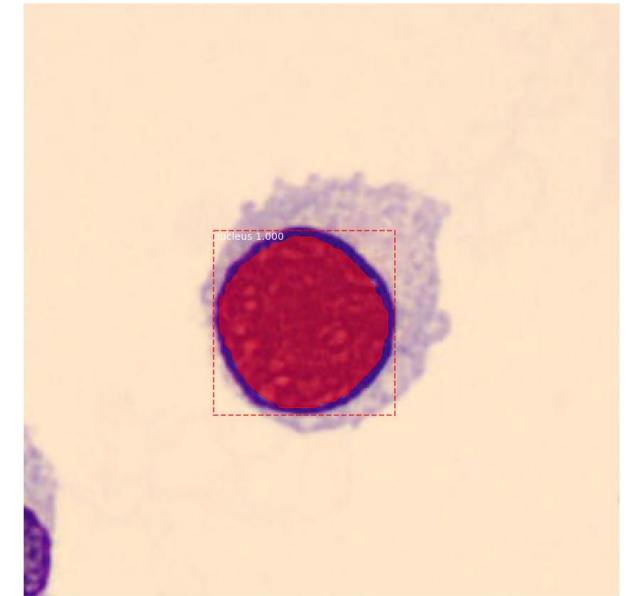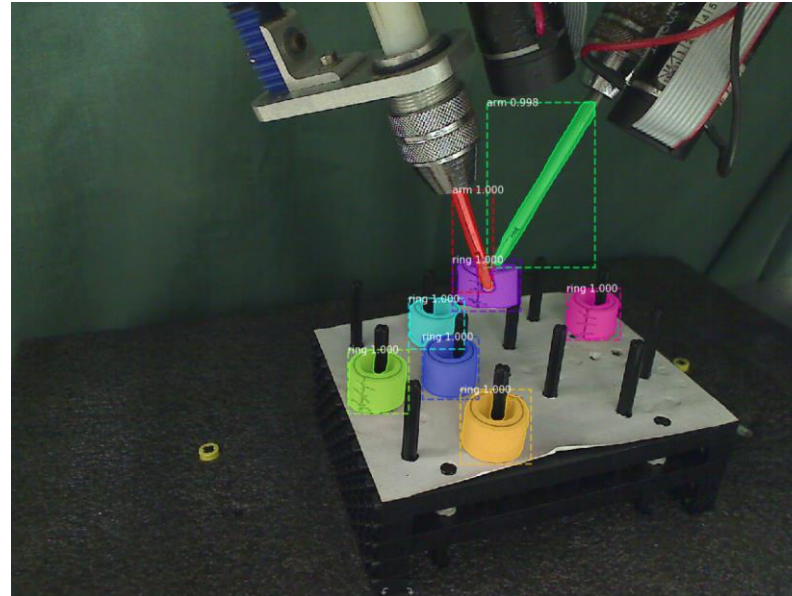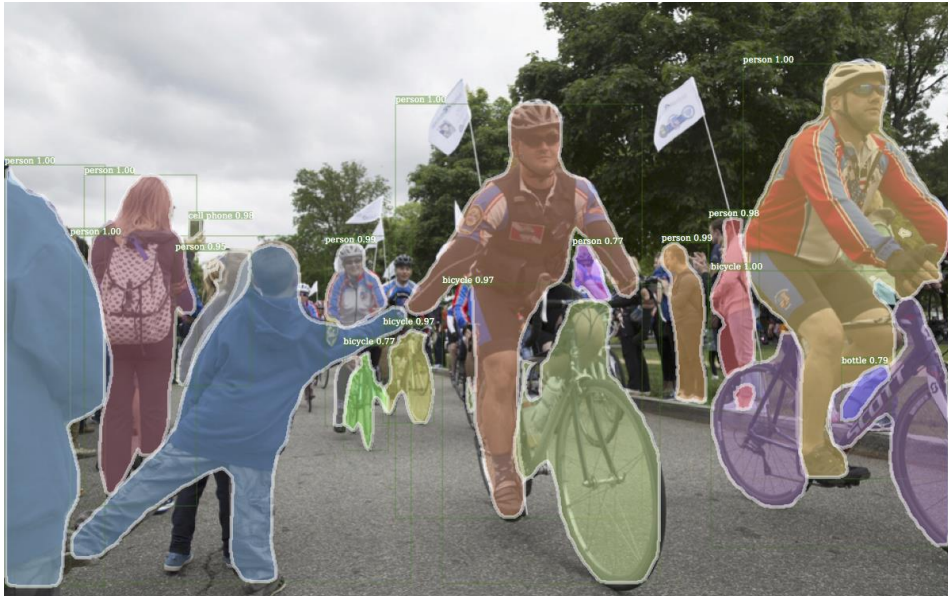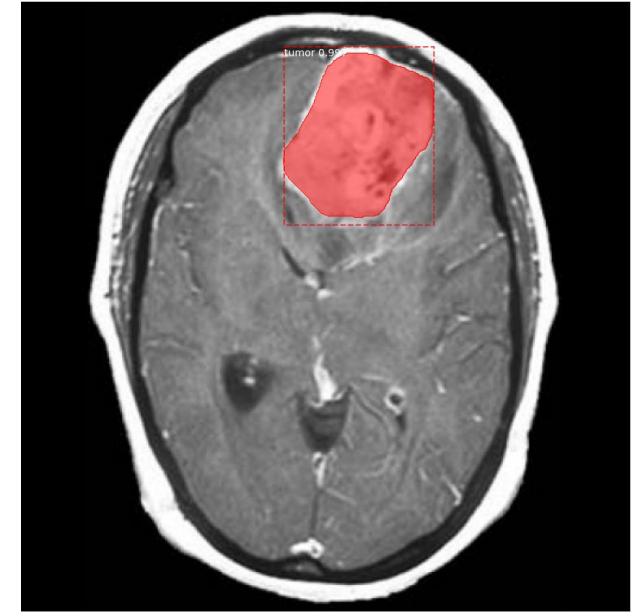# Image-Segmentation-MaskRCNN-Keras-Jupyter

Inheriting the algorithm MaskRCNN of FasterRCNN, it improves the original architecture and adds the part of Semantic Segmentation, which can segment, detect and classify objects. It is a classic Segmentation algorithm.
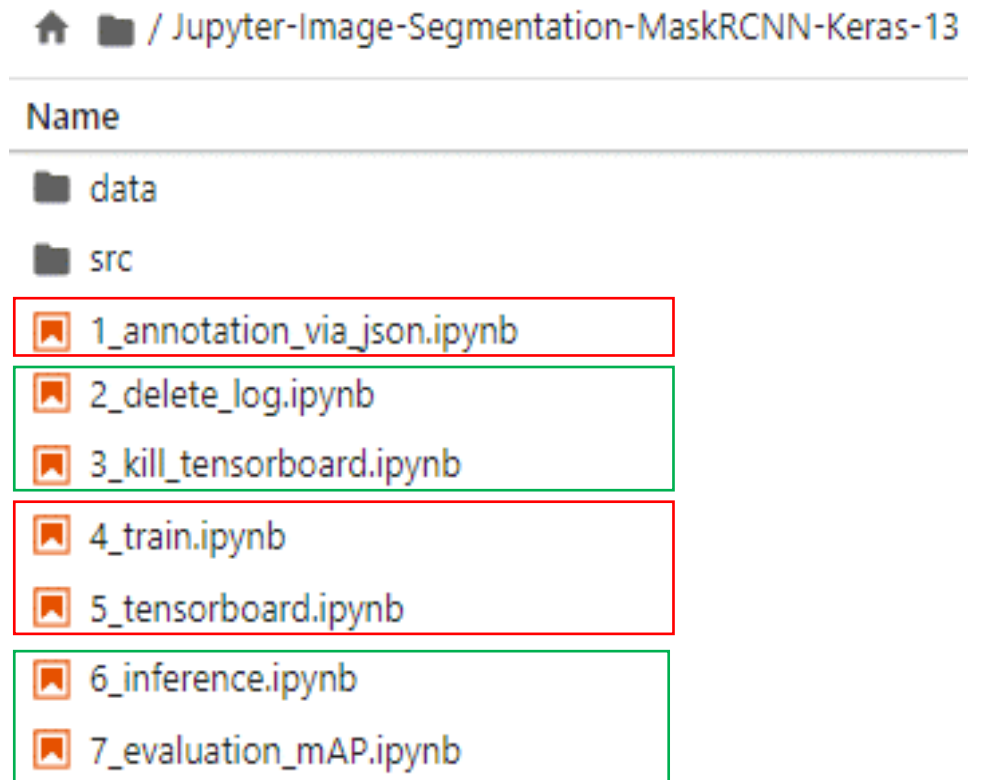
Version 20230223

Applications

- The Mask R-CNN algorithm can be applied to medical image analysis, fish image analysis, biological image analysis, advanced driver assistance systems, autonomous vehicles, factory defect detection, mask detection, satellite image segmentation, etc.

How to use

The main process is:

Annotate images -> Prepare files for training -> Training -> Inference

1_annotation_via_json.ipynb

Open the webpage for image annotation.

ipynb parameter:

- "port" is the port used by the webpage. If the port is occupied by the user, please change another port value by yourself.

- "image_folder" is the image path.

- "annotation_path" is the path to the annotation archive.

See Annotation.pdf for how to use annotation pages.
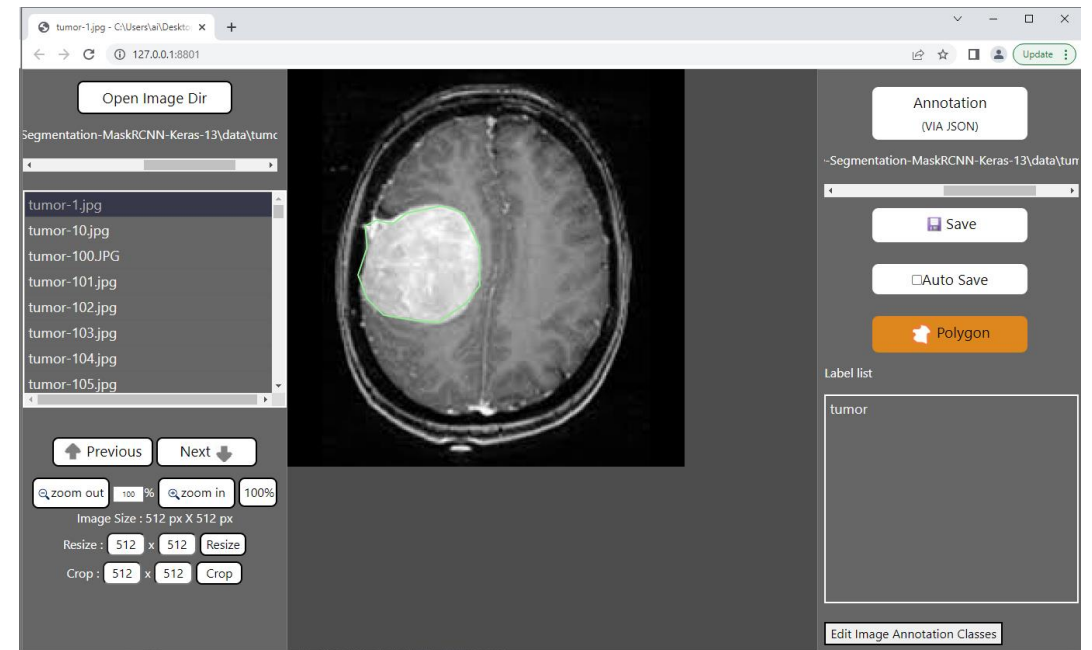
2_delete_log.ipynb

Delete the log files left over from previous training.

3_kill_tensorboard.ipynb

Close a TensorBoard that was not closed before.

4_train.ipynb

Start training.

ipynb parameter description:

- --dataset is the training dataset path.

- --weights_path is the pretrained model model path.

- --weights_dir is the path to store the model generated by training

- --image is the path of the inferred image, which can be a single image path or an image folder path.

- --predefined_classes_txt is a txt file of class names used during training.

- --logs is the path to store the log files generated by training.

```
100/100 [==============================] - ETA: 0s - batch: 49.5000 - size: 1.0000 - loss: 1.6889 - rpn_class_loss: 0.0305 - rpn_bbox_loss: 0.
7189 - mrcnn_class_loss: 0.0621 - mrcnn_bbox_loss: 0.4348 - mrcnn_mask_loss: 0.4426
C:\Users\ai\Desktop\App4AI-2225\gpu\python\lib\site-packages\keras\engine\training_v1.py:2045: UserWarning: `Model.state_updates` will be remo
ved in a future version. This property should not be used in TensorFlow 2.0, as `updates` are applied automatically.
  updates = self.state_updates
100/100 [==============================] - 142s 1s/step - batch: 49.5000 - size: 1.0000 - loss: 1.6889 - rpn_class_loss: 0.0305 - rpn_bbox_los
s: 0.7189 - mrcnn_class_loss: 0.0621 - mrcnn_bbox_loss: 0.4348 - mrcnn_mask_loss: 0.4426 - val_loss: 1.0373 - val_rpn_class_loss: 0.0194 - val
_rpn_bbox_loss: 0.2847 - val_mrcnn_class_loss: 0.0729 - val_mrcnn_bbox_loss: 0.3683 - val_mrcnn_mask_loss: 0.2921
Epoch 2/60
 45/100 [============>.................] - ETA: 44s - batch: 22.0000 - size: 1.0000 - loss: 0.9220 - rpn_class_loss: 0.0200 - rpn_bbox_loss:
0.2563 - mrcnn_class_loss: 0.0486 - mrcnn_bbox_loss: 0.2815 - mrcnn_mask_loss: 0.3157
```
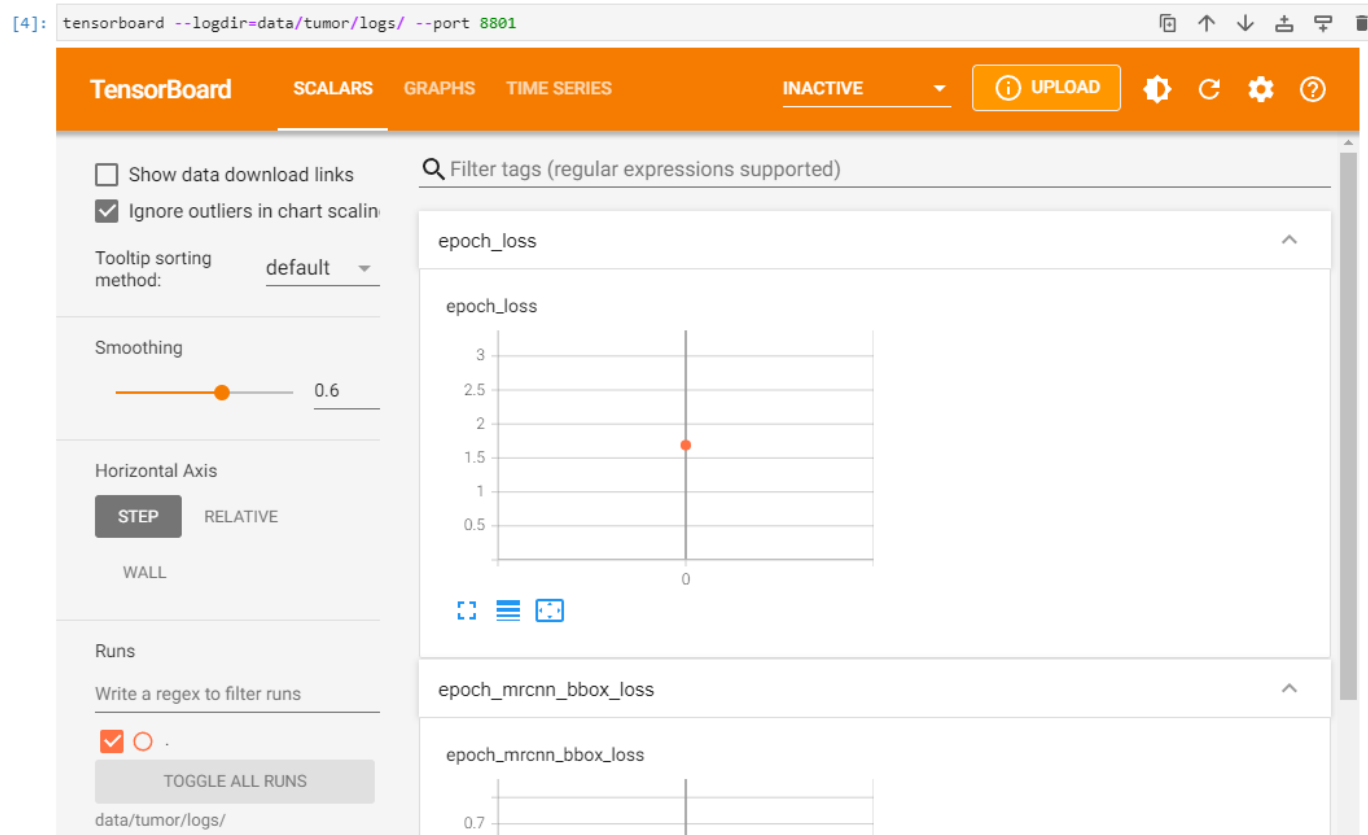
5_tensorboard.ipynb

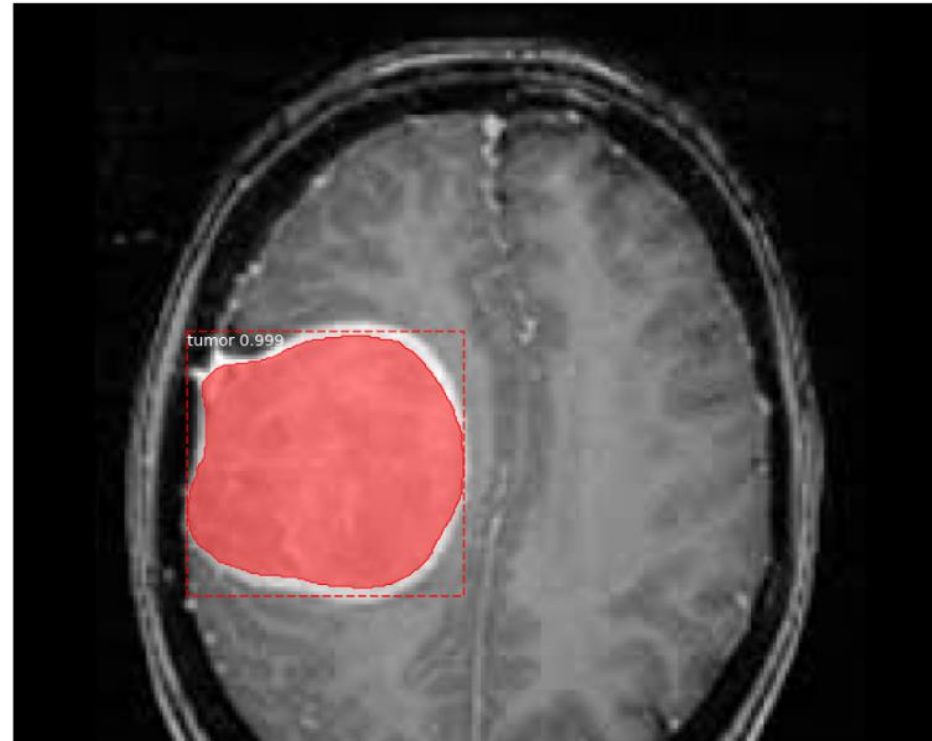You can view the training loss curve and other related information through TensorBoard.

6_inference.ipynb

Infer a single image or all images in a folder.

ipynb parameter description:

- --weights is the inference model path.

- --image is the path of the inferred image, which can be a single image path or an image folder path.

- --predefined_classes_txt is a txt file of class names used during training.

7_evaluation_mAP.ipynb

Calculate mAP for each class in the image folder.

ipynb parameter description:
- --weights is the model path used to compute mAP.
- --DATA_DIR is the dataset path.
- --PREDEFINED_CLASSES_PATH is the class name txt file to use when training.
- --Inference_DATA_DIR is the specified folder in the dataset for mAP calculation, such as train, val.

```
LEARNING_RATE                    0.001
LOSS_WEIGHTS                     {'rpn_class_loss': 1.0, 'rpn_bbox_loss': 1.0, 'mrcnn_class_loss': 1.0, 'mrcnn_bbox_loss': 1.0, 'mrcnn_mask_loss': 1.0}
MASK_POOL_SIZE                   14
MASK_SHAPE                       [28, 28]
MAX_GT_INSTANCES                 100
MEAN_PIXEL                       [123.7 116.8 103.9]
MINI_MASK_SHAPE                  (56, 56)
NAME                             mask
NUM_CLASSES                      2
POOL_SIZE                        7
POST_NMS_ROIS_INFERENCE          1000
POST_NMS_ROIS_TRAINING           2000
ROI_POSITIVE_RATIO               0.33
RPN_ANCHOR_RATIOS                [0.5, 1, 2]
RPN_ANCHOR_SCALES                (32, 64, 128, 256, 512)
RPN_ANCHOR_STRIDE                1
RPN_BBOX_STD_DEV                 [0.1 0.1 0.2 0.2]
RPN_NMS_THRESHOLD                0.7
RPN_TRAIN_ANCHORS_PER_IMAGE      256
STEPS_PER_EPOCH                  100
TRAIN_BN                         False
TRAIN_ROIS_PER_IMAGE             200
USE_MINI_MASK                    False
USE_RPN_ROIS                     True
VALIDATION_STEPS                 30
WEIGHT_DECAY                     0.01


28
WARNING:tensorflow:From C:\Users\ai\Desktop\App4AI-2225\gpu\python\lib\site-packages\tensorflow\python\util\deprecation.py:629: calling map_fn_v2 (from tensorflow.python.
ops.map_fn) with dtype is deprecated and will be removed in a future version.
Instructions for updating:
Use fn_output_signature instead
Loading weights  data/tumor/model/mask_rcnn_mask.h5
C:\Users\ai\Desktop\App4AI-2225\gpu\python\lib\site-packages\keras\engine\training_v1.py:2067: UserWarning: `Model.state_updates` will be removed in a future version. Thi
s property should not be used in TensorFlow 2.0, as `updates` are applied automatically.
  updates=self.state_updates,
85.36% = tumor AP
mAP = 85.36%
```

Reference

- Please refer to the readme.txt in the SDK folder.
- LEADERG AppForAI: https://www.leaderg.com/appforai-windows