py_wsi_fork

- **dataset.py**
  - DataSet.class
    - 属性：images,image_cls,labels,num_images,set_id,epochs_completed
    - 2️⃣方法：setter...,shuffle_all,next_batch
  - shuffle_mutiple(list_of_lists)
  - 2️⃣fetch_dataset(turtle,set_id,total_sets,augment)
  - 🔴read_datasets(turtle,set_id=-1,valid_id=-1,total_sets=5,shuffle_all=true,augment=True,is_Test=False)
  - 🔴augment_patches(patches,augment_id)
- **helper.py**
  - 2️⃣start_timer()
  - 2️⃣end_timer(start_time)
- **imagepy_toolkit.py**
  - 🔴def show_images(images, per_row, per_column):
  - 🔴def show_labeled_patches(images, clss):
  - def show_images_and_gt(images, coords, pixel_classes, seg_maps):
  - def show_patch_and_gt(images, seg_maps, pixel_classes, per_row, per_column):
- **item.py**
  - Item.class
    - 方法： def get_label_array(self, num_classes)；
      2️⃣ def get_patch(self):
      def get_patch_as_image(self):
    - 字段：channels,coords,data,label,size
  - SegItem.class
    - 方法：get_seg_map;get_seg_map_as_image;
    - 字段：seg_map,....(Item.class)
- **patch_reader.py**
  - 2️⃣def check_label_exists(label, label_map):
  - 2️⃣def generate_segmentation_patch(seg_map_full, point, patch_size):

- 🔴2️⃣ def generate_label(regions, region_labels, point, label_map):
- 2️⃣ def transform_regions(regions, image_size, total_image_size):
- 2️⃣ def gen_full_segmentation_map(level_dims, regions, region_labels, label_map):
- 🔴2️⃣ def get_regions(path):
- 2️⃣ def patch_to_tile_size(patch_size, overlap):
- def tile_to_patch_size(tile_size, overlap):
- 🔴 def sample_and_store_patches

store.py

- 2️⃣ def save_in_lmdb(env, patches, coords, file_name, labels=[], seg_maps=[]):
- 2️⃣ def save_meta_in_lmdb(meta_env, file, tile_dims):
- 2️⃣ def get_patch_from_lmdb(txn, x, y, file_name):
- 2️⃣ def get_meta_from_lmdb(meta_env, file):
- 2️⃣ def new_lmdb(location, name, map_size_bytes):
- def print_lmdb_keys(env):
- 2️⃣ def read_lmdb(location, name):
- 2️⃣ def save_to_hdf5(db_location, patches, coords, file_name, labels):
- 2️⃣ def save_to_disk(db_location, patches, coords, file_name, labels, seg_maps):

🟢 **turtle.py(Turtle.class)**

方法

- def __init__(self,
- 🔴 def retrieve_tile_dimensions(self, file_name, patch_size=0, overlap=0, tile_size=0):
- 🔴 def retrieve_sample_patch(self, file_name, patch_size, level, overlap=0):
- 🔴 def get_set_patches(self, set_id, total_sets, select=[]):
- 🔴2️⃣ def get_patches_from_file(self, file_name, verbose=False):
- 🔴 def sample_and_store_patches(self,
- def set_label_map(self, label_map):
- def set_file_dir(self, file_dir):
- def set_xml_dir(self, xml_dir):
- 🔴2️⃣ def get_xml_files(self):

- def set_db_location(self, db_location):
- def set_db_name(self, db_name):
- def __get_files_from_dir(self, file_dir, file_type='.svs'):
- def __get_db_meta_name(self, db_name):
- 2 def __check_file_found(self, file_name):
- 2 def __get_patches_from_hdf5(self, file_name, verbose=False):
- 2 def __sample_store_hdf5(self, patch_size, level, overlap, xml_dir, limit_bounds, rows_per_txn):
- 2 def __get_patches_from_disk(self, wsi_name, verbose=False):
- 2 def __sample_store_disk(self, patch_size, level, overlap, xml_dir,
- 2 def __get_items_from_file(self, file_name):
- 2 def __items_to_patches_and_meta(self, items):
- def __calculate_map_size(self, patch_size, level, overlap, limit_bounds,
- 2 def __sample_store_lmdb(self, patch_size, level, overlap, xml_dir,

字段

- 🔴 db_location
- db_meta_name
- 🔴 db_name
- 🔴 file_dir
- 🔴 files
- 🔴 label_map
- 🔴 num_files
- 🔴 storage_type
- 🔴 xml_dir