Introduction of Quick Demo for SDM-RAN

- What does this document describe?

This document describes the configurations and operation steps of the proposed SDM-RAN demo program on COCO2017 dataset [1] (Few-Shot Object Detection At Once: FSOD-AO) task and FSC-147 dataset [2] (counting task). The codes are uploaded at the address https://github.com/Brronnie/SDM-RAN. The environments with Python 3.7~3.8, Pytorth 1.10~1.13, numpy 1.23, opencv 4.6.0 configuration is needed to run the codes.

- The total structure of the codes

Fig. A presents the structure of the SDM-RAN demo programs.

SDM-RAN contains two subdirectories: FSOD-AO and VCount. The codes, models and related images (from COCO dataset) for FSOD-AO task are contained in FSOD-AO file folder. And those for visual counting task (data from FSC-147 dataset) are contained in VCount file folder.

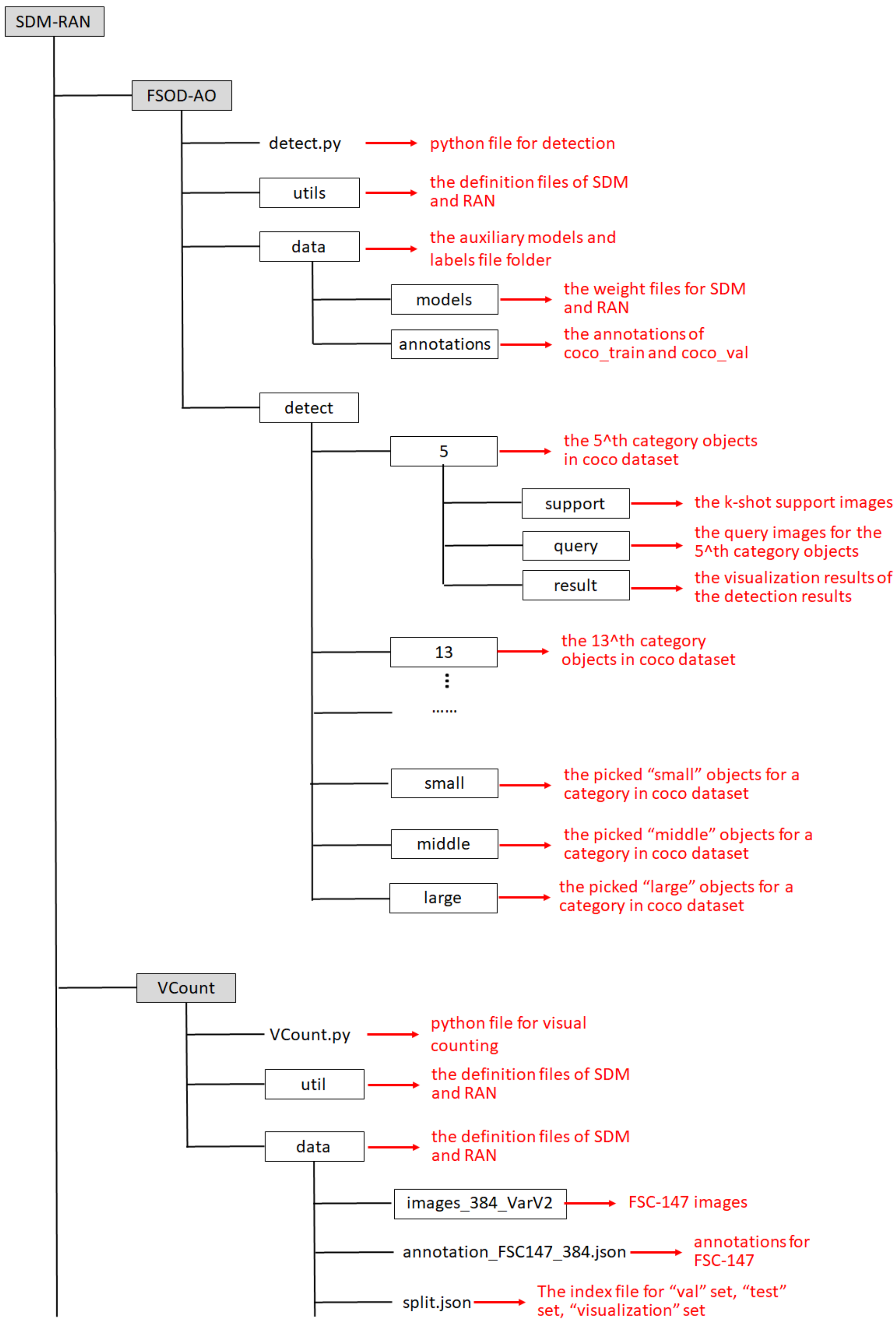

Fig. A: The structure of the SDM-RAN demo programs.

- Quick demo for FSOD-AO

(1) To run the code for FSOD-AO task, please first download all files from https://github.com/Brronnie/SDM-RAN/FSOD-AO.

(2) Download the file “Siamese.pt” from the link “https://drive.google.com/file/d/19Ga696qZKdwLGhIBb2EdoK2FkKovtCF8/view?usp=share_link”, copy the file “Siamese.pt” to “SDM-RAN/FSOD-AO/data/models/Siamese.pt”.

(3) At a terminal window, change the work directory to the file path “FSOD-AO” using “cd FSOD-AO”.

(4) Input command: “**python detect.py --category 13**”, where “detect.py” is python file for FSOD-AO detection, “--category 13” indicates the category number of the object annotated in COCO dataset, for example: 13 is the object index for “stop sign”.

(5) The code will detect all the images in “FSOD-AO\detect\13\” file folder. Meanwhile, the detection visualization results will be presented simultaneously with each image being processing. Fig. B ~ Fig. F present several visualization samples. In these images, the left image of top row is the 1-shot support image, the middle image of top row is a query image, the right image of top row is Similarity Density Map (SDM), the left image of bottom row is the overlaid density image and query image, the middle image of bottom row present the top possible regions after purification and NMS process, the right image of bottom row is the FSOD-AO results obtained by our method.

(6) After all the images are processed for current category object, the values for AP, AP50 and AP75 are listed in the output window. Fig. G presents the AP, AP50 and AP75 values outputted by “detection.py” for category 13 using 1-shot.

(7) Input command: “**python detect.py --category middle**”. The program will detect the middle objects for a given category. In the demonstration version, the middle objects of category 13 “stop sign” are given in the “middle” file folder. The users can replace the support and query images for other categories in “small”, “middle” and “large” file folders.

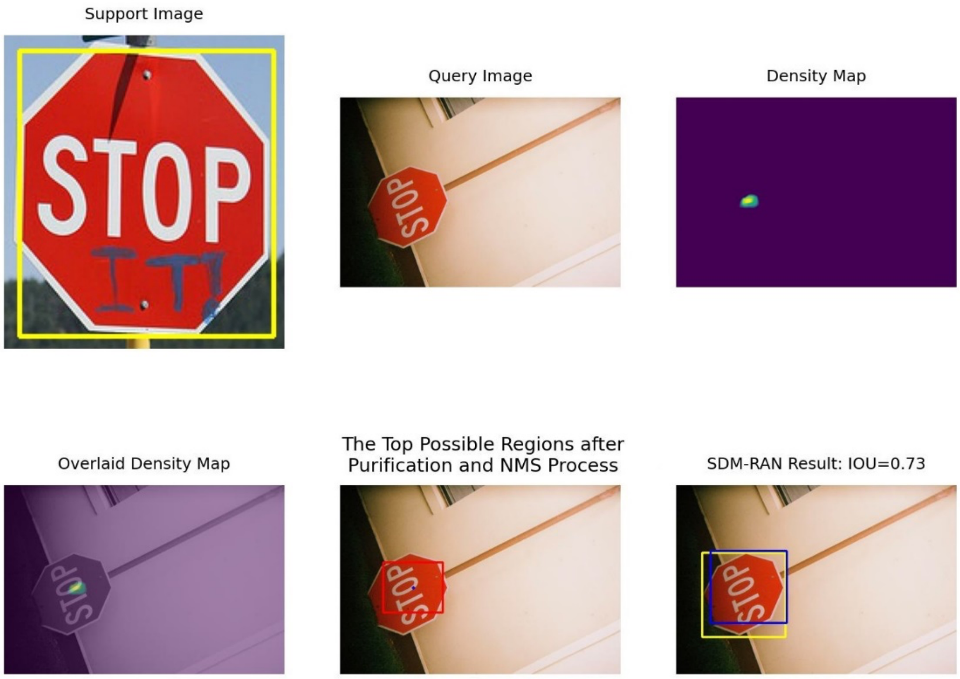

Fig. B: The 1-shot detection visualization results for a image of category 13 “stop sign”. The left image of top row is the 1-shot support image. The middle image of top row is a query image, the right image of top row is Similarity Density Map (SDM). The left image of bottom row is the overlaid density image and query image. The middle image of bottom row present the top possible regions after purification and NMS process. The right image of bottom row is the FSOD-AO results obtained by our method, where the boxes in blue are the detection results of our method and the boxes in yellow are ground truth.

Fig. C: The 1-shot detection visualization results for an image of category 13 “stop sign”.

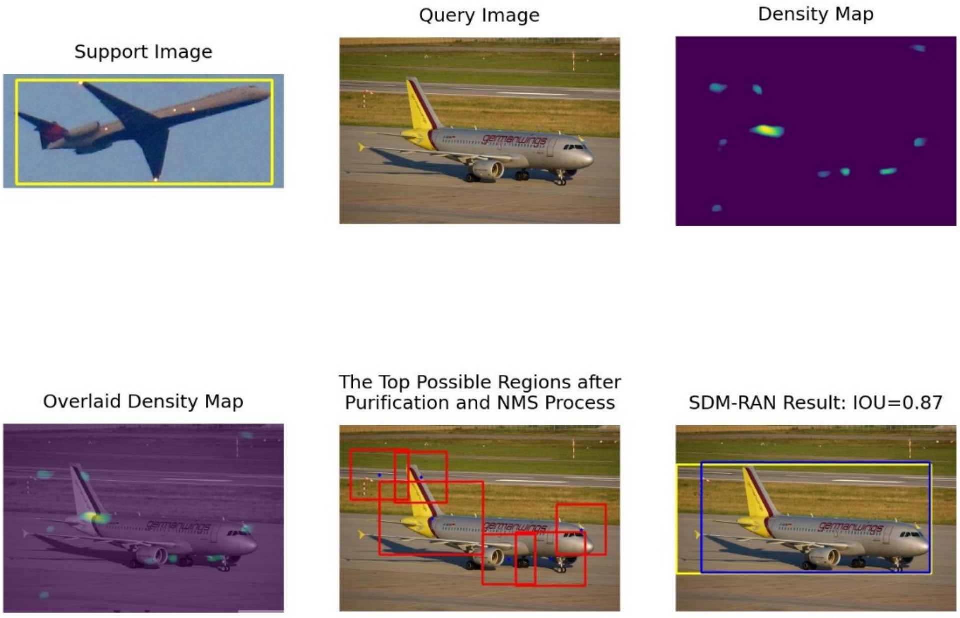

Fig. D: The 1-shot detection visualization results for an image of category 5 “plane”.

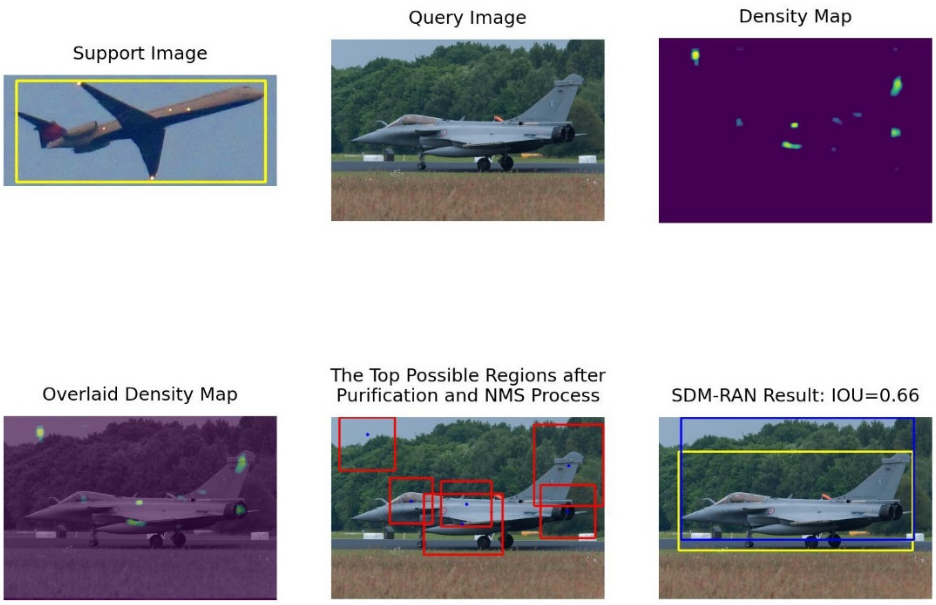

Fig. F: The 1-shot detection visualization results for an image of category 5 “plane”.

```
The Average Precision (AP) Result for Category 5
AP: 0.1122
AP50: 0.2653
AP75: 0.1224
```

Fig. G: The 1-shot values of AP, AP50 and AP75 outputted by “detection.py” for random 50 images of category 5 “plane”.

- Quick demo for VCount

To run the code for visual counting task, please first download all files from https://github.com/Brronnie/SDM-RAN/VCount.

(2) Download the file “RAN.pt” from the link

“https://drive.google.com/file/d/145s_OeErMBGoQBs9NOMnwNCSn8pGp6gM/view?usp=share_link”, copy the file “RAN.pt” to the “SDM-RAN/VCount/util/RAN.pt”.

(3) At a terminal window, change the work directory to the file path “VCount” using “cd VCount”.

(4) Input command: “**python VCount.py --split test --shot 1**”, where “VCount.py” is python file for visual counting, “--split test” indicates the “test” set of the FSC-147 is used for counting validation, and “--shot 1” indicates that 1-shot is carried out on the assigned set.

(5) The code will count all the images in “test” set of FSC-147. Meanwhile, the counting results for FamNet [2] and our are presented simultaneously with each image being processing. Fig. H present three samples of counting results for several visualization images.

(6) After all the images are processed for current set, the MAE and RMSE values for FamNet [2] and ours are listed in the output window. Fig. I presents the MAE and RMSE values obtained by FamNet [2] and ours for “Val” set, where the smaller the MAE and RMSE values are better.

(7) Input command: “**python VCount.py --split visualization --shot 3**”, where “VCount.py” is python file for visual counting, “--split visualization” indicates the “visualization” set of the FSC-147 is used for counting validation, and the visualization results will be given in the “visualization_results”. Fig. J presents several counting results obtained by our method using 1-shot on FSC-147 dataset. The images in the top row are the original images, where the boxes in white are the support images appointed by users. The images in the bottom row are the counting results, where each red dot indicates an object.

```
img_id: 817.jpg
GT value: 34
Baseline(FamNet): 47
Ours: 33
------------------------
img_id: 827.jpg
GT value: 40
Baseline(FamNet): 35
Ours: 34
------------------------
img_id: 988.jpg
GT value: 12
Baseline(FamNet): 11
Ours: 11
```

The counting results of baseline method [2] and ours for visualization image 715.jpg

The counting results of baseline method [2] and ours for visualization image 827.jpg

The counting results of baseline method [2] and ours for visualization image 988.jpg

Fig. H: The samples of counting results for image 817.jpg, 827.jpg and 988.jpg of FSC-147.

```
Baseline(FamNet) on val data, MAE:  24.21, RMSE:  71.20
Our method on val data, MAE:  21.39, RMSE:  70.56
```

Fig. I: The MAE and RMSE values obtained by FamNet [2] and our method for “Val” set of FSC-147.

Fig. J: Several counting results obtained by our method of 1-shot on “visualization” set of FSC-147

References:

[1] H. F. Xinlei Chen, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollar, C. Lawrence Zitnick, "Microsoft COCO Captions: Data Collection and Evaluation Server," 2015.

[2] U. S. Viresh Ranjan, Thu Nguyen, Minh Hoai, "Learning To Count Everything," in *In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR 2021)*, Nashville, TN, USA, 2021.