys.stdout.write('\r >> Converting image %d/%d %s' %
(i+1, len(filenames), filename))
65                         sys.stdout.flush()
66
67                         #读取图片
68                         image_data = Image.open(filename)
69                         image_data = image_data.resize((224,224)) # 160*60
70                         image_data = np.array(image_data.convert('L')) #灰度化
71                         image_data = image_data.tobytes() #转化为bytes
72
73                         #获取label
74                         labels = filename.split('/')[0:5]
75                         num_labels = []
76                         for j in range(5):
77                             str = labels[j]
78                             if str.isdigit():
79                                 num_labels.append(int(str))
80                             elif str.isalpha():
81                                 num_labels.append(ord(str))
82
83                         #生成protocol数据
84                         example = image_to_tfexample(image_data, num_labels[0],
num_labels[1], num_labels[2], num_labels[3],num_labels[4])
85                         tfrecord_writer.write(example.SerializeToString())
86                     except IOError as err:
87                         print("Could not read:", filenames[i])
88                         print("Error:", err)
89                         print("skip it \n")
90
91             sys.stdout.write('\n')
92             sys.stdout.flush()
93
94
95     if __name__ == '__main__':
96         #判断tfrecord是否存在
97         if _dataset_exists(DATASET_DIR):
98             print('tfrecord is Exists')
99         else:
100             #获得所有图片以及分类
101             photo_filenames = _get_filename_and_classes(DATASET_DIR)

```

```
102     #把分类转换为字典格式，类似于{'house':3, 'flower':1, 'plane':4,  
    'guitar':2, 'animal':0}  
103     class_name_to_ids = dict(zip(class_name, range(len(class_names))))  
104  
105     #把数据切分为训练集和测试集  
106     random.seed(_RANDOM_SEED)  
107     random.shuffle(photo_filenames)  
108     training_filenames = photo_filenames[_NUM_TEST:]  
109     testing_filenames = photo_filenames[:_NUM_TEST]  
110  
111     #数据转换  
112     _covert_dataset('train', training_filenames, DATASET_DIR)  
113     _covert_dataset('test', testing_filenames, DATASET_DIR)  
114  
115     #输出labels文件  
116     labels_to_class_names =  
    dict(zip(range(len(class_names)), class_names))  
117     write_label_file(labels_to_class_names, DATASET_DIR)  
118     print('produce tfrecord sucessful')  
119  
120
```