

A very dear mind

0.1.7

다음에 의해 생성됨 : Doxygen 1.8.16

1 Faster_RCNN (base network: VGG16)	1
1.1 작성정보	1
2 Extra	3
2.1 ground-truth:	3
2.2 detection-results:	3
2.3 Find the files that contain a specific class of objects	4
2.4 Intersect ground-truth and detection-results files	4
3 네임스페이스 색인	5
3.1 패키지	5
4 계통도 색인	7
4.1 클래스 계통도	7
5 클래스 색인	9
5.1 클래스 목록	9
6 파일 색인	11
6.1 파일 목록	11
7 네임스페이스 문서화	13
7.1 convert_dr_darkflow_json 네임스페이스 참조	13
7.1.1 변수 문서화	13
7.1.1.1 bottom	13
7.1.1.2 conf	13
7.1.1.3 data	14
7.1.1.4 DR_PATH	14
7.1.1.5 json_list	14
7.1.1.6 left	14
7.1.1.7 obj_name	14
7.1.1.8 parent_path	14
7.1.1.9 right	15
7.1.1.10 top	15
7.2 convert_dr_yolo 네임스페이스 참조	15
7.2.1 변수 문서화	15
7.2.1.1 bbox	15
7.2.1.2 bottom	16
7.2.1.3 class_name	16
7.2.1.4 confidence	16
7.2.1.5 DR_PATH	16
7.2.1.6 height	16
7.2.1.7 image_name	16
7.2.1.8 image_path	17

7.2.1.9 IMG_FORMAT	17
7.2.1.10 IN_FILE	17
7.2.1.11 info	17
7.2.1.12 left	17
7.2.1.13 outfile	17
7.2.1.14 parent_path	18
7.2.1.15 right	18
7.2.1.16 SEPARATOR_KEY	18
7.2.1.17 top	18
7.2.1.18 width	18
7.3 convert_gt_xml 네임스페이스 참조	18
7.3.1 변수 문서화	19
7.3.1.1 bndbox	19
7.3.1.2 bottom	19
7.3.1.3 GT_PATH	19
7.3.1.4 left	19
7.3.1.5 obj_name	19
7.3.1.6 parent_path	20
7.3.1.7 right	20
7.3.1.8 root	20
7.3.1.9 top	20
7.3.1.10 xml_list	20
7.4 convert_gt_yolo 네임스페이스 참조	20
7.4.1 함수 문서화	21
7.4.1.1 convert_yolo_coordinates_to_voc()	21
7.4.2 변수 문서화	22
7.4.2.1 bottom	22
7.4.2.2 content	22
7.4.2.3 GT_PATH	22
7.4.2.4 height_n	22
7.4.2.5 image_name	23
7.4.2.6 img	23
7.4.2.7 img_height	23
7.4.2.8 img_width	23
7.4.2.9 left	23
7.4.2.10 obj_id	24
7.4.2.11 obj_list	24
7.4.2.12 obj_name	24
7.4.2.13 parent_path	24
7.4.2.14 right	24
7.4.2.15 top	25
7.4.2.16 txt_list	25

7.4.2.17 width_n	25
7.4.2.18 x_c_n	25
7.4.2.19 y_c_n	25
7.5 convert_keras 네임스페이스 참조	25
7.5.1 상세한 설명	25
7.6 convert_keras-yolo3 네임스페이스 참조	26
7.6.1 변수 문서화	26
7.6.1.1 action	26
7.6.1.2 annot	26
7.6.1.3 annotation_dir_name	27
7.6.1.4 annotation_file	27
7.6.1.5 annotation_version	27
7.6.1.6 ap	27
7.6.1.7 ARGS	27
7.6.1.8 class_id	27
7.6.1.9 class_map	28
7.6.1.10 default	28
7.6.1.11 destination_dir	28
7.6.1.12 exist_ok	28
7.6.1.13 file_name	28
7.6.1.14 group	28
7.6.1.15 help	29
7.6.1.16 img_path	29
7.6.1.17 out_box	29
7.6.1.18 output_file_path	29
7.6.1.19 output_path	29
7.6.1.20 required	29
7.6.1.21 score	30
7.6.1.22 type	30
7.6.1.23 x_max	30
7.6.1.24 x_min	30
7.6.1.25 y_max	30
7.6.1.26 y_min	30
7.7 data 네임스페이스 참조	31
7.8 data.dataset 네임스페이스 참조	31
7.8.1 함수 문서화	31
7.8.1.1 AddGaussianNoise()	31
7.8.1.2 blur()	32
7.8.1.3 caffe_normalize()	32
7.8.1.4 ChangeLuminance()	33
7.8.1.5 inverse_normalize()	33
7.8.1.6 preprocess()	33

7.8.1.7 pytorch_normalize()	35
7.9 data.util 네임스페이스 참조	36
7.9.1 함수 문서화	36
7.9.1.1 _slice_to_bounds()	36
7.9.1.2 crop_bbox()	37
7.9.1.3 flip_bbox()	38
7.9.1.4 random_flip()	39
7.9.1.5 read_image()	41
7.9.1.6 resize_bbox()	42
7.9.1.7 translate_bbox()	43
7.10 data.voc_dataset 네임스페이스 참조	44
7.10.1 변수 문서화	44
7.10.1.1 VOC_BBOX_LABEL_NAMES	44
7.11 demo 네임스페이스 참조	44
7.11.1 변수 문서화	45
7.11.1.1 _bboxes	45
7.11.1.2 _labels	45
7.11.1.3 _scores	45
7.11.1.4 anno	45
7.11.1.5 bbox	45
7.11.1.6 bndbox_anno	46
7.11.1.7 caffe_pretrain	46
7.11.1.8 difficult	46
7.11.1.9 faster_rcnn	46
7.11.1.10 img	46
7.11.1.11 label	46
7.11.1.12 name	47
7.11.1.13 ori_img_	47
7.11.1.14 output_img	47
7.11.1.15 trainer	47
7.11.1.16 visualize	47
7.11.1.17 VOC_BBOX_LABEL_NAMES	47
7.12 Evaluation 네임스페이스 참조	48
7.12.1 함수 문서화	48
7.12.1.1 eval()	48
7.12.2 변수 문서화	49
7.12.2.1 batch_size	49
7.12.2.2 eval_result	49
7.12.2.3 faster_rcnn	49
7.12.2.4 num_workers	49
7.12.2.5 pin_memory	49
7.12.2.6 shuffle	50

7.12.2.7 test_dataloader	50
7.12.2.8 testset	50
7.13 find_class 네임스페이스 참조	50
7.13.1 함수 문서화	50
7.13.1.1 find_class()	51
7.13.2 변수 문서화	51
7.13.2.1 DR_PATH	51
7.13.2.2 GT_PATH	51
7.13.2.3 parent_path	51
7.13.2.4 searching_class_name	52
7.14 img_crawling 네임스페이스 참조	52
7.14.1 함수 문서화	52
7.14.1.1 imageCrawling()	52
7.14.1.2 img_check()	53
7.14.1.3 img_delete()	54
7.14.2 변수 문서화	54
7.14.2.1 _create_default_https_context	54
7.15 intersect-gt-and-dr 네임스페이스 참조	54
7.15.1 함수 문서화	55
7.15.1.1 backup()	55
7.15.2 변수 문서화	55
7.15.2.1 backup_folder	55
7.15.2.2 dr_backup	55
7.15.2.3 dr_files	55
7.15.2.4 DR_PATH	56
7.15.2.5 gt_backup	56
7.15.2.6 gt_files	56
7.15.2.7 GT_PATH	56
7.15.2.8 intersection	56
7.15.2.9 parent_path	56
7.16 main 네임스페이스 참조	57
7.16.1 함수 문서화	59
7.16.1.1 adjust_axes()	60
7.16.1.2 draw_plot_func()	60
7.16.1.3 draw_text_in_image()	62
7.16.1.4 error()	62
7.16.1.5 file_lines_to_list()	63
7.16.1.6 is_float_between_0_and_1()	63
7.16.1.7 log_average_miss_rate()	63
7.16.1.8 voc_ap()	64
7.16.2 변수 문서화	65
7.16.2.1 _	65

7.16.2.2 _difficult	66
7.16.2.3 action	66
7.16.2.4 alpha	66
7.16.2.5 already_seen_classes	66
7.16.2.6 ap	66
7.16.2.7 ap_dictionary	66
7.16.2.8 area_under_curve_x	67
7.16.2.9 area_under_curve_y	67
7.16.2.10 args	67
7.16.2.11 axes	67
7.16.2.12 bb	67
7.16.2.13 bbggt	67
7.16.2.14 bbox	68
7.16.2.15 bi	68
7.16.2.16 BLACK	68
7.16.2.17 bottom	68
7.16.2.18 bottom_border	68
7.16.2.19 bounding_boxes	68
7.16.2.20 class_name	69
7.16.2.21 color	69
7.16.2.22 confidence	69
7.16.2.23 count_non_zero_values_in_dictionary	69
7.16.2.24 count_true_positives	69
7.16.2.25 counter_images_per_class	69
7.16.2.26 cumsum	70
7.16.2.27 det_counter_per_class	70
7.16.2.28 dr_classes	70
7.16.2.29 dr_data	70
7.16.2.30 dr_file	70
7.16.2.31 dr_files_list	70
7.16.2.32 DR_PATH	71
7.16.2.33 draw_plot	71
7.16.2.34 edgecolor	71
7.16.2.35 error_msg	71
7.16.2.36 f	71
7.16.2.37 faster_rcnn	71
7.16.2.38 fig	72
7.16.2.39 file_id	72
7.16.2.40 font	72
7.16.2.41 fp	72
7.16.2.42 fppi	72
7.16.2.43 green	72

7.16.2.44 ground_truth_data	73
7.16.2.45 ground_truth_files_list	73
7.16.2.46 ground_truth_img	73
7.16.2.47 gt_bboxes	73
7.16.2.48 gt_classes	73
7.16.2.49 gt_counter_per_class	73
7.16.2.50 gt_difficults	74
7.16.2.51 gt_file	74
7.16.2.52 gt_labels	74
7.16.2.53 gt_match	74
7.16.2.54 GT_PATH	74
7.16.2.55 height	74
7.16.2.56 help	75
7.16.2.57 ignore	75
7.16.2.58 ih	75
7.16.2.59 img	75
7.16.2.60 img_cumulative	75
7.16.2.61 img_cumulative_path	75
7.16.2.62 IMG_PATH	76
7.16.2.63 index	76
7.16.2.64 iou_list	76
7.16.2.65 is_difficult	76
7.16.2.66 iw	76
7.16.2.67 key	76
7.16.2.68 lamr	77
7.16.2.69 lamr_dictionary	77
7.16.2.70 left	77
7.16.2.71 light_blue	77
7.16.2.72 light_red	77
7.16.2.73 line_width	77
7.16.2.74 lines	78
7.16.2.75 lines_list	78
7.16.2.76 mAP	78
7.16.2.77 margin	78
7.16.2.78 min_overlap	78
7.16.2.79 MINOVERLAP	78
7.16.2.80 mprec	79
7.16.2.81 mr	79
7.16.2.82 mrec	79
7.16.2.83 n_args	79
7.16.2.84 n_classes	79
7.16.2.85 n_det	79

7.16.2.86 n_images	80
7.16.2.87 nargs	80
7.16.2.88 nd	80
7.16.2.89 no_animation	80
7.16.2.90 no_plot	80
7.16.2.91 output_img_path	80
7.16.2.92 output_path	81
7.16.2.93 ov	81
7.16.2.94 ovmax	81
7.16.2.95 parser	81
7.16.2.96 plot_color	81
7.16.2.97 plot_title	81
7.16.2.98 prec	82
7.16.2.99 pred_bboxes	82
7.16.2.100 pred_bboxes_	82
7.16.2.101 pred_labels	82
7.16.2.102 pred_labels_	82
7.16.2.103 pred_scores	82
7.16.2.104 pred_scores_	83
7.16.2.105 rank_pos	83
7.16.2.106 rec	83
7.16.2.107 results_files_path	83
7.16.2.108 reverse	83
7.16.2.109 right	83
7.16.2.110 rounded_prec	84
7.16.2.111 rounded_rec	84
7.16.2.112 show_animation	84
7.16.2.113 sizes	84
7.16.2.114 specific_iou_classes	84
7.16.2.115 specific_iou_flagged	84
7.16.2.116 status	85
7.16.2.117 str	85
7.16.2.118 sum_AP	85
7.16.2.119 TEMP_FILES_PATH	85
7.16.2.120 temp_path	85
7.16.2.121 test_dataloader	85
7.16.2.122 testset	86
7.16.2.123 text	86
7.16.2.124 tmp_class_name	86
7.16.2.125 to_show	86
7.16.2.126 top	86
7.16.2.127 tp	86

7.16.2.128 trainer	87
7.16.2.129 true_p_bar	87
7.16.2.130 type	87
7.16.2.131 ua	87
7.16.2.132 v_pos	87
7.16.2.133 white	87
7.16.2.134 widht	88
7.16.2.135 window_title	88
7.16.2.136 x_label	88
7.17 model 네임스페이스 참조	88
7.18 model.faster_rcnn 네임스페이스 참조	88
7.19 model.roi_module 네임스페이스 참조	88
7.19.1 함수 문서화	89
7.19.1.1 GET_BLOCKS()	89
7.19.1.2 load_kernel()	89
7.19.1.3 normal_init()	90
7.19.2 변수 문서화	90
7.19.2.1 CUDA_NUM_THREADS	90
7.19.2.2 for_each_device	90
7.19.2.3 kernel_backward	90
7.19.2.4 kernel_forward	91
7.19.2.5 Stream	91
7.20 model.rpn 네임스페이스 참조	91
7.20.1 함수 문서화	91
7.20.1.1 normal_init()	91
7.21 model.utils 네임스페이스 참조	92
7.22 model.utils.bbox_tools 네임스페이스 참조	92
7.22.1 함수 문서화	92
7.22.1.1 bbox2loc()	92
7.22.1.2 bbox_iou()	94
7.22.1.3 loc2bbox()	95
7.23 model.utils.nms 네임스페이스 참조	97
7.24 model.utils.nms._nms_gpu_post_py 네임스페이스 참조	97
7.24.1 함수 문서화	97
7.24.1.1 _nms_gpu_post()	98
7.25 model.utils.nms.build 네임스페이스 참조	98
7.25.1 변수 문서화	98
7.25.1.1 cmdclass	98
7.25.1.2 ext_modules	99
7.25.1.3 name	99
7.26 model.utils.nms.non_maximum_suppression 네임스페이스 참조	99
7.26.1 함수 문서화	99

7.26.1.1 _call_nms_kernel()	100
7.26.1.2 _load_kernel()	100
7.26.1.3 _non_maximum_suppression_gpu()	101
7.26.1.4 non_maximum_suppression()	102
7.26.2 변수 문서화	103
7.26.2.1 _nms_gpu_code	104
7.26.2.2 for_each_device	104
7.27 model.utils.roi_sample 네임스페이스 참조	104
7.28 model.utils.rpn_gt_loc_label 네임스페이스 참조	104
7.28.1 함수 문서화	104
7.28.1.1 _get_inside_index()	104
7.28.1.2 _unmap()	105
7.29 model.vgg16 네임스페이스 참조	106
7.29.1 함수 문서화	106
7.29.1.1 decom_vgg16()	106
7.30 Train 네임스페이스 참조	107
7.30.1 함수 문서화	107
7.30.1.1 eval()	108
7.30.2 변수 문서화	109
7.30.2.1 bbox	109
7.30.2.2 best_map	110
7.30.2.3 dataloader	110
7.30.2.4 dataset	110
7.30.2.5 eval_result	110
7.30.2.6 faster_rcnn	110
7.30.2.7 img	111
7.30.2.8 label	111
7.30.2.9 loss_list	111
7.30.2.10 loss_list_roi_cls	111
7.30.2.11 loss_list_roi_loc	111
7.30.2.12 loss_list_rpn_cls	111
7.30.2.13 loss_list_rpn_loc	112
7.30.2.14 lr_	112
7.30.2.15 save_optimizer	112
7.30.2.16 scale	112
7.30.2.17 test_dataloader	112
7.30.2.18 testset	113
7.30.2.19 trainer	113
7.30.2.20 True	113
7.31 trainer 네임스페이스 참조	113
7.32 trainer.trainer 네임스페이스 참조	113
7.32.1 함수 문서화	114

7.32.1.1 <code>_fast_rcnn_loc_loss()</code>	114
7.32.1.2 <code>_smooth_l1_loss()</code>	114
7.32.2 변수 문서화	115
7.32.2.1 <code>LossTuple</code>	115
7.33 <code>utils</code> 네임스페이스 참조	115
7.34 <code>utils.anchors</code> 네임스페이스 참조	116
7.34.1 함수 문서화	116
7.34.1.1 <code>generate_anchor_base()</code>	116
7.34.1.2 <code>get_anchors()</code>	116
7.34.1.3 <code>get_rois_from_loc_anchors()</code>	117
7.35 <code>utils.array_tool</code> 네임스페이스 참조	119
7.35.1 상세한 설명	119
7.35.2 함수 문서화	119
7.35.2.1 <code>scalar()</code>	120
7.35.2.2 <code>tonumpy()</code>	120
7.35.2.3 <code>totensor()</code>	120
7.35.2.4 <code>tovariable()</code>	121
7.36 <code>utils.config</code> 네임스페이스 참조	121
7.36.1 변수 문서화	121
7.36.1.1 <code>opt</code>	121
7.37 <code>utils.data_load</code> 네임스페이스 참조	122
7.37.1 함수 문서화	122
7.37.1.1 <code>load_pkl()</code>	122
7.37.1.2 <code>save_pkl()</code>	122
7.37.2 변수 문서화	123
7.37.2.1 <code>VOC_BBOX_LABEL_NAMES</code>	123
7.38 <code>utils.eval_tool</code> 네임스페이스 참조	123
7.38.1 함수 문서화	123
7.38.1.1 <code>calc_detection_voc_ap()</code>	124
7.38.1.2 <code>calc_detection_voc_prec_rec()</code>	125
7.38.1.3 <code>eval_detection_voc()</code>	128
7.39 <code>utils.py_nms</code> 네임스페이스 참조	131
7.39.1 함수 문서화	131
7.39.1.1 <code>py_cpu_nms()</code>	131
7.40 <code>utils.vis_tool</code> 네임스페이스 참조	132
7.40.1 함수 문서화	132
7.40.1.1 <code>fig2data()</code>	133
7.40.1.2 <code>fig4vis()</code>	133
7.40.1.3 <code>vis_bbox()</code>	134
7.40.1.4 <code>vis_image()</code>	136
7.40.1.5 <code>visdom_bbox()</code>	137
7.40.2 변수 문서화	137

7.40.2.1 VOC_BBOX_LABEL_NAMES	138
8 클래스 문서화	139
8.1 model.utils.rpn_gt_loc_label.AnchorTargetCreator 클래스 참조	139
8.1.1 상세한 설명	141
8.1.2 생성자 & 소멸자 문서화	141
8.1.2.1 __init__()	141
8.1.3 멤버 함수 문서화	141
8.1.3.1 __call__()	141
8.1.3.2 _calc_iious()	143
8.1.3.3 _create_label()	144
8.1.4 멤버 데이터 문서화	145
8.1.4.1 n_sample	145
8.1.4.2 neg_iou_thresh	146
8.1.4.3 pos_iou_thresh	146
8.1.4.4 pos_ratio	146
8.2 utils.config.Config 클래스 참조	146
8.2.1 상세한 설명	147
8.2.2 멤버 함수 문서화	147
8.2.2.1 _parse()	147
8.2.2.2 _state_dict()	148
8.2.3 멤버 데이터 문서화	148
8.2.3.1 caffe_pretrain	148
8.2.3.2 caffe_pretrain_path	149
8.2.3.3 epoch	149
8.2.3.4 load_path	149
8.2.3.5 lr	149
8.2.3.6 lr_decay	149
8.2.3.7 max_size	149
8.2.3.8 min_size	150
8.2.3.9 num_workers	150
8.2.3.10 roi_sigma	150
8.2.3.11 rpn_sigma	150
8.2.3.12 test_num	150
8.2.3.13 test_num_workers	150
8.2.3.14 use_adam	151
8.2.3.15 voc_data_dir	151
8.2.3.16 weight_decay	151
8.3 data.dataset.Dataset 클래스 참조	151
8.3.1 상세한 설명	152
8.3.2 생성자 & 소멸자 문서화	152
8.3.2.1 __init__()	152

8.3.3 멤버 함수 문서화	152
8.3.3.1 <code>__getitem__()</code>	153
8.3.3.2 <code>__len__()</code>	153
8.3.4 멤버 데이터 문서화	153
8.3.4.1 <code>db</code>	153
8.3.4.2 <code>opt</code>	154
8.3.4.3 <code>tsf</code>	154
8.4 <code>model.faster_rcnn.FasterRCNN</code> 클래스 참조	154
8.4.1 상세한 설명	156
8.4.2 생성자 & 소멸자 문서화	156
8.4.2.1 <code>__init__()</code>	156
8.4.3 멤버 함수 문서화	157
8.4.3.1 <code>_suppress()</code>	157
8.4.3.2 <code>forward()</code>	158
8.4.3.3 <code>get_optimizer()</code> [1/2]	159
8.4.3.4 <code>get_optimizer()</code> [2/2]	159
8.4.3.5 <code>n_class()</code>	160
8.4.3.6 <code>predict()</code>	161
8.4.3.7 <code>scale_lr()</code>	162
8.4.3.8 <code>use_preset()</code>	163
8.4.4 멤버 데이터 문서화	164
8.4.4.1 <code>extractor</code>	164
8.4.4.2 <code>head</code>	164
8.4.4.3 <code>loc_normalize_mean</code>	164
8.4.4.4 <code>loc_normalize_std</code>	164
8.4.4.5 <code>nms_thresh</code>	164
8.4.4.6 <code>optimizer</code>	165
8.4.4.7 <code>rpn</code>	165
8.4.4.8 <code>score_thresh</code>	165
8.5 <code>trainer.trainer.FasterRCNNTrainer</code> 클래스 참조	166
8.5.1 상세한 설명	168
8.5.2 생성자 & 소멸자 문서화	168
8.5.2.1 <code>__init__()</code>	168
8.5.3 멤버 함수 문서화	168
8.5.3.1 <code>forward()</code>	169
8.5.3.2 <code>get_meter_data()</code>	171
8.5.3.3 <code>load()</code>	171
8.5.3.4 <code>reset_meters()</code>	172
8.5.3.5 <code>save()</code>	172
8.5.3.6 <code>train_step()</code>	173
8.5.3.7 <code>update_meters()</code>	174
8.5.4 멤버 데이터 문서화	174

8.5.4.1 anchor_target_creator	174
8.5.4.2 faster_rcnn	174
8.5.4.3 loc_normalize_mean	175
8.5.4.4 loc_normalize_std	175
8.5.4.5 optimizer	175
8.5.4.6 proposal_target_creator	175
8.5.4.7 roi_sigma	175
8.5.4.8 rpn_sigma	175
8.6 model.utils.roi_sample.ProposalTargetCreator 클래스 참조	176
8.6.1 상세한 설명	177
8.6.2 생성자 & 소멸자 문서화	178
8.6.2.1 __init__()	178
8.6.3 멤버 함수 문서화	178
8.6.3.1 __call__()	178
8.6.4 멤버 데이터 문서화	181
8.6.4.1 n_sample	181
8.6.4.2 neg_iou_thresh_hi	181
8.6.4.3 neg_iou_thresh_lo	181
8.6.4.4 pos_iou_thresh	182
8.6.4.5 pos_ratio	182
8.7 model.rpn.RegionProposalNetwork 클래스 참조	182
8.7.1 상세한 설명	183
8.7.2 생성자 & 소멸자 문서화	184
8.7.2.1 __init__()	184
8.7.3 멤버 함수 문서화	184
8.7.3.1 forward()	185
8.7.4 멤버 데이터 문서화	187
8.7.4.1 anchor_base	187
8.7.4.2 conv1	188
8.7.4.3 feat_stride	188
8.7.4.4 loc	188
8.7.4.5 score	188
8.8 model.roi_module.Roi 클래스 참조	189
8.8.1 상세한 설명	190
8.8.2 생성자 & 소멸자 문서화	191
8.8.2.1 __init__()	191
8.8.3 멤버 함수 문서화	191
8.8.3.1 backward()	191
8.8.3.2 forward()	192
8.8.4 멤버 데이터 문서화	192
8.8.4.1 argmax_data	193
8.8.4.2 backward_fn	193

8.8.4.3 forward_fn	193
8.8.4.4 in_size	193
8.8.4.5 N	193
8.8.4.6 rois	193
8.8.4.7 spatial_scale	194
8.9 model.roi_module.RoIPooling2D 클래스 참조	194
8.9.1 상세한 설명	195
8.9.2 생성자 & 소멸자 문서화	195
8.9.2.1 __init__()	195
8.9.3 멤버 함수 문서화	196
8.9.3.1 forward()	196
8.9.4 멤버 데이터 문서화	196
8.9.4.1 Rol	196
8.10 data.dataset.TestDataset 클래스 참조	197
8.10.1 상세한 설명	197
8.10.2 생성자 & 소멸자 문서화	197
8.10.2.1 __init__()	197
8.10.3 멤버 함수 문서화	198
8.10.3.1 __getitem__()	198
8.10.3.2 __len__()	198
8.10.4 멤버 데이터 문서화	198
8.10.4.1 db	198
8.10.4.2 opt	199
8.11 data.dataset.Transform 클래스 참조	199
8.11.1 상세한 설명	200
8.11.2 생성자 & 소멸자 문서화	200
8.11.2.1 __init__()	201
8.11.3 멤버 함수 문서화	201
8.11.3.1 __call__()	201
8.11.4 멤버 데이터 문서화	201
8.11.4.1 max_size	202
8.11.4.2 min_size	202
8.12 model.roi_module.VGG16RoIHead 클래스 참조	202
8.12.1 상세한 설명	203
8.12.2 생성자 & 소멸자 문서화	204
8.12.2.1 __init__()	204
8.12.3 멤버 함수 문서화	204
8.12.3.1 forward()	204
8.12.4 멤버 데이터 문서화	206
8.12.4.1 classifier	206
8.12.4.2 cls_loc	206
8.12.4.3 n_class	206

8.12.4.4 roi	206
8.12.4.5 roi_size	206
8.12.4.6 score	207
8.12.4.7 spatial_scale	207
8.13 utils.vis_tool.Visualizer 클래스 참조	207
8.13.1 상세한 설명	209
8.13.2 생성자 & 소멸자 문서화	209
8.13.2.1 __init__()	209
8.13.3 멤버 함수 문서화	209
8.13.3.1 __getattr__()	209
8.13.3.2 img()	210
8.13.3.3 img_many()	210
8.13.3.4 load_state_dict()	211
8.13.3.5 log()	211
8.13.3.6 plot()	211
8.13.3.7 plot_many()	212
8.13.3.8 reinit()	213
8.13.3.9 state_dict()	213
8.13.4 멤버 데이터 문서화	213
8.13.4.1 _vis_kw	213
8.13.4.2 index	213
8.13.4.3 log_text	214
8.13.4.4 vis	214
8.14 data.voc_dataset.VOCBboxDataset 클래스 참조	214
8.14.1 상세한 설명	215
8.14.2 생성자 & 소멸자 문서화	216
8.14.2.1 __init__()	216
8.14.3 멤버 함수 문서화	216
8.14.3.1 __len__()	216
8.14.3.2 get_example()	216
8.14.4 멤버 데이터 문서화	218
8.14.4.1 __getitem__	218
8.14.4.2 data_dir	218
8.14.4.3 ids	218
8.14.4.4 label_names	218
8.14.4.5 return_difficult	219
8.14.4.6 use_difficult	219
8.15 utils.data_load.VOCBboxDataset 클래스 참조	219
8.15.1 상세한 설명	220
8.15.2 생성자 & 소멸자 문서화	221
8.15.2.1 __init__()	221
8.15.3 멤버 함수 문서화	221

8.15.3.1	<code>__len__()</code>	221
8.15.3.2	<code>get_example()</code>	221
8.15.4	멤버 데이터 문서화	222
8.15.4.1	<code>__getitem__</code>	222
8.15.4.2	<code>data_dir</code>	223
8.15.4.3	<code>ids</code>	223
8.15.4.4	<code>label_names</code>	223
8.15.4.5	<code>return_difficult</code>	223
8.15.4.6	<code>use_difficult</code>	223
8.16	<code>utils.data_load.VOCDataSet</code> 클래스 참조	224
8.16.1	상세한 설명	225
8.16.2	생성자 & 소멸자 문서화	225
8.16.2.1	<code>__init__()</code>	225
8.16.3	멤버 함수 문서화	225
8.16.3.1	<code>__getitem__()</code>	226
8.16.3.2	<code>__len__()</code>	226
8.16.4	멤버 데이터 문서화	226
8.16.4.1	<code>db</code>	227
8.16.4.2	<code>opt</code>	227
8.16.4.3	<code>train</code>	227
9	파일 문서화	229
9.1	<code>data/__init__.py</code> 파일 참조	229
9.2	<code>model/__init__.py</code> 파일 참조	229
9.3	<code>model/utils/__init__.py</code> 파일 참조	229
9.4	<code>model/utils/nms/__init__.py</code> 파일 참조	229
9.5	<code>trainer/__init__.py</code> 파일 참조	229
9.6	<code>utils/__init__.py</code> 파일 참조	230
9.7	<code>data/dataset.py</code> 파일 참조	230
9.8	<code>data/util.py</code> 파일 참조	230
9.9	<code>data/voc_dataset.py</code> 파일 참조	231
9.10	<code>data/VOCdevkit/create_segmentations_from_detections.m</code> 파일 참조	231
9.10.1	함수 문서화	232
9.10.1.1	<code>addpath()</code>	232
9.10.1.2	<code>copyfile()</code>	232
9.10.1.3	<code>CREATE_SEGMENTATIONS_FROM_DETECTIONS()</code> [1/2]	233
9.10.1.4	<code>CREATE_SEGMENTATIONS_FROM_DETECTIONS()</code> [2/2]	233
9.10.1.5	<code>create_segmentations_from_detections()</code>	233
9.10.1.6	<code>im()</code>	233
9.10.1.7	<code>imwrite()</code> [1/2]	233
9.10.1.8	<code>imwrite()</code> [2/2]	233
9.10.1.9	<code>isempty()</code>	234

9.10.1.10 png()	234
9.10.1.11 uint8()	234
9.10.1.12 ~exist() [1/3]	234
9.10.1.13 ~exist() [2/3]	234
9.10.1.14 ~exist() [3/3]	234
9.10.1.15 ~strcmp()	235
9.10.2 변수 문서화	235
9.10.2.1 bbox	235
9.10.2.2 BBOXS	235
9.10.2.3 classim	235
9.10.2.4 classlabelfile	235
9.10.2.5 clsnum	236
9.10.2.6 cmap	236
9.10.2.7 conf	236
9.10.2.8 detnum	236
9.10.2.9 drawnow	236
9.10.2.10 end	236
9.10.2.11 gtclasslabelfile	237
9.10.2.12 imgfile	237
9.10.2.13 imgid	237
9.10.2.14 imgids	237
9.10.2.15 imginfo	237
9.10.2.16 imname	237
9.10.2.17 ind	238
9.10.2.18 instlabelfile	238
9.10.2.19 j	238
9.10.2.20 previd	238
9.10.2.21 resultsdir	238
9.10.2.22 resultsdirinst	239
9.10.2.23 resultsfile	239
9.10.2.24 tic	239
9.10.2.25 VOCinit	239
9.11 data/VOCdevkit/example_classifier.m 파일 참조	239
9.11.1 함수 문서화	240
9.11.1.1 addpath()	240
9.11.1.2 fclose()	240
9.11.1.3 FD()	240
9.11.1.4 fprintf() [1/2]	241
9.11.1.5 fprintf() [2/2]	241
9.11.1.6 load()	241
9.11.1.7 save()	241
9.11.1.8 test()	241

9.11.2 변수 문서화	241
9.11.2.1 c	242
9.11.2.2 classifier	242
9.11.2.3 d	242
9.11.2.4 dn	242
9.11.2.5 dp	242
9.11.2.6 drawnow	242
9.11.2.7 extractor	243
9.11.2.8 fd	243
9.11.2.9 fid	243
9.11.2.10 gt	243
9.11.2.11 i	243
9.11.2.12 l	243
9.11.2.13 tic	244
9.11.2.14 VOCinit	244
9.12 data/VOCdevkit/example_detector.m 파일 참조	244
9.12.1 함수 문서화	244
9.12.1.1 addpath()	244
9.12.1.2 fprintf()	245
9.12.1.3 test()	245
9.12.2 변수 문서화	245
9.12.2.1 detector	245
9.12.2.2 drawnow	245
9.12.2.3 i	245
9.12.2.4 rec	246
9.12.2.5 tic	246
9.12.2.6 VOCinit	246
9.13 data/VOCdevkit/example_layout.m 파일 참조	246
9.13.1 함수 문서화	246
9.13.1.1 addpath()	246
9.13.1.2 test()	247
9.13.2 변수 문서화	247
9.13.2.1 cls	247
9.13.2.2 detector	247
9.13.2.3 VOCinit	247
9.14 data/VOCdevkit/example_segmenter.m 파일 참조	247
9.14.1 함수 문서화	248
9.14.1.1 create_segmentations_from_detections()	248
9.14.2 변수 문서화	248
9.14.2.1 comp3	248
9.15 data/VOCdevkit/viewanno.m 파일 참조	248
9.15.1 함수 문서화	249

9.15.1.1	addpath()	249
9.15.1.2	bb()	249
9.15.1.3	fprintf()	249
9.15.1.4	imshow() [1/3]	250
9.15.1.5	imshow() [2/3]	250
9.15.1.6	imshow() [3/3]	250
9.15.1.7	objects()	250
9.15.1.8	plot()	250
9.15.1.9	subplot() [1/3]	250
9.15.1.10	subplot() [2/3]	251
9.15.1.11	subplot() [3/3]	251
9.15.1.12	text() [1/2]	251
9.15.1.13	text() [2/2]	251
9.15.1.14	title() [1/3]	252
9.15.1.15	title() [2/3]	252
9.15.1.16	title() [3/3]	252
9.15.2	변수 문서화	252
9.15.2.1	__pad0__	252
9.15.2.2	__pad1__	252
9.15.2.3	clf	252
9.15.2.4	difficult	253
9.15.2.5	i	253
9.15.2.6	l	253
9.15.2.7	image	253
9.15.2.8	j	253
9.15.2.9	k	254
9.15.2.10	linewidth	254
9.15.2.11	ls	254
9.15.2.12	off	254
9.15.2.13	on	254
9.15.2.14	pause	254
9.15.2.15	segmentations	255
9.15.2.16	set	255
9.15.2.17	truncated	255
9.15.2.18	VOCinit	255
9.16	data/VOCdevkit/viewdet.m 파일 참조	255
9.16.1	함수 문서화	256
9.16.1.1	fprintf()	256
9.16.1.2	gt()	256
9.16.1.3	isempty()	257
9.16.1.4	length()	257
9.16.1.5	max()	257

9.16.1.6 min() [1/2]	257
9.16.1.7 min() [2/2]	258
9.16.1.8 plot() [1/3]	258
9.16.1.9 plot() [2/3]	258
9.16.1.10 plot() [3/3]	259
9.16.2 변수 문서화	259
9.16.2.1 BB	259
9.16.2.2 bb	259
9.16.2.3 bi	259
9.16.2.4 clsinds	259
9.16.2.5 confidence	260
9.16.2.6 d	260
9.16.2.7 drawnow	260
9.16.2.8 i	260
9.16.2.9 ids	260
9.16.2.10 ih	261
9.16.2.11 image	261
9.16.2.12 iw	261
9.16.2.13 j	261
9.16.2.14 jmax	261
9.16.2.15 nd	261
9.16.2.16 npos	262
9.16.2.17 off	262
9.16.2.18 ov	262
9.16.2.19 ovmax	262
9.16.2.20 ovmax< VOCopts.minoverlap continue end % read image I=imread(sprintf(VOCopts.imgpath, gtids{i}));% draw detection bounding box and ground truth bounding box(if any) imagesc(I);hold on;if ovmax >	262
9.16.2.21 rec	263
9.16.2.22 results	263
9.16.2.23 tic	263
9.16.2.24 ua	263
9.17 data/VOCdevkit/VOCcode/PASemptyobject.m 파일 참조	263
9.17.1 변수 문서화	264
9.17.1.1 bbox	264
9.17.1.2 class	264
9.17.1.3 difficult	264
9.17.1.4 mask	264
9.17.1.5 object	264
9.17.1.6 orglabel	265
9.17.1.7 polygon	265
9.17.1.8 truncated	265
9.17.1.9 view	265

9.18 data/VOCdevkit/VOCcode/PASemptyrecord.m 파일 참조	265
9.18.1 변수 문서화	265
9.18.1.1 database	266
9.18.1.2 imgsize	266
9.18.1.3 objects	266
9.18.1.4 record	266
9.19 data/VOCdevkit/VOCcode/PASerrmsg.m 파일 참조	266
9.19.1 함수 문서화	267
9.19.1.1 fprintf() [1/2]	267
9.19.1.2 fprintf() [2/2]	267
9.19.1.3 if() [1/2]	267
9.19.1.4 if() [2/2]	267
9.19.2 변수 문서화	267
9.19.2.1 end	267
9.19.2.2 k	268
9.19.2.3 keyboard	268
9.20 data/VOCdevkit/VOCcode/PASreadrecord.m 파일 참조	268
9.20.1 함수 문서화	268
9.20.1.1 strcmp()	268
9.20.2 변수 문서화	268
9.20.2.1 rec	268
9.21 data/VOCdevkit/VOCcode/PASreadrectxt.m 파일 참조	269
9.21.1 함수 문서화	269
9.21.1.1 if() [1/2]	269
9.21.1.2 if() [2/2]	269
9.21.1.3 objects()	269
9.21.1.4 PASerrmsg()	270
9.21.1.5 while()	270
9.21.2 변수 문서화	270
9.21.2.1 case	270
9.21.2.2 database	270
9.21.2.3 end	270
9.21.2.4 imgname	270
9.21.2.5 imgsize	271
9.21.2.6 line	271
9.21.2.7 matchnum	271
9.21.2.8 matchstrs	271
9.21.2.9 notEOF	271
9.21.2.10 PASmsg	271
9.21.2.11 record	272
9.21.2.12 tmp	272
9.22 data/VOCdevkit/VOCcode/VOCevalcls.m 파일 참조	272

9.22.1 함수 문서화	273
9.22.1.1 isempty() [1/2]	273
9.22.1.2 isempty() [2/2]	273
9.22.1.3 length()	273
9.22.1.4 out()	273
9.22.1.5 plot()	273
9.22.1.6 title()	273
9.22.2 변수 문서화	274
9.22.2.1 ap	274
9.22.2.2 drawnow	274
9.22.2.3 fp	274
9.22.2.4 function	274
9.22.2.5 grid	274
9.22.2.6 i	275
9.22.2.7 ids	275
9.22.2.8 j	275
9.22.2.9 out	275
9.22.2.10 prec	275
9.22.2.11 rec	276
9.22.2.12 recall	276
9.22.2.13 results	276
9.22.2.14 t	276
9.22.2.15 tic	276
9.22.2.16 tp	276
9.23 data/VOCdevkit/VOCcode/VOCevaldet.m 파일 참조	277
9.23.1 함수 문서화	278
9.23.1.1 fp()	278
9.23.1.2 gt() [1/2]	278
9.23.1.3 gt() [2/2]	278
9.23.1.4 isempty() [1/2]	278
9.23.1.5 isempty() [2/2]	278
9.23.1.6 length()	278
9.23.1.7 max()	279
9.23.1.8 min() [1/2]	279
9.23.1.9 min() [2/2]	279
9.23.1.10 plot()	279
9.23.1.11 positive()	279
9.23.1.12 title()	279
9.23.2 변수 문서화	280
9.23.2.1 ap	280
9.23.2.2 BB	280
9.23.2.3 bb	280

9.23.2.4 bi	280
9.23.2.5 clsinds	280
9.23.2.6 confidence	281
9.23.2.7 d	281
9.23.2.8 drawnow	281
9.23.2.9 fp	281
9.23.2.10 function	281
9.23.2.11 grid	282
9.23.2.12 i	282
9.23.2.13 ids	282
9.23.2.14 ih	282
9.23.2.15 iw	282
9.23.2.16 j	283
9.23.2.17 jmax	283
9.23.2.18 nd	283
9.23.2.19 npos	283
9.23.2.20 ov	283
9.23.2.21 ovmax	283
9.23.2.22 prec	284
9.23.2.23 rec	284
9.23.2.24 results	284
9.23.2.25 t	284
9.23.2.26 tic	284
9.23.2.27 tp	284
9.23.2.28 ua	285
9.24 data/VOCdevkit/VOCcode/VOCevallayout.m 파일 참조	285
9.24.1 함수 문서화	286
9.24.1.1 dp()	286
9.24.1.2 fp()	286
9.24.1.3 fprintf()	286
9.24.1.4 gt()	287
9.24.1.5 if()	287
9.24.1.6 isempty() [1/2]	287
9.24.1.7 isempty() [2/2]	287
9.24.1.8 length()	287
9.24.1.9 M()	287
9.24.1.10 max()	288
9.24.1.11 min() [1/2]	288
9.24.1.12 min() [2/2]	288
9.24.1.13 plot()	288
9.24.1.14 title()	288
9.24.1.15 tp()	288

9.24.1.16 ~isempty()	289
9.24.2 변수 문서화	289
9.24.2.1 ap	289
9.24.2.2 bb	289
9.24.2.3 bi	289
9.24.2.4 confidence	289
9.24.2.5 d	289
9.24.2.6 di	290
9.24.2.7 dp	290
9.24.2.8 drawnow	290
9.24.2.9 fp	290
9.24.2.10 function	290
9.24.2.11 grid	291
9.24.2.12 i	291
9.24.2.13 id	291
9.24.2.14 ids	291
9.24.2.15 ih	291
9.24.2.16 iw	292
9.24.2.17 j	292
9.24.2.18 jmax	292
9.24.2.19 k	292
9.24.2.20 l	292
9.24.2.21 nd	293
9.24.2.22 npos	293
9.24.2.23 objects	293
9.24.2.24 ov	293
9.24.2.25 ovmax	293
9.24.2.26 parts	293
9.24.2.27 prec	294
9.24.2.28 rec	294
9.24.2.29 res	294
9.24.2.30 t	294
9.24.2.31 tic	294
9.24.2.32 tp	294
9.24.2.33 ua	295
9.24.2.34 v	295
9.25 data/VOCdevkit/VOCcode/VOCevalseg.m 파일 참조	295
9.25.1 함수 문서화	295
9.25.1.1 VOCEVALSEG()	295
9.26 data/VOCdevkit/VOCcode/VOCinit.m 파일 참조	295
9.26.1 변수 문서화	296
9.26.1.1 cwd	296

9.26.1.2 dataset	296
9.26.1.3 VOC2006	296
9.27 data/VOCdevkit/VOCcode/VOClabelcolormap.m 파일 참조	296
9.27.1 함수 문서화	297
9.27.1.1 cmap() [1/3]	297
9.27.1.2 cmap() [2/3]	297
9.27.1.3 cmap() [3/3]	297
9.27.2 변수 문서화	297
9.27.2.1 b	297
9.27.2.2 CMAP	297
9.27.2.3 cmap	298
9.27.2.4 g	298
9.27.2.5 i	298
9.27.2.6 id	298
9.27.2.7 indices	298
9.27.2.8 j	299
9.27.2.9 r	299
9.28 data/VOCdevkit/VOCcode/VOCreadrecxml.m 파일 참조	299
9.28.1 함수 문서화	300
9.28.1.1 isfield() [1/6]	300
9.28.1.2 isfield() [2/6]	300
9.28.1.3 isfield() [3/6]	300
9.28.1.4 isfield() [4/6]	300
9.28.1.5 isfield() [5/6]	300
9.28.1.6 isfield() [6/6]	301
9.28.2 변수 문서화	301
9.28.2.1 bbox	301
9.28.2.2 database	301
9.28.2.3 depth	301
9.28.2.4 difficult	301
9.28.2.5 hasparts	301
9.28.2.6 height	302
9.28.2.7 i	302
9.28.2.8 imgname	302
9.28.2.9 imgsize	302
9.28.2.10 label	302
9.28.2.11 mask	302
9.28.2.12 orglabel	303
9.28.2.13 p	303
9.28.2.14 part	303
9.28.2.15 polygon	303
9.28.2.16 rec	303

9.28.2.17 segmented	304
9.28.2.18 truncated	304
9.28.2.19 view	304
9.28.2.20 width	304
9.28.2.21 x	304
9.28.2.22 xmax	304
9.28.2.23 xmin	305
9.28.2.24 ymax	305
9.28.2.25 ymin	305
9.29 data/VOCdevkit/VOCcode/VOCreadxml.m 파일 참조	305
9.29.1 변수 문서화	305
9.29.1.1 f	305
9.29.1.2 rec	306
9.30 data/VOCdevkit/VOCcode/VOCwritexml.m 파일 참조	306
9.30.1 함수 문서화	306
9.30.1.1 error()	306
9.30.1.2 fclose()	307
9.30.1.3 fprintf() [1/4]	307
9.30.1.4 fprintf() [2/4]	307
9.30.1.5 fprintf() [3/4]	307
9.30.1.6 fprintf() [4/4]	307
9.30.1.7 ischar()	307
9.30.1.8 isnumeric()	308
9.30.1.9 VOCwritexml()	308
9.30.1.10 writexml() [1/2]	308
9.30.1.11 writexml() [2/2]	308
9.30.1.12 ~iscell()	308
9.30.1.13 ~isempty()	308
9.30.2 변수 문서화	309
9.30.2.1 f	309
9.30.2.2 i	309
9.30.2.3 j	309
9.30.2.4 s	309
9.30.2.5 xml	309
9.31 data/VOCdevkit/VOCcode/VOCxml2struct.m 파일 참조	310
9.31.1 함수 문서화	310
9.31.1.1 ~isempty()	310
9.31.1.2 ~strcmp()	310
9.31.2 변수 문서화	310
9.31.2.1 function	310
9.31.2.2 ind	311
9.31.2.3 parent	311

9.31.2.4 res	311
9.31.2.5 tag	311
9.32 demo.py 파일 참조	311
9.33 Evaluation.py 파일 참조	312
9.34 img_crawling.py 파일 참조	312
9.35 main.py 파일 참조	313
9.36 model/faster_rcnn.py 파일 참조	316
9.37 model/roi_module.py 파일 참조	316
9.38 model/rpn.py 파일 참조	317
9.39 model/utils/bbox_tools.py 파일 참조	317
9.40 model/utils/nms/_nms_gpu_post.c 파일 참조	317
9.40.1 매크로 문서화	318
9.40.1.1 PY_SSIZE_T_CLEAN	318
9.41 model/utils/nms/_nms_gpu_post_py.py 파일 참조	318
9.42 model/utils/nms/build.py 파일 참조	318
9.43 model/utils/nms/non_maximum_suppression.py 파일 참조	318
9.44 model/utils/roi_sample.py 파일 참조	319
9.45 model/utils/rpn_gt_loc_label.py 파일 참조	319
9.46 model/vgg16.py 파일 참조	319
9.47 scripts/extra/convert_dr_darkflow_json.py 파일 참조	320
9.48 scripts/extra/convert_dr_yolo.py 파일 참조	320
9.49 scripts/extra/convert_gt_xml.py 파일 참조	321
9.50 scripts/extra/convert_gt_yolo.py 파일 참조	321
9.51 scripts/extra/convert_keras-yolo3.py 파일 참조	322
9.52 scripts/extra/find_class.py 파일 참조	323
9.53 scripts/extra/intersect-gt-and-dr.py 파일 참조	323
9.54 scripts/extra/README.md 파일 참조	323
9.55 Train.py 파일 참조	323
9.56 trainer/trainer.py 파일 참조	324
9.57 utils/anchors.py 파일 참조	325
9.58 utils/array_tool.py 파일 참조	325
9.59 utils/config.py 파일 참조	325
9.60 utils/data_load.py 파일 참조	326
9.61 utils/eval_tool.py 파일 참조	326
9.62 utils/py_nms.py 파일 참조	326
9.63 utils/vis_tool.py 파일 참조	327
Index	329

Chapter 1

Faster_RCNN (base network: VGG16)

세탁 태그 인식을 위한 Faster RCNN fine tuning code

1.1 작성정보

- 작성자 : 황병훈
- 작성일: 2019 - 11 - 05

Chapter 2

Extra

2.1 ground-truth:

- ### convert xml to our format:
1) Insert ground-truth xml files into ground-truth/ 2) Run the python script: `python convert_gt_xml.py`
- ### convert YOLO to our format:
1) Add class list to the file `class_list.txt` 2) Insert ground-truth files into ground-truth/ 3) Insert images into images/ 4) Run the python script: `python convert_gt_yolo.py`
- ### convert keras-yolo3 to our format:
1) Add or update the class list to the file `class_list.txt` 2) Use the parameter `--gt` to set the ground-truth source. 3) Run the python script: `python3 convert_keras-yolo3.py --gt <gt_file_path>` 1) Supports only python 3. 2) This code can handle recursive annotation structure. Just use the `-r` parameter. 3) The converted annotation is placed by default in a new `from_kerasyolo3` folder. You can change that with the parameter `-o`. 4) The format is defined according with github.com/qpwweee/keras-yolo3

2.2 detection-results:

- ### convert darkflow json to our format:
1) Insert result json files into detection-results/ 2) Run the python script: `python convert_dr_darkflow_json.py`
- ### convert YOLO to our format:
After running darknet on a list of images, e.g.: `darknet.exe detector test data/voc.data yolo-voc.cfg yolo-voc.weights -dont_show -ext_output < data/test.txt > result.txt`
1) Copy the file `result.txt` to the folder `extra/` 2) Run the python script: `python convert_dr_yolo.py`
- ### convert keras-yolo3 to our format:
1) Add or update the class list to the file `class_list.txt` 2) Use the parameter `--dr` to set the detection-results source. 3) Run the python script: `python3 convert_keras-yolo3.py --dr <dr_file_path>` 1) Supports only python 3. 2) This code can handle recursive annotation structure. Just use the `-r` parameter. 3) The converted annotation is placed by default in a new `from_kerasyolo3` folder. You can change that with the parameter `-o`. 4) The format is defined according with github.com/gustavovaliati/keras-yolo3

2.3 Find the files that contain a specific class of objects

1) Run the [find_class.py](#) script and specify the class as argument, e.g. `python find_class.py chair`

2.4 Intersect ground-truth and detection-results files

This script ensures same number of files in ground-truth and detection-results folder. When you encounter file not found error, it's usually because you have mismatched numbers of ground-truth and detection-results files. You can use this script to move ground-truth and detection-results files that are not in the intersection into a backup folder (`backup_no_matches_found`). This will retain only files that have the same name in both folders.

1) Prepare .txt files in your ground-truth and detection-results folders. 2) Run the [intersect-gt-and-dr.py](#) script to move non-intersected files into a backup folder (default: `backup_no_matches_found`).

`python intersect-gt-and-dr.py`

Chapter 3

네임스페이스 색인

3.1 패키지

다음은 패키지들입니다. (가능한한 간략한 설명만을 보여줍니다) :

convert_dr_darkflow_json	13
convert_dr_yolo	15
convert_gt_xml	18
convert_gt_yolo	20
convert_keras	25
convert_keras-yolo3	26
data	31
data.dataset	31
data.util	36
data.voc_dataset	44
demo	44
Evaluation	48
find_class	50
img_crawling	52
intersect-gt-and-dr	54
main	57
model	88
model.faster_rcnn	88
model.roi_module	88
model.rpn	91
model.utils	92
model.utils.bbox_tools	92
model.utils.nms	97
model.utils.nms._nms_gpu_post_py	97
model.utils.nms.build	98
model.utils.nms.non_maximum_suppression	99
model.utils.roi_sample	104
model.utils.rpn_gt_loc_label	104
model.vgg16	106
Train	107
trainer	113
trainer.trainer	113
utils	115
utils.anchors	116
utils.array_tool	119

utils.config	121
utils.data_load	122
utils.eval_tool	123
utils.py_nms	131
utils.vis_tool	132

Chapter 4

계통도 색인

4.1 클래스 계통도

이 상속 목록은 완전하진 않지만 알파벳순으로 대략적으로 정렬되어있습니다.:

utils.config.Config	146
data.dataset.Dataset	151
Module	
model.faster_rcnn.FasterRCNN	154
model.roi_module.RoIPooling2D	194
model.roi_module.VGG16RoIHead	202
model.rpn.RegionProposalNetwork	182
trainer.trainer.FasterRCNNTrainer	166
object	
data.dataset.Transform	199
model.utils.roi_sample.ProposalTargetCreator	176
model.utils.rpn_gt_loc_label.AnchorTargetCreator	139
utils.vis_tool.Visualizer	207
data.dataset.TestDataset	197
data.voc_dataset.VOCBboxDataset	214
utils.data_load.VOCBboxDataset	219
Dataset	
utils.data_load.VOCDataset	224
Function	
model.roi_module.Roi	189

Chapter 5

클래스 색인

5.1 클래스 목록

다음은 클래스, 구조체, 공용체 그리고 인터페이스들입니다. (간략한 설명만을 보여줍니다) :

model.utils.rpn_gt_loc_label.AnchorTargetCreator	139
utils.config.Config	146
data.dataset.Dataset	151
model.faster_rcnn.FasterRCNN	154
trainer.trainer.FasterRCNNTrainer	166
model.utils.roi_sample.ProposalTargetCreator	176
model.rpn.RegionProposalNetwork	182
model.roi_module.Roi	189
model.roi_module.RoiPooling2D	194
data.dataset.TestDataset	197
data.dataset.Transform	
이미지 크기 변경 및 B.B 크기 변경	199
model.roi_module.VGG16RoIHead	202
utils.vis_tool.Visualizer	207
data.voc_dataset.VOCBboxDataset	214
utils.data_load.VOCBboxDataset	219
utils.data_load.VOCDataset	224

Chapter 6

파일 색인

6.1 파일 목록

다음은 모든 파일에 대한 목록입니다. (간략한 설명만을 보여줍니다) :

demo.py	311
Evaluation.py	312
img_crawling.py	312
main.py	313
Train.py	323
data/__init__.py	229
data/dataset.py	230
data/util.py	230
data/voc_dataset.py	231
data/VOCdevkit/create_segmentations_from_detections.m	231
data/VOCdevkit/example_classifier.m	239
data/VOCdevkit/example_detector.m	244
data/VOCdevkit/example_layout.m	246
data/VOCdevkit/example_segmenter.m	247
data/VOCdevkit/viewanno.m	248
data/VOCdevkit/viewdet.m	255
data/VOCdevkit/VOCcode/PASemptyobject.m	263
data/VOCdevkit/VOCcode/PASemptyrecord.m	265
data/VOCdevkit/VOCcode/PASerrmsg.m	266
data/VOCdevkit/VOCcode/PASreadrecord.m	268
data/VOCdevkit/VOCcode/PASreadrectxt.m	269
data/VOCdevkit/VOCcode/VOCevalcls.m	272
data/VOCdevkit/VOCcode/VOCevaldet.m	277
data/VOCdevkit/VOCcode/VOCevallayout.m	285
data/VOCdevkit/VOCcode/VOCevalseg.m	295
data/VOCdevkit/VOCcode/VOCinit.m	295
data/VOCdevkit/VOCcode/VOClabelcolormap.m	296
data/VOCdevkit/VOCcode/VOCreadrecxml.m	299
data/VOCdevkit/VOCcode/VOCreadxml.m	305
data/VOCdevkit/VOCcode/VOCwritexml.m	306
data/VOCdevkit/VOCcode/VOCxml2struct.m	310
model/__init__.py	229
model/faster_rcnn.py	316
model/roi_module.py	316
model/rpn.py	317

model/vgg16.py	319
model/utils/__init__.py	229
model/utils/bbox_tools.py	317
model/utils/roi_sample.py	319
model/utils/rpn_gt_loc_label.py	319
model/utils/nms/__init__.py	229
model/utils/nms/_nms_gpu_post.c	317
model/utils/nms/_nms_gpu_post_py.py	318
model/utils/nms/build.py	318
model/utils/nms/non_maximum_suppression.py	318
scripts/extra/convert_dr_darkflow_json.py	320
scripts/extra/convert_dr_yolo.py	320
scripts/extra/convert_gt_xml.py	321
scripts/extra/convert_gt_yolo.py	321
scripts/extra/convert_keras-yolo3.py	322
scripts/extra/find_class.py	323
scripts/extra/intersect-gt-and-dr.py	323
trainer/__init__.py	229
trainer/trainer.py	324
utils/__init__.py	230
utils/anchors.py	325
utils/array_tool.py	325
utils/config.py	325
utils/data_load.py	326
utils/eval_tool.py	326
utils/py_nms.py	326
utils/vis_tool.py	327

Chapter 7

네임스페이스 문서화

7.1 convert_dr_darkflow_json 네임스페이스 참조

변수

- `parent_path` = `os.path.abspath(os.path.join(os.getcwd(), os.pardir))`
- `DR_PATH` = `os.path.join(parent_path, 'input', 'detection-results')`
- `json_list` = `glob.glob('*.json')`
 - create the backup dir if it doesn't exist already
- `data` = `json.load(open(tmp_file))`
- `obj_name` = `obj['label']`
- `conf` = `obj['confidence']`
- `left` = `obj['topleft']['x']`
- `top` = `obj['topleft']['y']`
- `right` = `obj['bottomright']['x']`
- `bottom` = `obj['bottomright']['y']`

7.1.1 변수 문서화

7.1.1.1 bottom

```
convert_dr_darkflow_json.bottom = obj['bottomright']['y']
```

convert_dr_darkflow_json.py 파일의 37 번째 라인에서 정의되었습니다.

7.1.1.2 conf

```
convert_dr_darkflow_json.conf = obj['confidence']
```

convert_dr_darkflow_json.py 파일의 33 번째 라인에서 정의되었습니다.

7.1.1.3 data

```
convert_dr_darkflow_json.data = json.load(open(tmp_file))
```

convert_dr_darkflow_json.py 파일의 30 번째 라인에서 정의되었습니다.

7.1.1.4 DR_PATH

```
convert_dr_darkflow_json.DR_PATH = os.path.join(parent_path, 'input', 'detection-results')
```

convert_dr_darkflow_json.py 파일의 12 번째 라인에서 정의되었습니다.

7.1.1.5 json_list

```
convert_dr_darkflow_json.json_list = glob.glob('*.json')
```

create the backup dir if it doesn't exist already

convert_dr_darkflow_json.py 파일의 22 번째 라인에서 정의되었습니다.

7.1.1.6 left

```
convert_dr_darkflow_json.left = obj['topleft']['x']
```

convert_dr_darkflow_json.py 파일의 34 번째 라인에서 정의되었습니다.

7.1.1.7 obj_name

```
convert_dr_darkflow_json.obj_name = obj['label']
```

convert_dr_darkflow_json.py 파일의 32 번째 라인에서 정의되었습니다.

7.1.1.8 parent_path

```
convert_dr_darkflow_json.parent_path = os.path.abspath(os.path.join(os.getcwd(), os.pardir))
```

convert_dr_darkflow_json.py 파일의 10 번째 라인에서 정의되었습니다.

7.1.1.9 right

```
convert_dr_darkflow_json.right = obj['bottomright']['x']
```

convert_dr_darkflow_json.py 파일의 36 번째 라인에서 정의되었습니다.

7.1.1.10 top

```
convert_dr_darkflow_json.top = obj['topleft']['y']
```

convert_dr_darkflow_json.py 파일의 35 번째 라인에서 정의되었습니다.

7.2 convert_dr_yolo 네임스페이스 참조

변수

- string `IN_FILE` = 'result.txt'
- `parent_path` = `os.path.abspath(os.path.join(os.getcwd(), os.pardir))`
- `DR_PATH` = `os.path.join(parent_path, 'input', 'detection-results')`
- string `SEPARATOR_KEY` = 'Enter Image Path:'
- string `IMG_FORMAT` = '.jpg'
- `outfile` = None
- `image_path` = `re.search(SEPARATOR_KEY + '(.)' + IMG_FORMAT, line)`
- `image_name` = `os.path.basename(image_path.group(1))`
- `class_name`
- `info`
- `confidence`
- `bbox` = `bbox.replace(' ', '')`
- `left`
- `top`
- `width`
- `height`
- `right` = `left + width`
- `bottom` = `top + height`

7.2.1 변수 문서화

7.2.1.1 bbox

```
convert_dr_yolo.bbox = bbox.replace(' ', '')
```

convert_dr_yolo.py 파일의 38 번째 라인에서 정의되었습니다.

7.2.1.2 bottom

```
convert_dr_yolo.bottom = top + height
```

convert_dr_yolo.py 파일의 44 번째 라인에서 정의되었습니다.

7.2.1.3 class_name

```
convert_dr_yolo.class_name
```

convert_dr_yolo.py 파일의 37 번째 라인에서 정의되었습니다.

7.2.1.4 confidence

```
convert_dr_yolo.confidence
```

convert_dr_yolo.py 파일의 38 번째 라인에서 정의되었습니다.

7.2.1.5 DR_PATH

```
convert_dr_yolo.DR_PATH = os.path.join(parent_path, 'input', 'detection-results')
```

convert_dr_yolo.py 파일의 12 번째 라인에서 정의되었습니다.

7.2.1.6 height

```
convert_dr_yolo.height
```

convert_dr_yolo.py 파일의 42 번째 라인에서 정의되었습니다.

7.2.1.7 image_name

```
convert_dr_yolo.image_name = os.path.basename(image_path.group(1))
```

convert_dr_yolo.py 파일의 29 번째 라인에서 정의되었습니다.

7.2.1.8 image_path

```
convert_dr_yolo.image_path = re.search(SEPARATOR_KEY + '(.*)' + IMG_FORMAT, line)
```

convert_dr_yolo.py 파일의 26 번째 라인에서 정의되었습니다.

7.2.1.9 IMG_FORMAT

```
string convert_dr_yolo.IMG_FORMAT = '.jpg'
```

convert_dr_yolo.py 파일의 17 번째 라인에서 정의되었습니다.

7.2.1.10 IN_FILE

```
string convert_dr_yolo.IN_FILE = 'result.txt'
```

convert_dr_yolo.py 파일의 7 번째 라인에서 정의되었습니다.

7.2.1.11 info

```
convert_dr_yolo.info
```

convert_dr_yolo.py 파일의 37 번째 라인에서 정의되었습니다.

7.2.1.12 left

```
convert_dr_yolo.left
```

convert_dr_yolo.py 파일의 42 번째 라인에서 정의되었습니다.

7.2.1.13 outfile

```
convert_dr_yolo.outfile = None
```

convert_dr_yolo.py 파일의 19 번째 라인에서 정의되었습니다.

7.2.1.14 parent_path

```
convert_dr_yolo.parent_path = os.path.abspath(os.path.join(os.getcwd(), os.pardir))
```

convert_dr_yolo.py 파일의 10 번째 라인에서 정의되었습니다.

7.2.1.15 right

```
convert_dr_yolo.right = left + width
```

convert_dr_yolo.py 파일의 43 번째 라인에서 정의되었습니다.

7.2.1.16 SEPARATOR_KEY

```
string convert_dr_yolo.SEPARATOR_KEY = 'Enter Image Path:'
```

convert_dr_yolo.py 파일의 16 번째 라인에서 정의되었습니다.

7.2.1.17 top

```
convert_dr_yolo.top
```

convert_dr_yolo.py 파일의 42 번째 라인에서 정의되었습니다.

7.2.1.18 width

```
convert_dr_yolo.width
```

convert_dr_yolo.py 파일의 42 번째 라인에서 정의되었습니다.

7.3 convert_gt_xml 네임스페이스 참조

변수

- `parent_path` = `os.path.abspath(os.path.join(os.getcwd(), os.pardir))`
- `GT_PATH` = `os.path.join(parent_path, 'input', 'ground-truth')`
- `xml_list` = `glob.glob('*.xml')`
 - create the backup dir if it doesn't exist already
- `root` = `ET.parse(tmp_file).getroot()`
- `obj_name` = `obj.find('name').text`
- `bndbox` = `obj.find('bndbox')`
- `left` = `bndbox.find('xmin').text`
- `top` = `bndbox.find('ymin').text`
- `right` = `bndbox.find('xmax').text`
- `bottom` = `bndbox.find('ymax').text`

7.3.1 변수 문서화

7.3.1.1 bndbox

```
convert_gt_xml.bndbox = obj.find('bndbox')
```

convert_gt_xml.py 파일의 33 번째 라인에서 정의되었습니다.

7.3.1.2 bottom

```
convert_gt_xml.bottom = bndbox.find('ymax').text
```

convert_gt_xml.py 파일의 37 번째 라인에서 정의되었습니다.

7.3.1.3 GT_PATH

```
convert_gt_xml.GT_PATH = os.path.join(parent_path, 'input', 'ground-truth')
```

convert_gt_xml.py 파일의 12 번째 라인에서 정의되었습니다.

7.3.1.4 left

```
convert_gt_xml.left = bndbox.find('xmin').text
```

convert_gt_xml.py 파일의 34 번째 라인에서 정의되었습니다.

7.3.1.5 obj_name

```
convert_gt_xml.obj_name = obj.find('name').text
```

convert_gt_xml.py 파일의 32 번째 라인에서 정의되었습니다.

7.3.1.6 parent_path

```
convert_gt_xml.parent_path = os.path.abspath(os.path.join(os.getcwd(), os.pardir))
```

convert_gt_xml.py 파일의 10 번째 라인에서 정의되었습니다.

7.3.1.7 right

```
convert_gt_xml.right = bndbox.find('xmax').text
```

convert_gt_xml.py 파일의 36 번째 라인에서 정의되었습니다.

7.3.1.8 root

```
convert_gt_xml.root = ET.parse(tmp_file).getroot()
```

convert_gt_xml.py 파일의 30 번째 라인에서 정의되었습니다.

7.3.1.9 top

```
convert_gt_xml.top = bndbox.find('ymin').text
```

convert_gt_xml.py 파일의 35 번째 라인에서 정의되었습니다.

7.3.1.10 xml_list

```
convert_gt_xml.xml_list = glob.glob('*.xml')
```

create the backup dir if it doesn't exist already

convert_gt_xml.py 파일의 22 번째 라인에서 정의되었습니다.

7.4 convert_gt_yolo 네임스페이스 참조

함수

- def `convert_yolo_coordinates_to_voc` (`x_c_n`, `y_c_n`, `width_n`, `height_n`, `img_width`, `img_height`)

변수

- `obj_list` = `f.readlines()`
remove whitespace characters like `\n` at the end of each line
- `parent_path` = `os.path.abspath(os.path.join(os.getcwd(), os.pardir))`
e.g.
- `GT_PATH` = `os.path.join(parent_path, 'input', 'ground-truth')`
- `txt_list` = `glob.glob('*.txt')`
create the backup dir if it doesn't exist already
- `image_name` = `tmp_file.split(".txt", 1)[0]`
get name before ".txt"
- `img` = `cv2.imread('../images/' + fname)`
check if image exists
- `img_height`
get image width and height
- `img_width`
- `content` = `f.readlines()`
image not found
- `obj_id`
split a line by spaces.
- `x_c_n`
- `y_c_n`
- `width_n`
- `height_n`
- `obj_name` = `obj_list[int(obj_id)]`
- `left`
- `top`
- `right`
- `bottom`

7.4.1 함수 문서화

7.4.1.1 convert_yolo_coordinates_to_voc()

```
def convert_gt_yolo.convert_yolo_coordinates_to_voc (
    x_c_n,
    y_c_n,
    width_n,
    height_n,
    img_width,
    img_height )
```

convert_gt_yolo.py 파일의 6 번째 라인에서 정의되었습니다.

```
6 def convert_yolo_coordinates_to_voc(x_c_n, y_c_n, width_n, height_n, img_width, img_height):
7
8     x_c = float(x_c_n) * img_width
9     y_c = float(y_c_n) * img_height
10    width = float(width_n) * img_width
11    height = float(height_n) * img_height
12
13    half_width = width / 2
14    half_height = height / 2
```

```

15
17 left = int(x_c - half_width) + 1
18 top = int(y_c - half_height) + 1
19 right = int(x_c + half_width) + 1
20 bottom = int(y_c + half_height) + 1
21 return left, top, right, bottom
22
23 # make sure that the cwd() in the beginning is the location of the python script (so that every path
    makes sense)
24 os.chdir(os.path.dirname(os.path.abspath(__file__)))
25
26 # read the class_list.txt to a list

```

7.4.2 변수 문서화

7.4.2.1 bottom

```
convert_gt_yolo.bottom
```

convert_gt_yolo.py 파일의 84 번째 라인에서 정의되었습니다.

7.4.2.2 content

```
list convert_gt_yolo.content = f.readlines()
```

```
image not found
```

remove whitespace characters like \n at the end of each line

convert_gt_yolo.py 파일의 72 번째 라인에서 정의되었습니다.

7.4.2.3 GT_PATH

```
convert_gt_yolo.GT_PATH = os.path.join(parent_path, 'input', 'ground-truth')
```

convert_gt_yolo.py 파일의 37 번째 라인에서 정의되었습니다.

7.4.2.4 height_n

```
convert_gt_yolo.height_n
```

convert_gt_yolo.py 파일의 82 번째 라인에서 정의되었습니다.

7.4.2.5 image_name

```
convert_gt_yolo.image_name = tmp_file.split(".txt",1)[0]
```

get name before ".txt"

convert_gt_yolo.py 파일의 55 번째 라인에서 정의되었습니다.

7.4.2.6 img

```
convert_gt_yolo.img = cv2.imread('../images/' + fname)
```

check if image exists

image found print(fname)

convert_gt_yolo.py 파일의 62 번째 라인에서 정의되었습니다.

7.4.2.7 img_height

```
convert_gt_yolo.img_height
```

get image width and height

convert_gt_yolo.py 파일의 64 번째 라인에서 정의되었습니다.

7.4.2.8 img_width

```
convert_gt_yolo.img_width
```

convert_gt_yolo.py 파일의 64 번째 라인에서 정의되었습니다.

7.4.2.9 left

```
convert_gt_yolo.left
```

convert_gt_yolo.py 파일의 84 번째 라인에서 정의되었습니다.

7.4.2.10 obj_id

`convert_gt_yolo.obj_id`

split a line by spaces.

"c" stands for center and "n" stands for normalized

`convert_gt_yolo.py` 파일의 82 번째 라인에서 정의되었습니다.

7.4.2.11 obj_list

`list convert_gt_yolo.obj_list = f.readlines()`

remove whitespace characters like `\n` at the end of each line

`convert_gt_yolo.py` 파일의 28 번째 라인에서 정의되었습니다.

7.4.2.12 obj_name

`convert_gt_yolo.obj_name = obj_list\[int\(obj_id\)\]`

`convert_gt_yolo.py` 파일의 83 번째 라인에서 정의되었습니다.

7.4.2.13 parent_path

`convert_gt_yolo.parent_path = os.path.abspath(os.path.join(os.getcwd(), os.pardir))`

e.g.

first object in the list `print(obj_list[0])`

`convert_gt_yolo.py` 파일의 35 번째 라인에서 정의되었습니다.

7.4.2.14 right

`convert_gt_yolo.right`

`convert_gt_yolo.py` 파일의 84 번째 라인에서 정의되었습니다.

7.4.2.15 top

```
convert_gt_yolo.top
```

convert_gt_yolo.py 파일의 84 번째 라인에서 정의되었습니다.

7.4.2.16 txt_list

```
convert_gt_yolo.txt_list = glob.glob('*.txt')
```

create the backup dir if it doesn't exist already

convert_gt_yolo.py 파일의 47 번째 라인에서 정의되었습니다.

7.4.2.17 width_n

```
convert_gt_yolo.width_n
```

convert_gt_yolo.py 파일의 82 번째 라인에서 정의되었습니다.

7.4.2.18 x_c_n

```
convert_gt_yolo.x_c_n
```

convert_gt_yolo.py 파일의 82 번째 라인에서 정의되었습니다.

7.4.2.19 y_c_n

```
convert_gt_yolo.y_c_n
```

convert_gt_yolo.py 파일의 82 번째 라인에서 정의되었습니다.

7.5 convert_keras 네임스페이스 참조

7.5.1 상세한 설명

-yolo3

ABOUT THIS SCRIPT:

Converts ground-truth from the annotation files
according to the <https://github.com/qqwweee/keras-yolo3>
or <https://github.com/gustavoaliati/keras-yolo3> format.

And converts the detection-results from the annotation files
according to the <https://github.com/gustavoaliati/keras-yolo3> format.

7.6 convert_keras-yolo3 네임스페이스 참조

변수

- `annotation_version` = `datetime.datetime.now().strftime('%Y%m%d%H%M%S')`
- `ap` = `argparse.ArgumentParser()`
- `required`
- `default`
- `type`
- `help`
- `action`
- `group` = `ap.add_mutually_exclusive_group(required=True)`
- `ARGS` = `ap.parse_args()`
- `class_map` = `class_file.readlines()`
- `annotation_file` = `ARGS.gt if ARGS.gt else ARGS.dr`
- `output_path`
- `exist_ok`
- `annot` = `annot.split(' ')`
- `img_path` = `annot[0].strip()`
- `annotation_dir_name` = `os.path.dirname(img_path)`
- `destination_dir` = `os.path.join(ARGS.output_path, annotation_dir_name)`
- `file_name` = `os.path.basename(img_path).replace('.jpg', '.txt')`
- `output_file_path` = `os.path.join(destination_dir, file_name)`
- `x_min`
- `y_min`
- `x_max`
- `y_max`
- `class_id`
- `string out_box`
- `score`

7.6.1 변수 문서화

7.6.1.1 action

`convert_keras-yolo3.action`

`convert_keras-yolo3.py` 파일의 31 번째 라인에서 정의되었습니다.

7.6.1.2 annot

`convert_keras-yolo3.annot` = `annot.split(' ')`

`convert_keras-yolo3.py` 파일의 55 번째 라인에서 정의되었습니다.

7.6.1.3 annotation_dir_name

```
convert_keras-yolo3.annotation_dir_name = os.path.dirname(img_path)
```

convert_keras-yolo3.py 파일의 58 번째 라인에서 정의되었습니다.

7.6.1.4 annotation_file

```
convert_keras-yolo3.annotation_file = ARGS.gt if ARGS.gt else ARGS.dr
```

convert_keras-yolo3.py 파일의 49 번째 라인에서 정의되었습니다.

7.6.1.5 annotation_version

```
convert_keras-yolo3.annotation_version = datetime.datetime.now().strftime('%Y%m%d%H%M%S')
```

convert_keras-yolo3.py 파일의 19 번째 라인에서 정의되었습니다.

7.6.1.6 ap

```
convert_keras-yolo3.ap = argparse.ArgumentParser()
```

convert_keras-yolo3.py 파일의 21 번째 라인에서 정의되었습니다.

7.6.1.7 ARGS

```
convert_keras-yolo3.ARGS = ap.parse_args()
```

convert_keras-yolo3.py 파일의 44 번째 라인에서 정의되었습니다.

7.6.1.8 class_id

```
convert_keras-yolo3.class_id
```

convert_keras-yolo3.py 파일의 78 번째 라인에서 정의되었습니다.

7.6.1.9 class_map

```
convert_keras-yolo3.class_map = class_file.readlines()
```

convert_keras-yolo3.py 파일의 47 번째 라인에서 정의되었습니다.

7.6.1.10 default

```
convert_keras-yolo3.default
```

convert_keras-yolo3.py 파일의 25 번째 라인에서 정의되었습니다.

7.6.1.11 destination_dir

```
convert_keras-yolo3.destination_dir = os.path.join(ARGS.output_path, annotation\_dir\_name)
```

convert_keras-yolo3.py 파일의 62 번째 라인에서 정의되었습니다.

7.6.1.12 exist_ok

```
convert_keras-yolo3.exist_ok
```

convert_keras-yolo3.py 파일의 51 번째 라인에서 정의되었습니다.

7.6.1.13 file_name

```
convert_keras-yolo3.file_name = os.path.basename(img\_path).replace('.jpg', '.txt')
```

convert_keras-yolo3.py 파일의 65 번째 라인에서 정의되었습니다.

7.6.1.14 group

```
convert_keras-yolo3.group = ap.add_mutually_exclusive_group(required=True)
```

convert_keras-yolo3.py 파일의 34 번째 라인에서 정의되었습니다.

7.6.1.15 help

convert_keras-yolo3.help

convert_keras-yolo3.py 파일의 27 번째 라인에서 정의되었습니다.

7.6.1.16 img_path

convert_keras-yolo3.img_path = [annot](#)[0].strip()

convert_keras-yolo3.py 파일의 56 번째 라인에서 정의되었습니다.

7.6.1.17 out_box

string convert_keras-yolo3.out_box

초기값:

```

1 = ' {} {} {} {} {} {} '.format(
2     class_map[int(class_id)].strip(), x_min, y_min, x_max, y_max)

```

convert_keras-yolo3.py 파일의 79 번째 라인에서 정의되었습니다.

7.6.1.18 output_file_path

convert_keras-yolo3.output_file_path = os.path.join([destination_dir](#), [file_name](#))

convert_keras-yolo3.py 파일의 66 번째 라인에서 정의되었습니다.

7.6.1.19 output_path

convert_keras-yolo3.output_path

convert_keras-yolo3.py 파일의 51 번째 라인에서 정의되었습니다.

7.6.1.20 required

convert_keras-yolo3.required

convert_keras-yolo3.py 파일의 24 번째 라인에서 정의되었습니다.

7.6.1.21 score

`convert_keras-yolo3.score`

`convert_keras-yolo3.py` 파일의 84 번째 라인에서 정의되었습니다.

7.6.1.22 type

`convert_keras-yolo3.type`

`convert_keras-yolo3.py` 파일의 26 번째 라인에서 정의되었습니다.

7.6.1.23 x_max

`convert_keras-yolo3.x_max`

`convert_keras-yolo3.py` 파일의 78 번째 라인에서 정의되었습니다.

7.6.1.24 x_min

`convert_keras-yolo3.x_min`

`convert_keras-yolo3.py` 파일의 78 번째 라인에서 정의되었습니다.

7.6.1.25 y_max

`convert_keras-yolo3.y_max`

`convert_keras-yolo3.py` 파일의 78 번째 라인에서 정의되었습니다.

7.6.1.26 y_min

`convert_keras-yolo3.y_min`

`convert_keras-yolo3.py` 파일의 78 번째 라인에서 정의되었습니다.

7.7 data 네임스페이스 참조

네임스페이스

- [dataset](#)
- [util](#)
- [voc_dataset](#)

7.8 data.dataset 네임스페이스 참조

클래스

- class [Dataset](#)
- class [TestDataset](#)
- class [Transform](#)
이미지 크기 변경 및 B.B 크기 변경

함수

- def [AddGaussianNoise](#) (im, noise_val)
- def [ChangeLuminance](#) (im, contrast_range, brightness_range)
- def [blur](#) (im)
- def [inverse_normalize](#) (img)
- def [pytorch_normalize](#) (img)
- def [caffe_normalize](#) (img)
- def [preprocess](#) (img, min_size=600, max_size=1000)
size가 다른 이미지에 대해서 resize 및 normalization

7.8.1 함수 문서화

7.8.1.1 AddGaussianNoise()

```
def data.dataset.AddGaussianNoise (
    im,
    noise_val )
```

dataset.py 파일의 14 번째 라인에서 정의되었습니다.

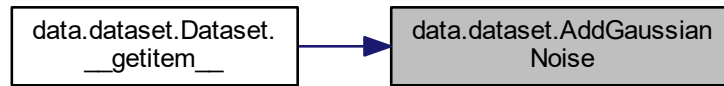
```
14 def AddGaussianNoise(im, noise_val):
15
16     img = im.astype(np.float64) / 255.0
17     _, row, col = im.shape
18     mean = 0
19     var = random.uniform(0, noise_val)
20
21     sigma = var ** 0.5
22     gaussian = np.random.normal(mean, sigma, (row, col, 1))
23     gaussian = np.concatenate((gaussian, gaussian, gaussian), axis=2)
24     gaussian = np.reshape(gaussian, (3, row, col))
25     gaussian_img = cv2.addWeighted(img, 0.9, gaussian, 0.1, 0) * 255.0
26
```

```

27     gaussian_img = np.clip(gaussian_img, 0, 255)
28     return gaussian_img.astype(np.uint8)
29
30

```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



7.8.1.2 blur()

```

def data.dataset.blur (
    im )

```

dataset.py 파일의 41 번째 라인에서 정의되었습니다.

```

41 def blur(im):
42     seq = iaa.Sequential([
43         iaa.GaussianBlur((0.0, 2.5)),
44         iaa.AverageBlur(k=(2, 7)),
45     ])
46     seq_det = seq.to_deterministic()
47     image_aug = seq_det.augment_images([im])[0]
48
49     return image_aug
50
51

```

7.8.1.3 caffe_normalize()

```

def data.dataset.caffe_normalize (
    img )

```

```

return appr -125-125 BGR

```

dataset.py 파일의 71 번째 라인에서 정의되었습니다.

```

71 def caffe_normalize(img):
72     """
73     return appr -125-125 BGR
74     """
75     img = img[[2, 1, 0], :, :] # RGB-BGR
76     img = img * 255
77     mean = np.array([122.7717, 115.9465, 102.9801]).reshape(3, 1, 1)
78     img = (img - mean).astype(np.float32, copy=True)
79     return img
80

```

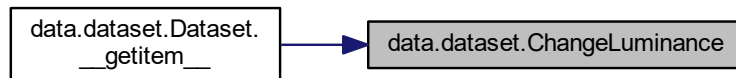
7.8.1.4 ChangeLuminance()

```
def data.dataset.ChangeLuminance (
    im,
    contrast_range,
    brightness_range )
```

dataset.py 파일의 31 번째 라인에서 정의되었습니다.

```
31 def ChangeLuminance(im, contrast_range, brightness_range):
32
33     contrast_val = randint(0, contrast_range*2) - contrast_range
34     brightness_val = randint(0, brightness_range*2) - brightness_range
35     img = np.int16(im)
36     img = img * (100+contrast_val)/100 + brightness_val
37     img = np.clip(img, 0, 255)
38     img = np.uint8(img)
39     return img
40
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



7.8.1.5 inverse_normalize()

```
def data.dataset.inverse_normalize (
    img )
```

dataset.py 파일의 52 번째 라인에서 정의되었습니다.

```
52 def inverse_normalize(img):
53     if opt.caffe_pretrain:
54         img = img + (np.array([122.7717, 115.9465, 102.9801]).reshape(3, 1, 1))
55         return img[::-1, :, :]
56     # approximate un-normalize for visualize
57     return (img * 0.225 + 0.45).clip(min=0, max=1) * 255
58
59
```

7.8.1.6 preprocess()

```
def data.dataset.preprocess (
    img,
    min_size = 600,
    max_size = 1000 )
```

size가 다른 이미지에 대해서 resize 및 normalization

매개변수

img	img (~numpy.ndarray): An image. This is in CHW and RGB format.
min_size	이미지 최소 사이즈
max_size	이미지 최대 사이즈

반환값

normalize(img) : return A preprocessed image.

Preprocess an image for feature extraction.

The length of the shorter edge is scaled to :obj:'self.min_size'.
After the scaling, if the length of the longer edge is longer than
:param min_size:
:obj:'self.max_size', the image is scaled to fit the longer edge
to :obj:'self.max_size'.

After resizing the image, the image is subtracted by a mean image value
:obj:'self.mean'.

Args:

img (~numpy.ndarray): An image. This is in CHW and RGB format.
The range of its value is :math:'[0, 255]'.
(~numpy.ndarray): An image. This is in CHW and RGB format.
The range of its value is :math:'[0, 255]'.

Returns:

~numpy.ndarray:
A preprocessed image.

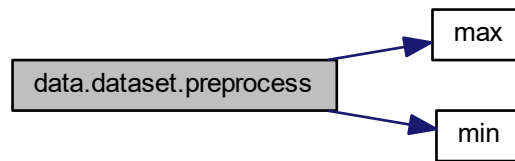
dataset.py 파일의 88 번째 라인에서 정의되었습니다.

```

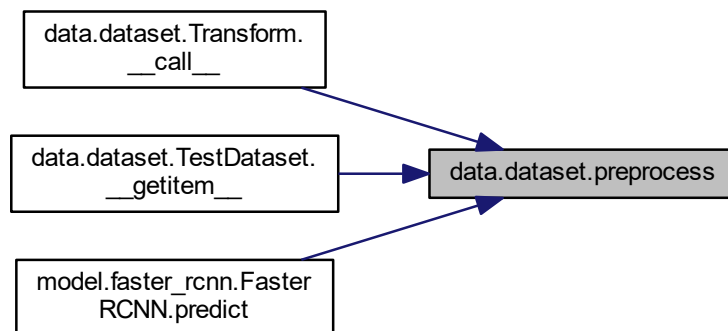
88 def preprocess(img, min_size=600, max_size=1000):
89
90     """Preprocess an image for feature extraction.
91
92     The length of the shorter edge is scaled to :obj:'self.min_size'.
93     After the scaling, if the length of the longer edge is longer than
94     :param min_size:
95     :obj:'self.max_size', the image is scaled to fit the longer edge
96     to :obj:'self.max_size'.
97
98     After resizing the image, the image is subtracted by a mean image value
99     :obj:'self.mean'.
100
101     Args:
102         img ( numpy.ndarray): An image. This is in CHW and RGB format.
103             The range of its value is :math:'[0, 255]'.
104         ( numpy.ndarray): An image. This is in CHW and RGB format.
105             The range of its value is :math:'[0, 255]'.
106
107     Returns:
108         numpy.ndarray:
109         A preprocessed image.
110
111     """
112     C, H, W = img.shape
113     scale1 = min_size / min(H, W)
114     scale2 = max_size / max(H, W)
115     scale = min(scale1, scale2)
116     img = img / 255.
117     img = sktsf.resize(img, (C, H * scale, W * scale), mode='reflect')
118     # both the longer and shorter should be less than
119     # max_size and min_size
120     if opt.caffe_pretrain:
121         normalize = caffe_normalize
122     else:
123         normalize = pytorch_normalize
124     return normalize(img)
125

```


이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



이 함수를 호출하는 함수들에 대한 그래프입니다.:



7.8.1.7 pytorch_normalize()

```
def data.dataset.pytorch_normalize (
    img )
```

```
https://github.com/pytorch/vision/issues/223
return appr -1~1 RGB
```

dataset.py 파일의 60 번째 라인에서 정의되었습니다.

```
60 def pytorch_normalize(img):
61     """
62     https://github.com/pytorch/vision/issues/223
63     return appr -1 1 RGB
64     """
65     normalize = tvtsf.Normalize(mean=[0.485, 0.456, 0.406],
66                                std=[0.229, 0.224, 0.225])
67     img = normalize(t.from_numpy(img))
68     return img.numpy()
69
70
```

7.9 data.util 네임스페이스 참조

함수

- def `read_image` (path, dtype=np.float32, color=True)
- def `resize_bbox` (bbox, in_size, out_size)
- def `flip_bbox` (bbox, size, y_flip=False, x_flip=False)
- def `crop_bbox` (bbox, y_slice=None, x_slice=None, allow_outside_center=True, return_param=False)
- def `_slice_to_bounds` (slice_)
- def `translate_bbox` (bbox, y_offset=0, x_offset=0)
- def `random_flip` (img, y_random=False, x_random=False, return_param=False, copy=False)

7.9.1 함수 문서화

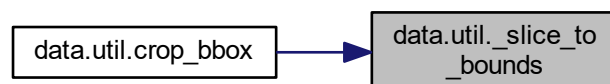
7.9.1.1 _slice_to_bounds()

```
def data.util._slice_to_bounds (
    slice_ ) [private]
```

util.py 파일의 189 번째 라인에서 정의되었습니다.

```
189 def _slice_to_bounds(slice_):
190     if slice_ is None:
191         return 0, np.inf
192
193     if slice_.start is None:
194         l = 0
195     else:
196         l = slice_.start
197
198     if slice_.stop is None:
199         u = np.inf
200     else:
201         u = slice_.stop
202
203     return l, u
204
205
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



7.9.1.2 crop_bbox()

```
def data.util.crop_bbox (
    bbox,
    y_slice = None,
    x_slice = None,
    allow_outside_center = True,
    return_param = False )
```

Translate bounding boxes to fit within the cropped area of an image.

This method is mainly used together with image cropping.
 This method translates the coordinates of bounding boxes like
`:func:'data.util.translate_bbox'`. In addition,
 this function truncates the bounding boxes to fit within the cropped area.
 If a bounding box does not overlap with the cropped area,
 this bounding box will be removed.

The bounding boxes are expected to be packed into a two dimensional
 tensor of shape `:math: '(R, 4) '`, where `:math: 'R'` is the number of
 bounding boxes in the image. The second axis represents attributes of
 the bounding box. They are `:math: '(y_{min}, x_{min}, y_{max}, x_{max}) '`,
 where the four attributes are coordinates of the top left and the
 bottom right vertices.

Args:

`bbox (~numpy.ndarray)`: Bounding boxes to be transformed. The shape is
`:math: '(R, 4) '`. `:math: 'R'` is the number of bounding boxes.
`y_slice (slice)`: The slice of y axis.
`x_slice (slice)`: The slice of x axis.
`allow_outside_center (bool)`: If this argument is `:obj: 'False'` ,
 bounding boxes whose centers are outside of the cropped area
 are removed. The default value is `:obj: 'True'` .
`return_param (bool)`: If `:obj: 'True'` , this function returns
 indices of kept bounding boxes.

Returns:

`~numpy.ndarray` or `(~numpy.ndarray, dict)`:

If `:obj: 'return_param = False'` , returns an array `:obj: 'bbox'` .

If `:obj: 'return_param = True'` ,
 returns a tuple whose elements are `:obj: 'bbox, param'` .
`:obj: 'param'` is a dictionary of intermediate parameters whose
 contents are listed below with key, value-type and the description
 of the value.

* ****index**** (`*numpy.ndarray*`): An array holding indices of used \
 bounding boxes.

util.py 파일의 117 번째 라인에서 정의되었습니다.

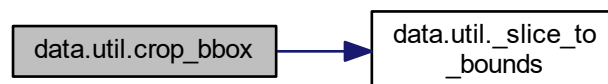
```
117 def crop_bbox(
118     bbox, y_slice=None, x_slice=None,
119     allow_outside_center=True, return_param=False):
120     """Translate bounding boxes to fit within the cropped area of an image.
121
122     This method is mainly used together with image cropping.
123     This method translates the coordinates of bounding boxes like
124     :func:'data.util.translate_bbox'. In addition,
125     this function truncates the bounding boxes to fit within the cropped area.
126     If a bounding box does not overlap with the cropped area,
127     this bounding box will be removed.
128
129     The bounding boxes are expected to be packed into a two dimensional
130     tensor of shape :math: '(R, 4) ', where :math: 'R' is the number of
131     bounding boxes in the image. The second axis represents attributes of
132     the bounding box. They are :math: '(y_{min}, x_{min}, y_{max}, x_{max}) ',
133     where the four attributes are coordinates of the top left and the
134     bottom right vertices.
135
136     Args:
137         bbox ( numpy.ndarray): Bounding boxes to be transformed. The shape is
```

```

138         :math:`(R, 4)`'. :math:`R` is the number of bounding boxes.
139     y_slice (slice): The slice of y axis.
140     x_slice (slice): The slice of x axis.
141     allow_outside_center (bool): If this argument is :obj:`False`,
142         bounding boxes whose centers are outside of the cropped area
143         are removed. The default value is :obj:`True`.
144     return_param (bool): If :obj:`True`, this function returns
145         indices of kept bounding boxes.
146
147     Returns:
148         numpy.ndarray or ( numpy.ndarray, dict):
149
150         If :obj:`return_param = False`, returns an array :obj:`bbox`.
151
152         If :obj:`return_param = True`,
153         returns a tuple whose elements are :obj:`bbox, param`.
154         :obj:`param` is a dictionary of intermediate parameters whose
155         contents are listed below with key, value-type and the description
156         of the value.
157
158         * **index** (*numpy.ndarray*): An array holding indices of used \
159             bounding boxes.
160
161     """
162
163     t, b = _slice_to_bounds(y_slice)
164     l, r = _slice_to_bounds(x_slice)
165     crop_bb = np.array((t, l, b, r))
166
167     if allow_outside_center:
168         mask = np.ones(bbox.shape[0], dtype=bool)
169     else:
170         center = (bbox[:, :2] + bbox[:, 2:]) / 2
171         mask = np.logical_and(crop_bb[:2] <= center, center < crop_bb[2:]) \
172             .all(axis=1)
173
174     bbox = bbox.copy()
175     bbox[:, :2] = np.maximum(bbox[:, :2], crop_bb[:2])
176     bbox[:, 2:] = np.minimum(bbox[:, 2:], crop_bb[2:])
177     bbox[:, :2] -= crop_bb[:2]
178     bbox[:, 2:] -= crop_bb[2:]
179
180     mask = np.logical_and(mask, (bbox[:, :2] < bbox[:, 2:]).all(axis=1))
181     bbox = bbox[mask]
182
183     if return_param:
184         return bbox, {'index': np.flatnonzero(mask)}
185     else:
186         return bbox
187
188

```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



7.9.1.3 flip_bbox()

```

def data.util.flip_bbox (
    bbox,
    size,
    y_flip = False,
    x_flip = False )

```

Flip bounding boxes accordingly.

The bounding boxes are expected to be packed into a two dimensional tensor of shape $(R, 4)$, where R is the number of bounding boxes in the image. The second axis represents attributes of the bounding box. They are $(y_{\min}, x_{\min}, y_{\max}, x_{\max})$, where the four attributes are coordinates of the top left and the bottom right vertices.

Args:

`bbox` (~numpy.ndarray): An array whose shape is $(R, 4)$.
 R is the number of bounding boxes.
`size` (tuple): A tuple of length 2. The height and the width of the image before resized.
`y_flip` (bool): Flip bounding box according to a vertical flip of an image.
`x_flip` (bool): Flip bounding box according to a horizontal flip of an image.

Returns:

~numpy.ndarray:
 Bounding boxes flipped according to the given flips.

util.py 파일의 77 번째 라인에서 정의되었습니다.

```
77 def flip_bbox(bbox, size, y_flip=False, x_flip=False):
78     """Flip bounding boxes accordingly.
79
80     The bounding boxes are expected to be packed into a two dimensional
81     tensor of shape  $(R, 4)$ , where  $R$  is the number of
82     bounding boxes in the image. The second axis represents attributes of
83     the bounding box. They are  $(y_{\min}, x_{\min}, y_{\max}, x_{\max})$ ,
84     where the four attributes are coordinates of the top left and the
85     bottom right vertices.
86
87     Args:
88         bbox ( numpy.ndarray): An array whose shape is  $(R, 4)$ .
89          $R$  is the number of bounding boxes.
90         size (tuple): A tuple of length 2. The height and the width
91         of the image before resized.
92         y_flip (bool): Flip bounding box according to a vertical flip of
93         an image.
94         x_flip (bool): Flip bounding box according to a horizontal flip of
95         an image.
96
97     Returns:
98         numpy.ndarray:
99         Bounding boxes flipped according to the given flips.
100
101     """
102     H, W = size
103     bbox = bbox.copy()
104     if y_flip:
105         y_max = H - bbox[:, 0]
106         y_min = H - bbox[:, 2]
107         bbox[:, 0] = y_min
108         bbox[:, 2] = y_max
109     if x_flip:
110         x_max = W - bbox[:, 1]
111         x_min = W - bbox[:, 3]
112         bbox[:, 1] = x_min
113         bbox[:, 3] = x_max
114     return bbox
115
116
```

7.9.1.4 random_flip()

```
def data.util.random_flip (
    img,
    y_random = False,
    x_random = False,
    return_param = False,
    copy = False )
```

Randomly flip an image in vertical or horizontal direction.

Args:

`img` (~numpy.ndarray): An array that gets flipped. This is in CHW format.
`y_random` (bool): Randomly flip in vertical direction.
`x_random` (bool): Randomly flip in horizontal direction.
`return_param` (bool): Returns information of flip.
`copy` (bool): If False, a view of `:obj:'img'` will be returned.

Returns:

~numpy.ndarray or (~numpy.ndarray, dict):

If `:obj:'return_param' = False`,
returns an array `:obj:'out_img'` that is the result of flipping.

If `:obj:'return_param' = True`,
returns a tuple whose elements are `:obj:'out_img', param`.
`:obj:'param'` is a dictionary of intermediate parameters whose contents are listed below with key, value-type and the description of the value.

* **y_flip** (*bool*): Whether the image was flipped in the vertical direction or not.
* **x_flip** (*bool*): Whether the image was flipped in the horizontal direction or not.

util.py 파일의 240 번째 라인에서 정의되었습니다.

```
240 def random_flip(img, y_random=False, x_random=False,
241                 return_param=False, copy=False):
242     """Randomly flip an image in vertical or horizontal direction.
243
244     Args:
245         img ( numpy.ndarray): An array that gets flipped. This is in
246             CHW format.
247         y_random (bool): Randomly flip in vertical direction.
248         x_random (bool): Randomly flip in horizontal direction.
249         return_param (bool): Returns information of flip.
250         copy (bool): If False, a view of :obj:'img' will be returned.
251
252     Returns:
253         numpy.ndarray or ( numpy.ndarray, dict):
254
255         If :obj:'return_param' = False',
256         returns an array :obj:'out_img' that is the result of flipping.
257
258         If :obj:'return_param' = True',
259         returns a tuple whose elements are :obj:'out_img, param'.
260         :obj:'param' is a dictionary of intermediate parameters whose
261         contents are listed below with key, value-type and the description
262         of the value.
263
264         * y_flip (*bool*): Whether the image was flipped in the\
265           vertical direction or not.
266         * x_flip (*bool*): Whether the image was flipped in the\
267           horizontal direction or not.
268
269     """
270     y_flip, x_flip = False, False
271     if y_random:
272         y_flip = random.choice([True, False])
273     if x_random:
274         x_flip = random.choice([True, False])
275
276     if y_flip:
277         img = img[:, ::-1, :]
278     if x_flip:
279         img = img[:, :, ::-1]
280
281     if copy:
282         img = img.copy()
283
284     if return_param:
285         return img, {'y_flip': y_flip, 'x_flip': x_flip}
286     else:
287         return img
```

7.9.1.5 read_image()

```
def data.util.read_image (
    path,
    dtype = np.float32,
    color = True )
```

Read an image from a file.

This function reads an image from given file. The image is CHW format and the range of its value is $[0, 255]$. If `:obj:'color = True'`, the order of the channels is RGB.

Args:

`path (str)`: A path of image file.
`dtype`: The type of array. The default value is `:obj:'~numpy.float32'`.
`color (bool)`: This option determines the number of channels.
 If `:obj:'True'`, the number of channels is three. In this case, the order of the channels is RGB. This is the default behaviour.
 If `:obj:'False'`, this function returns a grayscale image.

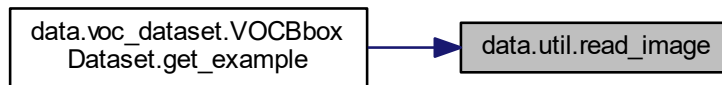
Returns:

`~numpy.ndarray`: An image.

util.py 파일의 6 번째 라인에서 정의되었습니다.

```
6 def read_image(path, dtype=np.float32, color=True):
7     """Read an image from a file.
8
9     This function reads an image from given file. The image is CHW format and
10    the range of its value is  $[0, 255]$ . If :obj:'color = True', the
11    order of the channels is RGB.
12
13    Args:
14        path (str): A path of image file.
15        dtype: The type of array. The default value is :obj:' numpy.float32'.
16        color (bool): This option determines the number of channels.
17            If :obj:'True', the number of channels is three. In this case,
18            the order of the channels is RGB. This is the default behaviour.
19            If :obj:'False', this function returns a grayscale image.
20
21    Returns:
22        numpy.ndarray: An image.
23    """
24
25    f = Image.open(path)
26    try:
27        if color:
28            img = f.convert('RGB')
29        else:
30            img = f.convert('P')
31        img = np.asarray(img, dtype=dtype)
32    finally:
33        if hasattr(f, 'close'):
34            f.close()
35
36    if img.ndim == 2:
37        # reshape (H, W) -> (1, H, W)
38        return img[np.newaxis]
39    else:
40        # transpose (H, W, C) -> (C, H, W)
41        return img.transpose((2, 0, 1))
42
43
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



7.9.1.6 resize_bbox()

```
def data.util.resize_bbox (
    bbox,
    in_size,
    out_size )
```

Resize bounding boxes according to image resize.

The bounding boxes are expected to be packed into a two dimensional tensor of shape $(R, 4)$, where R is the number of bounding boxes in the image. The second axis represents attributes of the bounding box. They are $(y_{\min}, x_{\min}, y_{\max}, x_{\max})$, where the four attributes are coordinates of the top left and the bottom right vertices.

Args:

`bbox` (~numpy.ndarray): An array whose shape is $(R, 4)$.
 R is the number of bounding boxes.
`in_size` (tuple): A tuple of length 2. The height and the width of the image before resized.
`out_size` (tuple): A tuple of length 2. The height and the width of the image after resized.

Returns:

~numpy.ndarray:
 Bounding boxes rescaled according to the given image shapes.

util.py 파일의 44 번째 라인에서 정의되었습니다.

```
44 def resize_bbox(bbox, in_size, out_size):
45     """Resize bounding boxes according to image resize.
46
47     The bounding boxes are expected to be packed into a two dimensional
48     tensor of shape  $(R, 4)$ , where  $R$  is the number of
49     bounding boxes in the image. The second axis represents attributes of
50     the bounding box. They are  $(y_{\min}, x_{\min}, y_{\max}, x_{\max})$ ,
51     where the four attributes are coordinates of the top left and the
52     bottom right vertices.
53
54     Args:
55         bbox ( numpy.ndarray): An array whose shape is  $(R, 4)$ .
56          $R$  is the number of bounding boxes.
57         in_size (tuple): A tuple of length 2. The height and the width
58         of the image before resized.
59         out_size (tuple): A tuple of length 2. The height and the width
60         of the image after resized.
61
62     Returns:
63         numpy.ndarray:
64         Bounding boxes rescaled according to the given image shapes.
```



```

65
66     """
67     bbox = bbox.copy()
68     y_scale = float(out_size[0]) / in_size[0]
69     x_scale = float(out_size[1]) / in_size[1]
70     bbox[:, 0] = y_scale * bbox[:, 0]
71     bbox[:, 2] = y_scale * bbox[:, 2]
72     bbox[:, 1] = x_scale * bbox[:, 1]
73     bbox[:, 3] = x_scale * bbox[:, 3]
74     return bbox
75
76

```

7.9.1.7 translate_bbox()

```

def data.util.translate_bbox (
    bbox,
    y_offset = 0,
    x_offset = 0 )

```

Translate bounding boxes.

This method is mainly used together with image transforms, such as padding and cropping, which translates the left top point of the image from coordinate $(0, 0)$ to coordinate $(y, x) = (y_{\text{offset}}, x_{\text{offset}})$.

The bounding boxes are expected to be packed into a two dimensional tensor of shape $(R, 4)$, where R is the number of bounding boxes in the image. The second axis represents attributes of the bounding box. They are $(y_{\text{min}}, x_{\text{min}}, y_{\text{max}}, x_{\text{max}})$, where the four attributes are coordinates of the top left and the bottom right vertices.

Args:

bbox (~numpy.ndarray): Bounding boxes to be transformed. The shape is $(R, 4)$. R is the number of bounding boxes.
y_offset (int or float): The offset along y axis.
x_offset (int or float): The offset along x axis.

Returns:

~numpy.ndarray:
Bounding boxes translated according to the given offsets.

util.py 파일의 206 번째 라인에서 정의되었습니다.

```

206 def translate_bbox(bbox, y_offset=0, x_offset=0):
207     """Translate bounding boxes.
208
209     This method is mainly used together with image transforms, such as padding
210     and cropping, which translates the left top point of the image from
211     coordinate  $(0, 0)$  to coordinate
212      $(y, x) = (y_{\text{offset}}, x_{\text{offset}})$ .
213
214     The bounding boxes are expected to be packed into a two dimensional
215     tensor of shape  $(R, 4)$ , where  $R$  is the number of
216     bounding boxes in the image. The second axis represents attributes of
217     the bounding box. They are  $(y_{\text{min}}, x_{\text{min}}, y_{\text{max}}, x_{\text{max}})$ ,
218     where the four attributes are coordinates of the top left and the
219     bottom right vertices.
220
221     Args:
222         bbox ( numpy.ndarray): Bounding boxes to be transformed. The shape is
223              $(R, 4)$ .  $R$  is the number of bounding boxes.
224         y_offset (int or float): The offset along y axis.
225         x_offset (int or float): The offset along x axis.
226
227     Returns:
228         numpy.ndarray:
229         Bounding boxes translated according to the given offsets.
230

```

```

231     """
232
233     out_bbox = bbox.copy()
234     out_bbox[:, :2] += (y_offset, x_offset)
235     out_bbox[:, 2:] += (y_offset, x_offset)
236
237     return out_bbox
238
239

```

7.10 data.voc_dataset 네임스페이스 참조

클래스

- class [VOCBboxDataset](#)

변수

- tuple [VOC_BBOX_LABEL_NAMES](#)

7.10.1 변수 문서화

7.10.1.1 VOC_BBOX_LABEL_NAMES

tuple data.voc_dataset.VOC_BBOX_LABEL_NAMES

초기값:

```

1 = (
2     'do_not_wash',
3     'do_not_bleach',
4     'do_not_iron',
5     'do_not_dry_clean'
6 )

```

voc_dataset.py 파일의 142 번째 라인에서 정의되었습니다.

7.11 demo 네임스페이스 참조

변수

- tuple [VOC_BBOX_LABEL_NAMES](#)
- [img](#) = read_image('C:/Users/user/Desktop/1.jpg')
- [anno](#)
- [bbox](#)
- [label](#)
- [difficult](#)
- [bndbox_anno](#)
- [name](#)
- [faster_rcnn](#)
- [trainer](#)
- [caffe_pretrain](#)
- [ori_img_](#)
- [_bboxes](#)
- [_labels](#)
- [_scores](#)
- [visualize](#)
- [output_img](#)

7.11.1 변수 문서화

7.11.1.1 `_bboxes`

`demo._bboxes` [private]

`demo.py` 파일의 62 번째 라인에서 정의되었습니다.

7.11.1.2 `_labels`

`demo._labels` [private]

`demo.py` 파일의 62 번째 라인에서 정의되었습니다.

7.11.1.3 `_scores`

`demo._scores` [private]

`demo.py` 파일의 62 번째 라인에서 정의되었습니다.

7.11.1.4 `anno`

`demo.anno`

초기값:

```
1 = ET.parse(  
2     # os.path.join(self.data_dir, 'Annotations', id_ + '.xml'))
```

`demo.py` 파일의 28 번째 라인에서 정의되었습니다.

7.11.1.5 `bbox`

`demo.bbox`

`demo.py` 파일의 31 번째 라인에서 정의되었습니다.

7.11.1.6 bndbox_anno

demo.bndbox_anno

demo.py 파일의 39 번째 라인에서 정의되었습니다.

7.11.1.7 caffe_pretrain

demo.caffe_pretrain

demo.py 파일의 55 번째 라인에서 정의되었습니다.

7.11.1.8 difficult

demo.difficult

demo.py 파일의 33 번째 라인에서 정의되었습니다.

7.11.1.9 faster_rcnn

demo.faster_rcnn

demo.py 파일의 51 번째 라인에서 정의되었습니다.

7.11.1.10 img

demo.img = read_image('C:/Users/user/Desktop/1.jpg')

demo.py 파일의 24 번째 라인에서 정의되었습니다.

7.11.1.11 label

demo.label

demo.py 파일의 32 번째 라인에서 정의되었습니다.

7.11.1.12 name

demo.name

demo.py 파일의 44 번째 라인에서 정의되었습니다.

7.11.1.13 ori_img_

demo.ori_img_

demo.py 파일의 56 번째 라인에서 정의되었습니다.

7.11.1.14 output_img

demo.output_img

demo.py 파일의 63 번째 라인에서 정의되었습니다.

7.11.1.15 trainer

demo.trainer

demo.py 파일의 52 번째 라인에서 정의되었습니다.

7.11.1.16 visualize

demo.visualize

demo.py 파일의 62 번째 라인에서 정의되었습니다.

7.11.1.17 VOC_BBOX_LABEL_NAMES

tuple demo.VOC_BBOX_LABEL_NAMES

초기값:

```
1 = (  
2     'do_not_wash',  
3     'do_not_bleach',  
4     'do_not_iron',  
5     'do_not_dry_clean'  
6 )
```

demo.py 파일의 14 번째 라인에서 정의되었습니다.

7.12 Evaluation 네임스페이스 참조

함수

- `def eval (dataloader, faster_rcnn, test_num=10000)`

변수

- `faster_rcnn`
- `testset`
- `test_dataloader`
- `int batch_size = 1,`
- `int num_workers = 8,`
- `bool shuffle = False,`
- `bool pin_memory = True)`
- `def eval_result = eval(test_dataloader, faster_rcnn, test_num=1000)`

7.12.1 함수 문서화

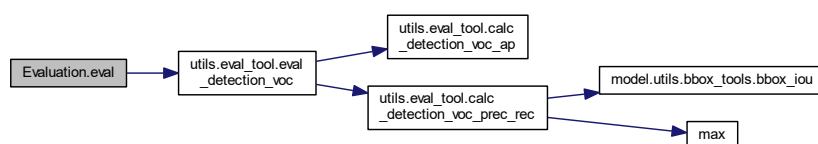
7.12.1.1 eval()

```
def Evaluation.eval (
    dataloader,
    faster_rcnn,
    test_num = 10000 )
```

Evaluation.py 파일의 20 번째 라인에서 정의되었습니다.

```
20 def eval(dataloader, faster_rcnn, test_num=10000):
21     pred_bboxes, pred_labels, pred_scores = list(), list(), list()
22     gt_bboxes, gt_labels, gt_difficults = list(), list(), list()
23     for ii, (imgs, sizes, gt_bboxes_, gt_labels_, gt_difficults_) in tqdm(enumerate(dataloader)):
24         sizes = [sizes[0][0], sizes[1][0]]
25         pred_bboxes_, pred_labels_, pred_scores_ = faster_rcnn.predict(imgs, [sizes])
26         gt_bboxes += list(gt_bboxes_.numpy())
27         gt_labels += list(gt_labels_.numpy())
28         gt_difficults += list(gt_difficults_.numpy())
29         pred_bboxes += pred_bboxes_
30         pred_labels += pred_labels_
31         pred_scores += pred_scores_
32         if ii == test_num: break
33
34     result = eval_detection_voc(
35         pred_bboxes, pred_labels, pred_scores,
36         gt_bboxes, gt_labels, gt_difficults,
37         use_07_metric=True)
38     return result
39
```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



7.12.2 변수 문서화

7.12.2.1 batch_size

```
int Evaluation.batch_size = 1,
```

Evaluation.py 파일의 45 번째 라인에서 정의되었습니다.

7.12.2.2 eval_result

```
def Evaluation.eval_result = eval(test_dataloader, faster_rcnn, test_num=1000)
```

Evaluation.py 파일의 50 번째 라인에서 정의되었습니다.

7.12.2.3 faster_rcnn

```
Evaluation.faster_rcnn
```

Evaluation.py 파일의 41 번째 라인에서 정의되었습니다.

7.12.2.4 num_workers

```
int Evaluation.num_workers = 8,
```

Evaluation.py 파일의 46 번째 라인에서 정의되었습니다.

7.12.2.5 pin_memory

```
bool Evaluation.pin_memory = True)
```

Evaluation.py 파일의 48 번째 라인에서 정의되었습니다.

7.12.2.6 shuffle

```
bool Evaluation.shuffle = False,
```

Evaluation.py 파일의 47 번째 라인에서 정의되었습니다.

7.12.2.7 test_dataloader

```
Evaluation.test_dataloader
```

Evaluation.py 파일의 44 번째 라인에서 정의되었습니다.

7.12.2.8 testset

```
Evaluation.testset
```

Evaluation.py 파일의 43 번째 라인에서 정의되었습니다.

7.13 find_class 네임스페이스 참조

함수

- def [find_class](#) (class_name)

변수

- [searching_class_name](#) = sys.argv[1]
- [parent_path](#) = os.path.abspath(os.path.join(os.getcwd(), os.pardir))
- [GT_PATH](#) = os.path.join([parent_path](#), 'input', 'ground-truth')
- [DR_PATH](#) = os.path.join([parent_path](#), 'input', 'detection-results')

7.13.1 함수 문서화

7.13.1.1 find_class()

```
def find_class.find_class (
    class_name )
```

find_class.py 파일의 14 번째 라인에서 정의되었습니다.

```
14 def find_class(class_name):
15     file_list = glob.glob('*.txt')
16     file_list.sort()
17     # iterate through the text files
18     file_found = False
19     for txt_file in file_list:
20         # open txt file lines to a list
21         with open(txt_file) as f:
22             content = f.readlines()
23             # remove whitespace characters like '\n' at the end of each line
24             content = [x.strip() for x in content]
25             # go through each line of each file
26             for line in content:
27                 class_name = line.split()[0]
28                 if class_name == searching_class_name:
29                     print(" " + txt_file)
30                     file_found = True
31                     break
32     if not file_found:
33         print(" No file found with that class")
34
```

7.13.2 변수 문서화

7.13.2.1 DR_PATH

```
find_class.DR_PATH = os.path.join(parent_path, 'input', 'detection-results')
```

find_class.py 파일의 38 번째 라인에서 정의되었습니다.

7.13.2.2 GT_PATH

```
find_class.GT_PATH = os.path.join(parent_path, 'input', 'ground-truth')
```

find_class.py 파일의 37 번째 라인에서 정의되었습니다.

7.13.2.3 parent_path

```
find_class.parent_path = os.path.abspath(os.path.join(os.getcwd(), os.pardir))
```

find_class.py 파일의 35 번째 라인에서 정의되었습니다.

7.13.2.4 searching_class_name

`find_class.searching_class_name = sys.argv[1]`

`find_class.py` 파일의 12 번째 라인에서 정의되었습니다.

7.14 img_crawling 네임스페이스 참조

함수

- `def imageCrawling (keyword, dir)`
구글 크롤링
- `def img_delete ()`
크롤링한 파일 open 여부 판단
- `def img_check ()`
파일 확장자 check jpg & png 만 남겨놓음

변수

- `_create_default_https_context`

7.14.1 함수 문서화

7.14.1.1 imageCrawling()

```
def img_crawling.imageCrawling (
    keyword,
    dir )
```

구글 크롤링

매개변수

keyword	크롤링 할 단어
dir	크롤링 데이터 저장 공간 경로

`img_crawling.py` 파일의 13 번째 라인에서 정의되었습니다.

```
13 def imageCrawling(keyword, dir):
14     response = google_images_download.googleimagesdownload()
15
16     arguments = {"keywords": keyword,
17                 "limit": 200,
18                 "print_urls": True,
19                 "no_directory": True,
20                 'output_directory': dir,
21                 "chromedriver": "C:/Users/user/Downloads/chromedriver_win32/chromedriver.exe"
```

```

22         }
23     paths = response.download(arguments)
24     print(paths)
25

```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



7.14.1.2 img_check()

```
def img_crawling.img_check ( )
```

파일 확장자 check jpg & png 만 남겨놓음

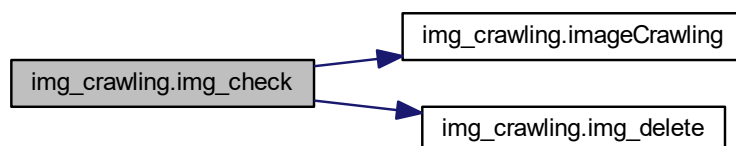
img_crawling.py 파일의 42 번째 라인에서 정의되었습니다.

```

42 def img_check():
43
44     img_dir = r"C:/Users/user/Desktop/dataset/"
45     for filename in os.listdir(img_dir):
46         fileExtension = os.path.splitext(img_dir+filename)[1]
47         if fileExtension == '.jpg' or fileExtension == '.png':
48             print("check")
49         else:
50             os.remove(img_dir + filename)
51
52

```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



7.14.1.3 img_delete()

def img_crawling.img_delete ()

크롤링한 파일 open 여부 판단

img_crawling.py 파일의 29 번째 라인에서 정의되었습니다.

```
29 def img_delete():
30     img_dir = r"C:/Users/user/Desktop/dataset"
31     for filename in os.listdir(img_dir):
32         try:
33             with Image.open(img_dir + '/' + filename) as im:
34                 print("ok")
35         except:
36             print(img_dir + "/" + filename)
37             os.remove(img_dir + "/" + filename)
38
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



7.14.2 변수 문서화

7.14.2.1 _create_default_https_context

img_crawling._create_default_https_context [private]

img_crawling.py 파일의 6 번째 라인에서 정의되었습니다.

7.15 intersect-gt-and-dr 네임스페이스 참조

함수

- def [backup](#) (src_folder, backup_files, [backup_folder](#))

변수

- [parent_path](#) = os.path.abspath(os.path.join(os.getcwd(), os.pardir))
This script ensures same number of files in ground-truth and detection-results folder.
- [GT_PATH](#) = os.path.join([parent_path](#), 'input', 'ground-truth')
- [DR_PATH](#) = os.path.join([parent_path](#), 'input', 'detection-results')
- string [backup_folder](#) = 'backup_no_matches_found'
- [gt_files](#) = glob.glob('*.txt')
- [dr_files](#) = glob.glob('*.txt')
- [gt_backup](#) = [gt_files](#) - [dr_files](#)
- [dr_backup](#) = [dr_files](#) - [gt_files](#)
- [intersection](#) = [gt_files](#) & [dr_files](#)

7.15.1 함수 문서화

7.15.1.1 backup()

```
def intersect-gt-and-dr.backup (
    src_folder,
    backup_files,
    backup_folder )
```

intersect-gt-and-dr.py 파일의 43 번째 라인에서 정의되었습니다.

```
43 def backup(src_folder, backup_files, backup_folder):
44     # non-intersection files (txt format) will be moved to a backup folder
45     if not backup_files:
46         print('No backup required for', src_folder)
47         return
48     os.chdir(src_folder)
49
50     if not os.path.exists(backup_folder):
51         os.makedirs(backup_folder)
52     for file in backup_files:
53         os.rename(file, backup_folder + '/' + file)
54
55 backup(GT_PATH, gt_backup, backup_folder)
56 backup(DR_PATH, dr_backup, backup_folder)
```

7.15.2 변수 문서화

7.15.2.1 backup_folder

```
string intersect-gt-and-dr.backup_folder = 'backup_no_matches_found'
```

intersect-gt-and-dr.py 파일의 21 번째 라인에서 정의되었습니다.

7.15.2.2 dr_backup

```
intersect-gt-and-dr.dr_backup = dr_files - gt_files
```

intersect-gt-and-dr.py 파일의 41 번째 라인에서 정의되었습니다.

7.15.2.3 dr_files

```
intersect-gt-and-dr.dr_files = glob.glob('*.txt')
```

intersect-gt-and-dr.py 파일의 29 번째 라인에서 정의되었습니다.

7.15.2.4 DR_PATH

```
intersect-gt-and-dr.DR_PATH = os.path.join(parent_path, 'input', 'detection-results')
```

intersect-gt-and-dr.py 파일의 19 번째 라인에서 정의되었습니다.

7.15.2.5 gt_backup

```
intersect-gt-and-dr.gt_backup = gt_files - dr_files
```

intersect-gt-and-dr.py 파일의 40 번째 라인에서 정의되었습니다.

7.15.2.6 gt_files

```
intersect-gt-and-dr.gt_files = glob.glob('*.txt')
```

intersect-gt-and-dr.py 파일의 24 번째 라인에서 정의되었습니다.

7.15.2.7 GT_PATH

```
intersect-gt-and-dr.GT_PATH = os.path.join(parent_path, 'input', 'ground-truth')
```

intersect-gt-and-dr.py 파일의 18 번째 라인에서 정의되었습니다.

7.15.2.8 intersection

```
intersect-gt-and-dr.intersection = gt_files & dr_files
```

intersect-gt-and-dr.py 파일의 62 번째 라인에서 정의되었습니다.

7.15.2.9 parent_path

```
intersect-gt-and-dr.parent_path = os.path.abspath(os.path.join(os.getcwd(), os.pardir))
```

This script ensures same number of files in ground-truth and detection-results folder.

When you encounter file not found error, it's usually because you have mismatched numbers of ground-truth and detection-results files. You can use this script to move ground-truth and detection-results files that are not in the intersection into a backup folder (backup_no_matches_found). This will retain only files that have the same name in both folders.

intersect-gt-and-dr.py 파일의 16 번째 라인에서 정의되었습니다.

7.16 main 네임스페이스 참조

함수

- def `log_average_miss_rate` (precision, fp_cumsum, num_images)
- def `error` (msg)
- def `is_float_between_0_and_1` (value)
- def `voc_ap` (rec, prec)
- def `file_lines_to_list` (path)
- def `draw_text_in_image` (img, text, pos, color, line_width)
- def `adjust_axes` (r, t, fig, axes)
- def `draw_plot_func` (dictionary, n_classes, window_title, plot_title, x_label, output_path, to_show, plot_color, true_p_bar)

변수

- `testset` = `TestDataset`(opt)
- `test_dataloader`
- `faster_rcnn` = `FasterRCNN`()
- `trainer` = `FasterRCNNTrainer`(`faster_rcnn`).`cuda`()
- `pred_bboxes`
- `pred_labels`
- `pred_scores`
- `gt_bboxes`
- `gt_labels`
- `gt_difficults`
- `list_sizes` = [`sizes`[0][0], `sizes`[1][0]]
- `pred_bboxes_`
- `pred_labels_`
- `pred_scores_`
- `f` = `open`('C:/Users/user/PycharmProjects/Faster-RCNN-Pytorch_6/input/detection-results/'+`test_dataloader.da`
`mode='wt', encoding='utf-8'`)
- float `MINOVERLAP` = 0.5
- `parser` = `argparse.ArgumentParser`()
- `help`
- `action`
- `nargs`
- `type`
- `str`
- `args` = `parser.parse_args`()
- `ignore`
- bool `specific_iou_flagged` = False
- `GT_PATH` = `os.path.join`(`os.getcwd`(), 'input', 'ground-truth')
- `DR_PATH` = `os.path.join`(`os.getcwd`(), 'input', 'detection-results')
- `IMG_PATH` = `os.path.join`(`os.getcwd`(), 'input', 'images-optional')
- `no_animation`
- bool `show_animation` = False
- bool `draw_plot` = False
- `no_plot`
- string `TEMP_FILES_PATH` = ".temp_files"
- string `results_files_path` = "results"
- `ground_truth_files_list` = `glob.glob`(`GT_PATH` + '/*.txt')
- dictionary `gt_counter_per_class` = {}

- dictionary `counter_images_per_class` = {}
- `file_id` = `txt_file.split(".txt", 1)[0]`
- `temp_path` = `os.path.join(DR_PATH, (file_id + ".txt"))`
- string `error_msg` = "Error. File not found: {}".format(`temp_path`)
- def `lines_list` = `file_lines_to_list(txt_file)`
- list `bounding_boxes` = []
- bool `is_difficult` = False
- list `already_seen_classes` = []
- `class_name` = `line.split()[0]`
- `left`
- `top`
- `right`
- `bottom`
- `_difficult`
- string `bbox` = `left + " " + top + " " + right + " " + bottom`
- `gt_classes` = `list(gt_counter_per_class.keys())`
- `n_classes` = `len(gt_classes)`
- `n_args` = `len(args.set_class_iou)`
- `specific_iou_classes` = `args.set_class_iou[:2]`
- `iou_list` = `args.set_class_iou[1::2]`
- `dr_files_list` = `glob.glob(DR_PATH + '/*.txt')`
- def `lines` = `file_lines_to_list(txt_file)`
- `tmp_class_name`
- `confidence`
- `key`
- `reverse`
- float `sum_AP` = 0.0
- dictionary `ap_dictionary` = {}
- dictionary `lamr_dictionary` = {}
- dictionary `count_true_positives` = {}
- string `dr_file` = `TEMP_FILES_PATH + "/" + class_name + "_dr.json"`
- `dr_data` = `json.load(open(dr_file))`
- `nd` = `len(dr_data)`
- list `tp` = `[0] * nd`
- list `fp` = `[0] * nd`
- `ground_truth_img` = `glob.glob1(IMG_PATH, file_id + ".*")`
- `img` = `cv2.imread(IMG_PATH + "/" + ground_truth_img[0], cv2.IMREAD_COLOR)`
- string `img_cumulative_path` = `results_files_path + "/images/" + ground_truth_img[0]`
- `img_cumulative` = `cv2.imread(img_cumulative_path)`
- int `bottom_border` = 60
- list `BLACK` = `[0, 0, 0]`
- string `gt_file` = `TEMP_FILES_PATH + "/" + file_id + "_ground_truth.json"`
- `ground_truth_data` = `json.load(open(gt_file))`
- int `ovmax` = -1
- int `gt_match` = -1
- list `bb` = `[float(x) for x in detection["bbox"].split()]`
- list `bbgt` = `[float(x) for x in obj["bbox"].split()]`
- list `bi` = `[max(bb[0],bbgt[0]), max(bb[1],bbgt[1]), min(bb[2],bbgt[2]), min(bb[3],bbgt[3])]`
- list `iw` = `bi[2] - bi[0] + 1`
- list `ih` = `bi[3] - bi[1] + 1`
- tuple `ua`
- list `ov` = `iw * ih / ua`
- string `status` = "NO MATCH FOUND!"
- float `min_overlap` = `MINOVERLAP`
- `index` = `specific_iou_classes.index(class_name)`

- `height`
- `width`
- tuple `white` = (255,255,255)
- tuple `light_blue` = (255,200,100)
- tuple `green` = (0,255,0)
- tuple `light_red` = (30,30,255)
- int `margin` = 10
- `v_pos` = int(`height` - `margin` - (`bottom_border` / 2.0))
- string `text` = "Image: " + `ground_truth_img`[0] + " "
- `line_width`
- tuple `color` = `light_red`
- `_`
- `rank_pos` = str(`idx`+1)
- `font` = cv2.FONT_HERSHEY_SIMPLEX
- string `output_img_path` = `results_files_path` + "/images/detections_one_by_one/" + `class_name` + "_detection" + str(`idx`) + ".jpg"
- int `cumsum` = 0
- list `rec` = `tp`[:]
- list `prec` = `tp`[:]
- `ap`
- `mrec`
- `mprec`
- list `rounded_prec` = ['%.2f' % elem for elem in `prec`]
- list `rounded_rec` = ['%.2f' % elem for elem in `rec`]
- dictionary `n_images` = `counter_images_per_class`[`class_name`]
- `lamr`
- `mr`
- `fppi`
- `area_under_curve_x` = `mrec`[:-1] + [`mrec`[-2]] + [`mrec`[-1]]
- `area_under_curve_y` = `mprec`[:-1] + [0.0] + [`mprec`[-1]]
- `alpha`
- `edgecolor`
- `fig` = plt.gcf()
- `axes` = plt.gca()
- float `mAP` = `sum_AP` / `n_classes`
- dictionary `det_counter_per_class` = {}
- `dr_classes` = list(`det_counter_per_class`.keys())
- string `window_title` = "ground-truth-info"
- string `plot_title` = "ground-truth\n"
- string `x_label` = "Number of objects per class"
- string `output_path` = `results_files_path` + "/ground-truth-info.png"
- bool `to_show` = False
- string `plot_color` = 'forestgreen'
- `count_non_zero_values_in_dictionary` = sum(int(`x`) > 0 for `x` in list(`det_counter_per_class`.values()))
- dictionary `true_p_bar` = `count_true_positives`
- dictionary `n_det` = `det_counter_per_class`[`class_name`]

7.16.1 함수 문서화

7.16.1.1 adjust_axes()

```
def main.adjust_axes (
    r,
    t,
    fig,
    axes )
```

main.py 파일의 180 번째 라인에서 정의되었습니다.

```
180 def adjust_axes(r, t, fig, axes):
181     # get text width for re-scaling
182     bb = t.get_window_extent(renderer=r)
183     text_width_inches = bb.width / fig.dpi
184     # get axis width in inches
185     current_fig_width = fig.get_figwidth()
186     new_fig_width = current_fig_width + text_width_inches
187     propotion = new_fig_width / current_fig_width
188     # get axis limit
189     x_lim = axes.get_xlim()
190     axes.set_xlim([x_lim[0], x_lim[1]*propotion])
191
192 """
193 Draw plot using Matplotlib
194 """
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:

main.draw_plot_func

main.adjust_axes

7.16.1.2 draw_plot_func()

```
def main.draw__plot_func (
    dictionary,
    n_classes,
    window_title,
    plot_title,
    x_label,
    output_path,
    to_show,
    plot_color,
    true_p_bar )
```

main.py 파일의 195 번째 라인에서 정의되었습니다.

```
195 def draw_plot_func(dictionary, n_classes, window_title, plot_title, x_label, output_path, to_show,
    plot_color, true_p_bar):
196     # sort the dictionary by decreasing value, into a list of tuples
197     sorted_dic_by_value = sorted(dictionary.items(), key=operator.itemgetter(1))
198     # unpacking the list of tuples into two lists
199     sorted_keys, sorted_values = zip(*sorted_dic_by_value)
200     #
201     if true_p_bar != "":
202         """
203         Special case to draw in:
```

```

204         - green -> TP: True Positives (object detected and matches ground-truth)
205         - red -> FP: False Positives (object detected but does not match ground-truth)
206         - orange -> FN: False Negatives (object not detected but present in the ground-truth)
207     """
208     fp_sorted = []
209     tp_sorted = []
210     for key in sorted_keys:
211         fp_sorted.append(dictionary[key] - true_p_bar[key])
212         tp_sorted.append(true_p_bar[key])
213     plt.barh(range(n_classes), fp_sorted, align='center', color='crimson', label='False Positive')
214     plt.barh(range(n_classes), tp_sorted, align='center', color='forestgreen', label='True
Positive', left=fp_sorted)
215     # add legend
216     plt.legend(loc='lower right')
217     """
218     Write number on side of bar
219     """
220     fig = plt.gcf() # gcf - get current figure
221     axes = plt.gca()
222     r = fig.canvas.get_renderer()
223     for i, val in enumerate(sorted_values):
224         fp_val = fp_sorted[i]
225         tp_val = tp_sorted[i]
226         fp_str_val = " " + str(fp_val)
227         tp_str_val = fp_str_val + " " + str(tp_val)
228         # trick to paint multicolor with offset:
229         # first paint everything and then repaint the first number
230         t = plt.text(val, i, tp_str_val, color='forestgreen', va='center', fontweight='bold')
231         plt.text(val, i, fp_str_val, color='crimson', va='center', fontweight='bold')
232         if i == (len(sorted_values)-1): # largest bar
233             adjust_axes(r, t, fig, axes)
234     else:
235         plt.barh(range(n_classes), sorted_values, color=plot_color)
236         """
237         Write number on side of bar
238         """
239         fig = plt.gcf() # gcf - get current figure
240         axes = plt.gca()
241         r = fig.canvas.get_renderer()
242         for i, val in enumerate(sorted_values):
243             str_val = " " + str(val) # add a space before
244             if val < 1.0:
245                 str_val = " {:.2f}".format(val)
246             t = plt.text(val, i, str_val, color=plot_color, va='center', fontweight='bold')
247             # re-set axes to show number inside the figure
248             if i == (len(sorted_values)-1): # largest bar
249                 adjust_axes(r, t, fig, axes)
250     # set window title
251     fig.canvas.set_window_title(window_title)
252     # write classes in y axis
253     tick_font_size = 12
254     plt.yticks(range(n_classes), sorted_keys, fontsize=tick_font_size)
255     """
256     Re-scale height accordingly
257     """
258     init_height = fig.get_figheight()
259     # comput the matrix height in points and inches
260     dpi = fig.dpi
261     height_pt = n_classes * (tick_font_size * 1.4) # 1.4 (some spacing)
262     height_in = height_pt / dpi
263     # compute the required figure height
264     top_margin = 0.15 # in percentage of the figure height
265     bottom_margin = 0.05 # in percentage of the figure height
266     figure_height = height_in / (1 - top_margin - bottom_margin)
267     # set new height
268     if figure_height > init_height:
269         fig.set_figheight(figure_height)
270
271     # set plot title
272     plt.title(plot_title, fontsize=14)
273     # set axis titles
274     # plt.xlabel('classes')
275     plt.xlabel(x_label, fontsize='large')
276     # adjust size of window
277     fig.tight_layout()
278     # save the plot
279     fig.savefig(output_path)
280     # show image
281     if to_show:
282         plt.show()
283     # close the plot
284     plt.close()
285

```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



7.16.1.3 draw_text_in_image()

```
def main.draw_text_in_image (
    img,
    text,
    pos,
    color,
    line_width )
```

main.py 파일의 163 번째 라인에서 정의되었습니다.

```
163 def draw_text_in_image(img, text, pos, color, line_width):
164     font = cv2.FONT_HERSHEY_PLAIN
165     fontScale = 1
166     lineType = 1
167     bottomLeftCornerOfText = pos
168     cv2.putText(img, text,
169                 bottomLeftCornerOfText,
170                 font,
171                 fontScale,
172                 color,
173                 lineType)
174     text_width, _ = cv2.getTextSize(text, font, fontScale, lineType)[0]
175     return img, (line_width + text_width)
176
177 """
178 Plot - adjust axes
179 """
```

7.16.1.4 error()

```
def main.error (
    msg )
```

main.py 파일의 78 번째 라인에서 정의되었습니다.

```
78 def error(msg):
79     print(msg)
80     sys.exit(0)
81
82 """
83 check if the number is a float between 0.0 and 1.0
84 """
```

7.16.1.5 file_lines_to_list()

```
def main.file_lines_to_list (
    path )
```

main.py 파일의 152 번째 라인에서 정의되었습니다.

```
152 def file_lines_to_list(path):
153     # open txt file lines to a list
154     with open(path) as f:
155         content = f.readlines()
156     # remove whitespace characters like '\n' at the end of each line
157     content = [x.strip() for x in content]
158     return content
159
160 """
161 Draws text in image
162 """
```

7.16.1.6 is_float_between_0_and_1()

```
def main.is_float_between_0_and_1 (
    value )
```

main.py 파일의 85 번째 라인에서 정의되었습니다.

```
85 def is_float_between_0_and_1(value):
86     try:
87         val = float(value)
88         if val > 0.0 and val < 1.0:
89             return True
90         else:
91             return False
92     except ValueError:
93         return False
94
95 """
96 Calculate the AP given the recall and precision array
97 1st) We compute a version of the measured precision/recall curve with
98     precision monotonically decreasing
99 2nd) We compute the AP as the area under this curve by numerical integration.
100 """
```

7.16.1.7 log_average_miss_rate()

```
def main.log_average_miss_rate (
    precision,
    fp_cumsum,
    num_images )
```

log-average miss rate:
Calculated by averaging miss rates at 9 evenly spaced FPPI points between 10e-2 and 10e0, in log-space.

output:

```
lamr | log-average miss rate
mr | miss rate
fpfi | false positives per image
```

references:

[1] Dollar, Piotr, et al. "Pedestrian Detection: An Evaluation of the State of the Art." Pattern Analysis and Machine Intelligence, IEEE Transactions on 34.4 (2012): 743 - 761.

main.py 파일의 33 번째 라인에서 정의되었습니다.

```

33 def log_average_miss_rate(precision, fp_cumsum, num_images):
34     """
35     log-average miss rate:
36     Calculated by averaging miss rates at 9 evenly spaced FPPI points
37     between 10e-2 and 10e0, in log-space.
38
39     output:
40         lamr | log-average miss rate
41         mr | miss rate
42         fppi | false positives per image
43
44     references:
45     [1] Dollar, Piotr, et al. "Pedestrian Detection: An Evaluation of the
46         State of the Art." Pattern Analysis and Machine Intelligence, IEEE
47         Transactions on 34.4 (2012): 743 - 761.
48     """
49
50     # if there were no detections of that class
51     if precision.size == 0:
52         lamr = 0
53         mr = 1
54         fppi = 0
55         return lamr, mr, fppi
56
57     fppi = fp_cumsum / float(num_images)
58     mr = (1 - precision)
59
60     fppi_tmp = np.insert(fppi, 0, -1.0)
61     mr_tmp = np.insert(mr, 0, 1.0)
62
63     # Use 9 evenly spaced reference points in log-space
64     ref = np.logspace(-2.0, 0.0, num = 9)
65     for i, ref_i in enumerate(ref):
66         # np.where() will always find at least 1 index, since min(ref) = 0.01 and min(fppi_tmp) = -1.0
67         j = np.where(fppi_tmp <= ref_i)[-1][-1]
68         ref[i] = mr_tmp[j]
69
70     # log(0) is undefined, so we use the np.maximum(1e-10, ref)
71     lamr = math.exp(np.mean(np.log(np.maximum(1e-10, ref))))
72
73     return lamr, mr, fppi
74
75 """
76 throw error and exit
77 """

```

7.16.1.8 voc_ap()

```

def main.voc_ap (
    rec,
    prec )

--- Official matlab code VOC2012---
mrec=[0 ; rec ; 1];
mpre=[0 ; prec ; 0];
for i=numel(mpre)-1:-1:1
    mpre(i)=max(mpre(i),mpre(i+1));
end
i=find(mrec(2:end)~=mrec(1:end-1))+1;
ap=sum((mrec(i)-mrec(i-1)).*mpre(i));

```

main.py 파일의 101 번째 라인에서 정의되었습니다.

```

101 def voc_ap(rec, prec):
102     """
103     --- Official matlab code VOC2012---
104     mrec=[0 ; rec ; 1];
105     mpre=[0 ; prec ; 0];
106     for i=numel(mpre)-1:-1:1
107         mpre(i)=max(mpre(i),mpre(i+1));
108     end
109     i=find(mrec(2:end) ==mrec(1:end-1))+1;
110     ap=sum((mrec(i)-mrec(i-1)).*mpre(i));

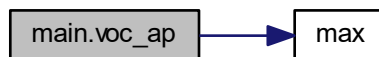
```

```

111     """
112     rec.insert(0, 0.0) # insert 0.0 at beginning of list
113     rec.append(1.0) # insert 1.0 at end of list
114     mrec = rec[:]
115     prec.insert(0, 0.0) # insert 0.0 at beginning of list
116     prec.append(0.0) # insert 0.0 at end of list
117     mpre = prec[:]
118     """
119     This part makes the precision monotonically decreasing
120     (goes from the end to the beginning)
121     matlab: for i=numel(mpre)-1:-1:1
122             mpre(i)=max(mpre(i),mpre(i+1));
123     """
124     # matlab indexes start in 1 but python in 0, so I have to do:
125     #     range(start=(len(mpre) - 2), end=0, step=-1)
126     # also the python function range excludes the end, resulting in:
127     #     range(start=(len(mpre) - 2), end=-1, step=-1)
128     for i in range(len(mpre)-2, -1, -1):
129         mpre[i] = max(mpre[i], mpre[i+1])
130     """
131     This part creates a list of indexes where the recall changes
132     matlab: i=find(mrec(2:end) ==mrec(1:end-1))+1;
133     """
134     i_list = []
135     for i in range(1, len(mrec)):
136         if mrec[i] != mrec[i-1]:
137             i_list.append(i) # if it was matlab would be i + 1
138     """
139     The Average Precision (AP) is the area under the curve
140     (numerical integration)
141     matlab: ap=sum((mrec(i)-mrec(i-1)).*mpre(i));
142     """
143     ap = 0.0
144     for i in i_list:
145         ap += (mrec[i]-mrec[i-1])*mpre[i]
146     return ap, mrec, mpre
147
148
149 """
150 Convert the lines of a file to a list
151 """

```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



7.16.2 변수 문서화

7.16.2.1 _

main._ [private]

main.py 파일의 691 번째 라인에서 정의되었습니다.

7.16.2.2 `_difficult`

`main._difficult` [private]

`main.py` 파일의 450 번째 라인에서 정의되었습니다.

7.16.2.3 `action`

`main.action`

`main.py` 파일의 340 번째 라인에서 정의되었습니다.

7.16.2.4 `alpha`

`main.alpha`

`main.py` 파일의 768 번째 라인에서 정의되었습니다.

7.16.2.5 `already_seen_classes`

`list main.already_seen_classes = []`

`main.py` 파일의 446 번째 라인에서 정의되었습니다.

7.16.2.6 `ap`

`main.ap`

`main.py` 파일의 742 번째 라인에서 정의되었습니다.

7.16.2.7 `ap_dictionary`

`dictionary main.ap_dictionary = {}`

`main.py` 파일의 567 번째 라인에서 정의되었습니다.

7.16.2.8 area_under_curve_x

```
main.area_under_curve_x = mrec[:-1] + [mrec[-2]] + [mrec[-1]]
```

main.py 파일의 766 번째 라인에서 정의되었습니다.

7.16.2.9 area_under_curve_y

```
main.area_under_curve_y = mprec[:-1] + [0.0] + [mprec[-1]]
```

main.py 파일의 767 번째 라인에서 정의되었습니다.

7.16.2.10 args

```
main.args = parser.parse_args()
```

main.py 파일의 347 번째 라인에서 정의되었습니다.

7.16.2.11 axes

```
main.axes = plt.gca()
```

main.py 파일의 779 번째 라인에서 정의되었습니다.

7.16.2.12 bb

```
list main.bb = [ float(x) for x in detection["bbox"].split() ]
```

main.py 파일의 618 번째 라인에서 정의되었습니다.

7.16.2.13 bbgt

```
list main.bbgt = [ float(x) for x in obj["bbox"].split() ]
```

main.py 파일의 622 번째 라인에서 정의되었습니다.

7.16.2.14 bbox

```
string main.bbox = left + " " + top + " " + right + " " + bottom
```

main.py 파일의 464 번째 라인에서 정의되었습니다.

7.16.2.15 bi

```
list main.bi = [max(bb[0],bbgt[0]), max(bb[1],bbgt[1]), min(bb[2],bbgt[2]), min(bb[3],bbgt[3])]
```

main.py 파일의 623 번째 라인에서 정의되었습니다.

7.16.2.16 BLACK

```
list main.BLACK = [0, 0, 0]
```

main.py 파일의 609 번째 라인에서 정의되었습니다.

7.16.2.17 bottom

```
main.bottom
```

main.py 파일의 450 번째 라인에서 정의되었습니다.

7.16.2.18 bottom_border

```
int main.bottom_border = 60
```

main.py 파일의 608 번째 라인에서 정의되었습니다.

7.16.2.19 bounding_boxes

```
list main.bounding_boxes = []
```

main.py 파일의 444 번째 라인에서 정의되었습니다.

7.16.2.20 class_name

```
main.class_name = line.split()[0]
```

main.py 파일의 450 번째 라인에서 정의되었습니다.

7.16.2.21 color

```
tuple main.color = light_red
```

main.py 파일의 685 번째 라인에서 정의되었습니다.

7.16.2.22 confidence

```
main.confidence
```

main.py 파일의 547 번째 라인에서 정의되었습니다.

7.16.2.23 count_non_zero_values_in_dictionary

```
main.count_non_zero_values_in_dictionary = sum(int(x) > 0 for x in list(det_counter_per_class.values()))
```

main.py 파일의 873 번째 라인에서 정의되었습니다.

7.16.2.24 count_true_positives

```
dictionary main.count_true_positives = {}
```

main.py 파일의 572 번째 라인에서 정의되었습니다.

7.16.2.25 counter_images_per_class

```
dictionary main.counter_images_per_class = {}
```

main.py 파일의 430 번째 라인에서 정의되었습니다.

7.16.2.26 cumsum

```
int main.cumsum = 0
```

main.py 파일의 724 번째 라인에서 정의되었습니다.

7.16.2.27 det_counter_per_class

```
dictionary main.det_counter_per_class = {}
```

main.py 파일의 806 번째 라인에서 정의되었습니다.

7.16.2.28 dr_classes

```
main.dr_classes = list(det_counter_per_class.keys())
```

main.py 파일의 822 번째 라인에서 정의되었습니다.

7.16.2.29 dr_data

```
main.dr_data = json.load(open(dr_file))
```

main.py 파일의 579 번째 라인에서 정의되었습니다.

7.16.2.30 dr_file

```
string main.dr_file = TEMP_FILES_PATH + "/" + class_name + "_dr.json"
```

main.py 파일의 578 번째 라인에서 정의되었습니다.

7.16.2.31 dr_files_list

```
main.dr_files_list = glob.glob(DR_PATH + '/*.txt')
```

main.py 파일의 527 번째 라인에서 정의되었습니다.

7.16.2.32 DR_PATH

```
main.DR_PATH = os.path.join(os.getcwd(), 'input', 'detection-results')

```

main.py 파일의 373 번째 라인에서 정의되었습니다.

7.16.2.33 draw_plot

```
bool main.draw_plot = False

```

main.py 파일의 395 번째 라인에서 정의되었습니다.

7.16.2.34 edgecolor

```
main.edgecolor

```

main.py 파일의 768 번째 라인에서 정의되었습니다.

7.16.2.35 error_msg

```
string main.error_msg = "Error. File not found: {}\n".format(temp\_path)

```

main.py 파일의 439 번째 라인에서 정의되었습니다.

7.16.2.36 f

```
main.f = open('C:/Users/user/PycharmProjects/Faster-RCNN-Pytorch_6/input/detection-results/' + test\_dataloader.dataset.db.ids[ii] + '.txt',
mode='wt', encoding='utf-8')

```

main.py 파일의 320 번째 라인에서 정의되었습니다.

7.16.2.37 faster_rcnn

```
main.faster_rcnn = FasterRCNN()

```

main.py 파일의 300 번째 라인에서 정의되었습니다.

7.16.2.38 fig

```
main.fig = plt.gcf()
```

main.py 파일의 770 번째 라인에서 정의되었습니다.

7.16.2.39 file_id

```
main.file_id = txt_file.split(".txt", 1)[0]
```

main.py 파일의 434 번째 라인에서 정의되었습니다.

7.16.2.40 font

```
main.font = cv2.FONT_HERSHEY_SIMPLEX
```

main.py 파일의 703 번째 라인에서 정의되었습니다.

7.16.2.41 fp

```
list main.fp = [0] * nd
```

main.py 파일의 586 번째 라인에서 정의되었습니다.

7.16.2.42 fppi

```
main.fppi
```

main.py 파일의 756 번째 라인에서 정의되었습니다.

7.16.2.43 green

```
tuple main.green = (0,255,0)
```

main.py 파일의 675 번째 라인에서 정의되었습니다.

7.16.2.44 ground_truth_data

```
main.ground_truth_data = json.load(open(gt_file))
```

main.py 파일의 614 번째 라인에서 정의되었습니다.

7.16.2.45 ground_truth_files_list

```
main.ground_truth_files_list = glob.glob(GT_PATH + '/*.txt')
```

main.py 파일의 424 번째 라인에서 정의되었습니다.

7.16.2.46 ground_truth_img

```
main.ground_truth_img = glob.glob1(IMG_PATH, file_id + ".*")
```

main.py 파일의 591 번째 라인에서 정의되었습니다.

7.16.2.47 gt_bboxes

```
main.gt_bboxes
```

main.py 파일의 309 번째 라인에서 정의되었습니다.

7.16.2.48 gt_classes

```
main.gt_classes = list(gt_counter_per_class.keys())
```

main.py 파일의 490 번째 라인에서 정의되었습니다.

7.16.2.49 gt_counter_per_class

```
dictionary main.gt_counter_per_class = {}
```

main.py 파일의 429 번째 라인에서 정의되었습니다.

7.16.2.50 gt_difficults

main.gt_difficults

main.py 파일의 309 번째 라인에서 정의되었습니다.

7.16.2.51 gt_file

```
string main.gt_file = TEMP_FILES_PATH + "/" + file_id + "_ground_truth.json"
```

main.py 파일의 613 번째 라인에서 정의되었습니다.

7.16.2.52 gt_labels

main.gt_labels

main.py 파일의 309 번째 라인에서 정의되었습니다.

7.16.2.53 gt_match

main.gt_match = -1

main.py 파일의 616 번째 라인에서 정의되었습니다.

7.16.2.54 GT_PATH

```
main.GT_PATH = os.path.join(os.getcwd(), 'input', 'ground-truth')
```

main.py 파일의 372 번째 라인에서 정의되었습니다.

7.16.2.55 height

main.height

main.py 파일의 671 번째 라인에서 정의되었습니다.

7.16.2.56 help

main.help

main.py 파일의 340 번째 라인에서 정의되었습니다.

7.16.2.57 ignore

main.ignore

main.py 파일의 363 번째 라인에서 정의되었습니다.

7.16.2.58 ih

```
list main.ih = bi[3] - bi[1] + 1
```

main.py 파일의 625 번째 라인에서 정의되었습니다.

7.16.2.59 img

```
main.img = cv2.imread(IMG_PATH + "/" + ground_truth_img[0], cv2.IMREAD_COLOR)
```

main.py 파일의 600 번째 라인에서 정의되었습니다.

7.16.2.60 img_cumulative

```
main.img_cumulative = cv2.imread(img_cumulative_path)
```

main.py 파일의 604 번째 라인에서 정의되었습니다.

7.16.2.61 img_cumulative_path

```
string main.img_cumulative_path = results_files_path + "/images/" + ground_truth_img[0]
```

main.py 파일의 602 번째 라인에서 정의되었습니다.

7.16.2.62 IMG_PATH

```
main.IMG_PATH = os.path.join(os.getcwd(), 'input', 'images-optional')
```

main.py 파일의 375 번째 라인에서 정의되었습니다.

7.16.2.63 index

```
main.index = specific_iou_classes.index(class_name)
```

main.py 파일의 642 번째 라인에서 정의되었습니다.

7.16.2.64 iou_list

```
main.iou_list = args.set_class_iou[1::2]
```

main.py 파일의 512 번째 라인에서 정의되었습니다.

7.16.2.65 is_difficult

```
bool main.is_difficult = False
```

main.py 파일의 445 번째 라인에서 정의되었습니다.

7.16.2.66 iw

```
list main.iw = bi[2] - bi[0] + 1
```

main.py 파일의 624 번째 라인에서 정의되었습니다.

7.16.2.67 key

```
main.key
```

main.py 파일의 559 번째 라인에서 정의되었습니다.

7.16.2.68 lamr

main.lamr

main.py 파일의 756 번째 라인에서 정의되었습니다.

7.16.2.69 lamr_dictionary

```
dictionary main.lamr_dictionary = {}
```

main.py 파일의 568 번째 라인에서 정의되었습니다.

7.16.2.70 left

main.left

main.py 파일의 450 번째 라인에서 정의되었습니다.

7.16.2.71 light_blue

```
tuple main.light_blue = (255,200,100)
```

main.py 파일의 674 번째 라인에서 정의되었습니다.

7.16.2.72 light_red

```
tuple main.light_red = (30,30,255)
```

main.py 파일의 676 번째 라인에서 정의되었습니다.

7.16.2.73 line_width

main.line_width

main.py 파일의 681 번째 라인에서 정의되었습니다.

7.16.2.74 lines

```
def main.lines = file_lines_to_list(txt_file)
```

main.py 파일의 543 번째 라인에서 정의되었습니다.

7.16.2.75 lines_list

```
def main.lines_list = file_lines_to_list(txt_file)
```

main.py 파일의 442 번째 라인에서 정의되었습니다.

7.16.2.76 mAP

```
float main.mAP = sum_AP / n_classes
```

main.py 파일의 794 번째 라인에서 정의되었습니다.

7.16.2.77 margin

```
int main.margin = 10
```

main.py 파일의 678 번째 라인에서 정의되었습니다.

7.16.2.78 min_overlap

```
main.min_overlap = MINOVERLAP
```

main.py 파일의 639 번째 라인에서 정의되었습니다.

7.16.2.79 MINOVERLAP

```
float main.MINOVERLAP = 0.5
```

main.py 파일의 337 번째 라인에서 정의되었습니다.

7.16.2.80 mprec

main.mprec

main.py 파일의 742 번째 라인에서 정의되었습니다.

7.16.2.81 mr

main.mr

main.py 파일의 756 번째 라인에서 정의되었습니다.

7.16.2.82 mrec

main.mrec

main.py 파일의 742 번째 라인에서 정의되었습니다.

7.16.2.83 n_args

main.n_args = len(args.set_class_iou)

main.py 파일의 503 번째 라인에서 정의되었습니다.

7.16.2.84 n_classes

main.n_classes = len(gt_classes)

main.py 파일의 493 번째 라인에서 정의되었습니다.

7.16.2.85 n_det

dictionary main.n_det = det_counter_per_class[class_name]

main.py 파일의 899 번째 라인에서 정의되었습니다.

7.16.2.86 n_images

dictionary main.n_images = [counter_images_per_class](#)[class_name]

main.py 파일의 755 번째 라인에서 정의되었습니다.

7.16.2.87 nargs

main.nargs

main.py 파일의 344 번째 라인에서 정의되었습니다.

7.16.2.88 nd

main.nd = len([dr_data](#))

main.py 파일의 584 번째 라인에서 정의되었습니다.

7.16.2.89 no_animation

main.no_animation

main.py 파일의 380 번째 라인에서 정의되었습니다.

7.16.2.90 no_plot

main.no_plot

main.py 파일의 402 번째 라인에서 정의되었습니다.

7.16.2.91 output_img_path

string main.output_img_path = [results_files_path](#) + "/images/detections_one_by_one/" + class_name + "_detection" + str(idx) + ".jpg"

main.py 파일의 717 번째 라인에서 정의되었습니다.

7.16.2.92 output_path

```
string main.output_path = results_files_path + "/ground-truth-info.png"
```

main.py 파일의 833 번째 라인에서 정의되었습니다.

7.16.2.93 ov

```
list main.ov = iw * ih / ua
```

main.py 파일의 630 번째 라인에서 정의되었습니다.

7.16.2.94 ovmax

```
list main.ovmax = -1
```

main.py 파일의 615 번째 라인에서 정의되었습니다.

7.16.2.95 parser

```
main.parser = argparse.ArgumentParser()
```

main.py 파일의 339 번째 라인에서 정의되었습니다.

7.16.2.96 plot_color

```
string main.plot_color = 'forestgreen'
```

main.py 파일의 835 번째 라인에서 정의되었습니다.

7.16.2.97 plot_title

```
string main.plot_title = "ground-truth\n"
```

main.py 파일의 830 번째 라인에서 정의되었습니다.

7.16.2.98 prec

```
list main.prec = tp[:]
```

main.py 파일의 737 번째 라인에서 정의되었습니다.

7.16.2.99 pred_bboxes

```
main.pred_bboxes
```

main.py 파일의 308 번째 라인에서 정의되었습니다.

7.16.2.100 pred_bboxes_

```
main.pred_bboxes_
```

main.py 파일의 312 번째 라인에서 정의되었습니다.

7.16.2.101 pred_labels

```
main.pred_labels
```

main.py 파일의 308 번째 라인에서 정의되었습니다.

7.16.2.102 pred_labels_

```
main.pred_labels_
```

main.py 파일의 312 번째 라인에서 정의되었습니다.

7.16.2.103 pred_scores

```
main.pred_scores
```

main.py 파일의 308 번째 라인에서 정의되었습니다.

7.16.2.104 pred_scores_

```
main.pred_scores_
```

main.py 파일의 312 번째 라인에서 정의되었습니다.

7.16.2.105 rank_pos

```
main.rank_pos = str(idx+1)
```

main.py 파일의 694 번째 라인에서 정의되었습니다.

7.16.2.106 rec

```
list main.rec = tp[:]
```

main.py 파일의 733 번째 라인에서 정의되었습니다.

7.16.2.107 results_files_path

```
string main.results_files_path = "results"
```

main.py 파일의 407 번째 라인에서 정의되었습니다.

7.16.2.108 reverse

```
main.reverse
```

main.py 파일의 559 번째 라인에서 정의되었습니다.

7.16.2.109 right

```
main.right
```

main.py 파일의 450 번째 라인에서 정의되었습니다.

7.16.2.110 rounded_prec

```
list main.rounded_prec = [ '%.2f' % elem for elem in prec ]
```

main.py 파일의 748 번째 라인에서 정의되었습니다.

7.16.2.111 rounded_rec

```
list main.rounded_rec = [ '%.2f' % elem for elem in rec ]
```

main.py 파일의 749 번째 라인에서 정의되었습니다.

7.16.2.112 show_animation

```
bool main.show_animation = False
```

main.py 파일의 385 번째 라인에서 정의되었습니다.

7.16.2.113 sizes

```
list main.sizes = [sizes[0][0], sizes[1][0]]
```

main.py 파일의 311 번째 라인에서 정의되었습니다.

7.16.2.114 specific_iou_classes

```
main.specific_iou_classes = args.set_class_iou[::2]
```

main.py 파일의 510 번째 라인에서 정의되었습니다.

7.16.2.115 specific_iou_flagged

```
bool main.specific_iou_flagged = False
```

main.py 파일의 365 번째 라인에서 정의되었습니다.

7.16.2.116 status

```
string main.status = "NO MATCH FOUND!"
```

main.py 파일의 637 번째 라인에서 정의되었습니다.

7.16.2.117 str

```
main.str
```

main.py 파일의 344 번째 라인에서 정의되었습니다.

7.16.2.118 sum_AP

```
float main.sum_AP = 0.0
```

main.py 파일의 566 번째 라인에서 정의되었습니다.

7.16.2.119 TEMP_FILES_PATH

```
string main.TEMP_FILES_PATH = ".temp_files"
```

main.py 파일의 404 번째 라인에서 정의되었습니다.

7.16.2.120 temp_path

```
main.temp_path = os.path.join(DR_PATH, (file_id + ".txt"))
```

main.py 파일의 437 번째 라인에서 정의되었습니다.

7.16.2.121 test_dataloader

```
main.test_dataloader
```

초기값:

```
1 = data_.DataLoader(testset,
2                               batch_size=1,
3                               num_workers=8,
4                               shuffle=False,
5                               pin_memory=True)
```

main.py 파일의 295 번째 라인에서 정의되었습니다.

7.16.2.122 testset

```
main.testset = TestDataset(opt)
```

main.py 파일의 294 번째 라인에서 정의되었습니다.

7.16.2.123 text

```
string main.text = "Image: " + ground_truth_img[0] + "
```

main.py 파일의 680 번째 라인에서 정의되었습니다.

7.16.2.124 tmp_class_name

```
main.tmp_class_name
```

main.py 파일의 547 번째 라인에서 정의되었습니다.

7.16.2.125 to_show

```
bool main.to_show = False
```

main.py 파일의 834 번째 라인에서 정의되었습니다.

7.16.2.126 top

```
main.top
```

main.py 파일의 450 번째 라인에서 정의되었습니다.

7.16.2.127 tp

```
list main.tp = [0] * nd
```

main.py 파일의 585 번째 라인에서 정의되었습니다.

7.16.2.128 trainer

```
main.trainer = FasterRCNNTrainer(faster_rcnn).cuda()
```

main.py 파일의 301 번째 라인에서 정의되었습니다.

7.16.2.129 true_p_bar

```
dictionary main.true_p_bar = count_true_positives
```

main.py 파일의 880 번째 라인에서 정의되었습니다.

7.16.2.130 type

```
main.type
```

main.py 파일의 344 번째 라인에서 정의되었습니다.

7.16.2.131 ua

```
tuple main.ua
```

초기값:

```
1 = (bb[2] - bb[0] + 1) * (bb[3] - bb[1] + 1) + (bbgt[2] - bbgt[0]  
2 + 1) * (bbgt[3] - bbgt[1] + 1) - iw * ih
```

main.py 파일의 628 번째 라인에서 정의되었습니다.

7.16.2.132 v_pos

```
main.v_pos = int(height - margin - (bottom_border / 2.0))
```

main.py 파일의 679 번째 라인에서 정의되었습니다.

7.16.2.133 white

```
tuple main.white = (255,255,255)
```

main.py 파일의 673 번째 라인에서 정의되었습니다.

7.16.2.134 widht

`main.widht`

`main.py` 파일의 671 번째 라인에서 정의되었습니다.

7.16.2.135 window_title

`string main.window_title = "ground-truth-info"`

`main.py` 파일의 829 번째 라인에서 정의되었습니다.

7.16.2.136 x_label

`string main.x_label = "Number of objects per class"`

`main.py` 파일의 832 번째 라인에서 정의되었습니다.

7.17 model 네임스페이스 참조

네임스페이스

- [faster_rcnn](#)
- [roi_module](#)
- [rpn](#)
- [utils](#)
- [vgg16](#)

7.18 model.faster_rcnn 네임스페이스 참조

클래스

- class [FasterRCNN](#)

7.19 model.roi_module 네임스페이스 참조

클래스

- class [RoI](#)
- class [RoIPooling2D](#)
- class [VGG16RoIHead](#)

함수

- def `normal_init` (m, mean, stddev, `truncated`=False)
- def `load_kernel` (kernel_name, code, **kwargs)
- def `GET_BLOCKS` (N, K=`CUDA_NUM_THREADS`)

변수

- `Stream` = namedtuple('Stream', ['ptr'])
- `for_each_device`
- int `CUDA_NUM_THREADS` = 1024
- string `kernel_forward`
- string `kernel_backward`

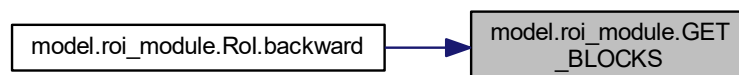
7.19.1 함수 문서화

7.19.1.1 GET_BLOCKS()

```
def model.roi_module.GET_BLOCKS (
    N,
    K = CUDA_NUM_THREADS )

roi_module.py 파일의 108 번째 라인에서 정의되었습니다.
108 def GET_BLOCKS(N, K=CUDA_NUM_THREADS):
109     return (N + K - 1) // K
110
111
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



7.19.1.2 load_kernel()

```
def model.roi_module.load_kernel (
    kernel_name,
    code,
    ** kwargs )

roi_module.py 파일의 98 번째 라인에서 정의되었습니다.
98 def load_kernel(kernel_name, code, **kwargs):
99     cp.cuda.runtime.free(0)
100     code = Template(code).substitute(**kwargs)
101     kernel_code = cupy.cuda.compile_with_cache(code)
102     return kernel_code.get_function(kernel_name)
103
104
```

7.19.1.3 normal_init()

```
def model.roi_module.normal_init (
    m,
    mean,
    stddev,
    truncated = False )
```

weight initializer: truncated normal and random normal.

roi_module.py 파일의 12 번째 라인에서 정의되었습니다.

```
12 def normal_init(m, mean, stddev, truncated=False):
13     """
14     weight initializer: truncated normal and random normal.
15     """
16     # x is a parameter
17     if truncated:
18         m.weight.data.normal_().fmod_(2).mul_(stddev).add_(mean) # not a perfect approximation
19     else:
20         m.weight.data.normal_(mean, stddev)
21         m.bias.data.zero_()
22
```

7.19.2 변수 문서화

7.19.2.1 CUDA_NUM_THREADS

```
int model.roi_module.CUDA_NUM_THREADS = 1024
```

roi_module.py 파일의 105 번째 라인에서 정의되었습니다.

7.19.2.2 for_each_device

```
model.roi_module.for_each_device
```

roi_module.py 파일의 97 번째 라인에서 정의되었습니다.

7.19.2.3 kernel_backward

```
string model.roi_module.kernel_backward
```

roi_module.py 파일의 244 번째 라인에서 정의되었습니다.

7.19.2.4 kernel_forward

string model.roi_module.kernel_forward

roi_module.py 파일의 181 번째 라인에서 정의되었습니다.

7.19.2.5 Stream

model.roi_module.Stream = namedtuple('Stream', ['ptr'])

roi_module.py 파일의 94 번째 라인에서 정의되었습니다.

7.20 model.rpn 네임스페이스 참조

클래스

- class [RegionProposalNetwork](#)

함수

- def [normal_init](#) (m, mean, stddev, [truncated](#)=False)

7.20.1 함수 문서화

7.20.1.1 normal_init()

```
def model.rpn.normal_init (
    m,
    mean,
    stddev,
    truncated = False )
```

rpn.py 파일의 171 번째 라인에서 정의되었습니다.

```
171 def normal_init(m, mean, stddev, truncated=False):
172     # x is a parameter
173     if truncated:
174         m.weight.data.normal_().fmod_(2).mul_(stddev).add_(mean) # not a perfect approximation
175     else:
176         m.weight.data.normal_(mean, stddev)
177         m.bias.data.zero_()
```

7.21 model.utils 네임스페이스 참조

네임스페이스

- [bbox_tools](#)
- [nms](#)
- [roi_sample](#)
- [rpn_gt_loc_label](#)

7.22 model.utils.bbox_tools 네임스페이스 참조

함수

- [def bbox2loc](#) (src_bbox, dst_bbox)
- [def bbox_iou](#) (bbox_a, bbox_b)
- [def loc2bbox](#) (src_bbox, loc)

7.22.1 함수 문서화

7.22.1.1 bbox2loc()

```
def model.utils.bbox_tools.bbox2loc (
    src_bbox,
    dst_bbox )
```

Encodes the source and the destination bounding boxes to "loc".

Given bounding boxes, this function computes offsets and scales to match the source bounding boxes to the target bounding boxes. Mathematically, given a bounding box whose center is $(y, x) = (p_y, p_x)$ and size (p_h, p_w) and the target bounding box whose center is (g_y, g_x) and size (g_h, g_w) , the offsets and scales (t_y, t_x, t_h, t_w) can be computed by the following formulas.

```
* :math:`t_y = \frac{(g_y - p_y)}{p_h}`
* :math:`t_x = \frac{(g_x - p_x)}{p_w}`
* :math:`t_h = \log(\frac{g_h}{p_h})`
* :math:`t_w = \log(\frac{g_w}{p_w})`
```

The output is same type as the type of the inputs.
The encoding formulas are used in works such as R-CNN [1].

.. [1] Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik. \ Rich feature hierarchies for accurate object detection and semantic \ segmentation. CVPR 2014.

Args:

```
src_bbox (array): An image coordinate array whose shape is
    :math:`(R, 4)` where  $R$  is the number of bounding boxes.
    These coordinates are
    :math:`(p_{ymin}, p_{xmin}, p_{ymax}, p_{xmax})`
dst_bbox (array): An image coordinate array whose shape is
```

```

:math:`(R, 4)`'.
These coordinates are
:math:`g_{ymin}, g_{xmin}, g_{ymax}, g_{xmax}`'.

Returns:
array:
Bounding box offsets and scales from :obj:`src_bbox` \
to :obj:`dst_bbox`. \
This has shape :math:`(R, 4)`'.
The second axis contains four values :math:`t_y, t_x, t_h, t_w`'.

```

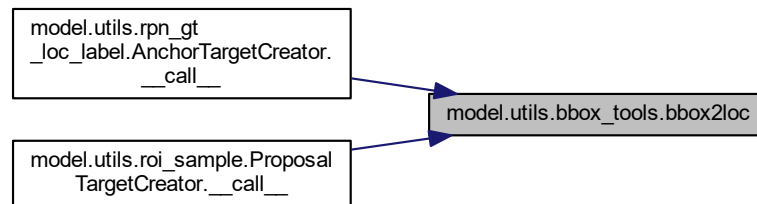
bbox_tools.py 파일의 4 번째 라인에서 정의되었습니다.

```

4 def bbox2loc(src_bbox, dst_bbox):
5     """Encodes the source and the destination bounding boxes to "loc".
6
7     Given bounding boxes, this function computes offsets and scales
8     to match the source bounding boxes to the target bounding boxes.
9     Mathematically, given a bounding box whose center is
10     :math:`(y, x) = p_y, p_x` and
11     size :math:`p_h, p_w` and the target bounding box whose center is
12     :math:`g_y, g_x` and size :math:`g_h, g_w`, the offsets and scales
13     :math:`t_y, t_x, t_h, t_w` can be computed by the following formulas.
14
15     * :math:`t_y = \frac{g_y - p_y}{p_h}`
16     * :math:`t_x = \frac{g_x - p_x}{p_w}`
17     * :math:`t_h = \log(\frac{g_h}{p_h})`
18     * :math:`t_w = \log(\frac{g_w}{p_w})`
19
20     The output is same type as the type of the inputs.
21     The encoding formulas are used in works such as R-CNN [#]_.
22
23     .. [#] Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik. \
24     Rich feature hierarchies for accurate object detection and semantic \
25     segmentation. CVPR 2014.
26
27     Args:
28         src_bbox (array): An image coordinate array whose shape is
29             :math:`(R, 4)`'. :math:`R` is the number of bounding boxes.
30             These coordinates are
31             :math:`p_{ymin}, p_{xmin}, p_{ymax}, p_{xmax}`'.
32         dst_bbox (array): An image coordinate array whose shape is
33             :math:`(R, 4)`'.
34             These coordinates are
35             :math:`g_{ymin}, g_{xmin}, g_{ymax}, g_{xmax}`'.
36
37     Returns:
38         array:
39             Bounding box offsets and scales from :obj:`src_bbox` \
40             to :obj:`dst_bbox`. \
41             This has shape :math:`(R, 4)`'.
42             The second axis contains four values :math:`t_y, t_x, t_h, t_w`'.
43
44     """
45
46     height = src_bbox[:, 2] - src_bbox[:, 0]
47     width = src_bbox[:, 3] - src_bbox[:, 1]
48     ctr_y = src_bbox[:, 0] + 0.5 * height
49     ctr_x = src_bbox[:, 1] + 0.5 * width
50
51     base_height = dst_bbox[:, 2] - dst_bbox[:, 0]
52     base_width = dst_bbox[:, 3] - dst_bbox[:, 1]
53     base_ctr_y = dst_bbox[:, 0] + 0.5 * base_height
54     base_ctr_x = dst_bbox[:, 1] + 0.5 * base_width
55
56     eps = xp.finfo(height.dtype).eps
57     height = xp.maximum(height, eps)
58     width = xp.maximum(width, eps)
59
60     dy = (base_ctr_y - ctr_y) / height
61     dx = (base_ctr_x - ctr_x) / width
62     dh = xp.log(base_height / height)
63     dw = xp.log(base_width / width)
64
65     loc = xp.vstack((dy, dx, dh, dw)).transpose()
66     return loc
67
68

```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



7.22.1.2 bbox_iou()

```
def model.utils.bbox_tools.bbox_iou (
    bbox_a,
    bbox_b )
```

Calculate the Intersection of Unions (IoUs) between bounding boxes.

IoU is calculated as a ratio of area of the intersection and area of the union.

This function accepts both :obj:`numpy.ndarray` and :obj:`cupy.ndarray` as inputs. Please note that both :obj:`bbox_a` and :obj:`bbox_b` need to be same type.

The output is same type as the type of the inputs.

Args:

`bbox_a` (array): An array whose shape is :math:`(N, 4)`
 :math:`N` is the number of bounding boxes.
 The dtype should be :obj:`numpy.float32`.
`bbox_b` (array): An array similar to :obj:`bbox_a`,
 whose shape is :math:`(K, 4)`
 The dtype should be :obj:`numpy.float32`.

Returns:

array:
 An array whose shape is :math:`(N, K)`
 An element at index :math:`(n, k)` contains IoUs between
 :math:`n`th bounding box in :obj:`bbox_a` and :math:`k`th bounding
 box in :obj:`bbox_b`.

`bbox_tools.py` 파일의 69 번째 라인에서 정의되었습니다.

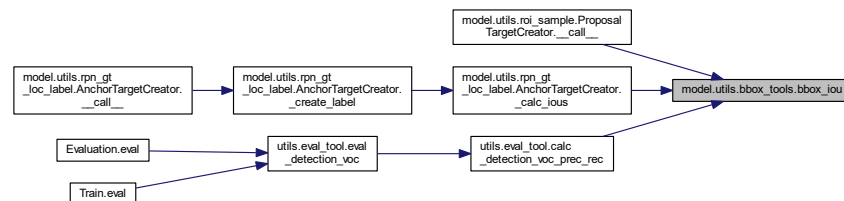
```
69 def bbox_iou(bbox_a, bbox_b):
70     """Calculate the Intersection of Unions (IoUs) between bounding boxes.
71
72     IoU is calculated as a ratio of area of the intersection
73     and area of the union.
74
75     This function accepts both :obj:`numpy.ndarray` and :obj:`cupy.ndarray` as
76     inputs. Please note that both :obj:`bbox_a` and :obj:`bbox_b` need to be
77     same type.
78     The output is same type as the type of the inputs.
79
80     Args:
81         bbox_a (array): An array whose shape is :math:`(N, 4)`
82             :math:`N` is the number of bounding boxes.
```

```

83         The dtype should be :obj:`numpy.float32`.
84         bbox_b (array): An array similar to :obj:`bbox_a`,
85         whose shape is :math:`(K, 4)` .
86         The dtype should be :obj:`numpy.float32`.
87
88     Returns:
89         array:
90         An array whose shape is :math:`(N, K)` . \
91         An element at index :math:`(n, k)` contains IoUs between \
92         :math:`n` th bounding box in :obj:`bbox_a` and :math:`k` th bounding \
93         box in :obj:`bbox_b`.
94
95     """
96     if bbox_a.shape[1] != 4 or bbox_b.shape[1] != 4:
97         raise IndexError
98
99     # top left
100     tl = xp.maximum(bbox_a[:, None, :2], bbox_b[:, :2])
101     # bottom right
102     br = xp.minimum(bbox_a[:, None, 2:], bbox_b[:, 2:])
103
104     area_i = xp.prod(br - tl, axis=2) * (tl < br).all(axis=2)
105     area_a = xp.prod(bbox_a[:, :2] - bbox_a[:, :2], axis=1)
106     area_b = xp.prod(bbox_b[:, :2] - bbox_b[:, :2], axis=1)
107     return area_i / (area_a[:, None] + area_b - area_i)
108
109
110

```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



7.22.1.3 loc2bbox()

```

def model.utils.bbox_tools.loc2bbox (
    src_bbox,
    loc )

```

Decode bounding boxes from bounding box offsets and scales.

Given bounding box offsets and scales computed by :meth:`bbox2loc`, this function decodes the representation to coordinates in 2D image coordinates.

Given scales and offsets :math:`t_y, t_x, t_h, t_w` and a bounding box whose center is :math:`(y, x) = p_y, p_x` and size :math:`p_h, p_w`, the decoded bounding box's center :math:\hat{y}, \hat{x}` and size :math:\hat{h}, \hat{w}` are calculated by the following formulas.

```

* :math:`\hat{y} = p_h t_y + p_y`
* :math:`\hat{x} = p_w t_x + p_x`
* :math:`\hat{h} = p_h \exp(t_h)`
* :math:`\hat{w} = p_w \exp(t_w)`

```

The decoding formulas are used in works such as R-CNN [#]_.

The output is same type as the type of the inputs.

.. [#] Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik. \ Rich feature hierarchies for accurate object detection and semantic \ segmentation. CVPR 2014.

Args:

src_bbox (array): A coordinates of bounding boxes.
Its shape is :math:(R, 4)'. These coordinates are :math:p_{ymin}, p_{xmin}, p_{ymax}, p_{xmax}'.
loc (array): An array with offsets and scales.
The shapes of :obj:'src_bbox' and :obj:'loc' should be same.
This contains values :math:t_y, t_x, t_h, t_w'.

Returns:

array:
Decoded bounding box coordinates. Its shape is :math:(R, 4)'. \ The second axis contains four values \ :math:\\hat{g}_{ymin}, \\hat{g}_{xmin}, \\hat{g}_{ymax}, \\hat{g}_{xmax}'.

bbox_tools.py 파일의 111 번째 라인에서 정의되었습니다.

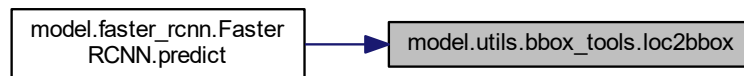
```
111 def loc2bbox(src_bbox, loc):
112     """Decode bounding boxes from bounding box offsets and scales.
113
114     Given bounding box offsets and scales computed by
115     :meth:'bbox2loc', this function decodes the representation to
116     coordinates in 2D image coordinates.
117
118     Given scales and offsets :math:t_y, t_x, t_h, t_w' and a bounding
119     box whose center is :math:(y, x) = p_y, p_x' and size :math:p_h, p_w',
120     the decoded bounding box's center :math:\\hat{g}_y', :math:\\hat{g}_x'
121     and size :math:\\hat{g}_h', :math:\\hat{g}_w' are calculated
122     by the following formulas.
123
124     * :math:\\hat{g}_y = p_h t_y + p_y'
125     * :math:\\hat{g}_x = p_w t_x + p_x'
126     * :math:\\hat{g}_h = p_h \\exp(t_h)'
127     * :math:\\hat{g}_w = p_w \\exp(t_w)'
128
129     The decoding formulas are used in works such as R-CNN [#]_.
130
131     The output is same type as the type of the inputs.
132
133     .. [#] Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik. \
134     Rich feature hierarchies for accurate object detection and semantic \
135     segmentation. CVPR 2014.
136
137     Args:
138         src_bbox (array): A coordinates of bounding boxes.
139         Its shape is :math:(R, 4)'. These coordinates are
140         :math:p_{ymin}, p_{xmin}, p_{ymax}, p_{xmax}'.
141         loc (array): An array with offsets and scales.
142         The shapes of :obj:'src_bbox' and :obj:'loc' should be same.
143         This contains values :math:t_y, t_x, t_h, t_w'.
144
145     Returns:
146         array:
147         Decoded bounding box coordinates. Its shape is :math:(R, 4)'. \
148         The second axis contains four values \
149         :math:\\hat{g}_{ymin}, \\hat{g}_{xmin}, \\hat{g}_{ymax}, \\hat{g}_{xmax}'.
150
151     """
152
153
154     if src_bbox.shape[0] == 0:
155         return xp.zeros((0, 4), dtype=loc.dtype)
156
157     src_bbox = src_bbox.astype(src_bbox.dtype, copy=False)
158
159     src_height = src_bbox[:, 2] - src_bbox[:, 0]
160     src_width = src_bbox[:, 3] - src_bbox[:, 1]
161     src_ctr_y = src_bbox[:, 0] + 0.5 * src_height
162     src_ctr_x = src_bbox[:, 1] + 0.5 * src_width
163
164     dy = loc[:, 0::4]
165     dx = loc[:, 1::4]
166     dh = loc[:, 2::4]
167     dw = loc[:, 3::4]
```

```

168
169     ctr_y = dy * src_height[:, xp.newaxis] + src_ctr_y[:, xp.newaxis]
170     ctr_x = dx * src_width[:, xp.newaxis] + src_ctr_x[:, xp.newaxis]
171     h = xp.exp(dh) * src_height[:, xp.newaxis]
172     w = xp.exp(dw) * src_width[:, xp.newaxis]
173
174     dst_bbox = xp.zeros(loc.shape, dtype=loc.dtype)
175     dst_bbox[:, 0::4] = ctr_y - 0.5 * h
176     dst_bbox[:, 1::4] = ctr_x - 0.5 * w
177     dst_bbox[:, 2::4] = ctr_y + 0.5 * h
178     dst_bbox[:, 3::4] = ctr_x + 0.5 * w
179
180     return dst_bbox

```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



7.23 model.utils.nms 네임스페이스 참조

네임스페이스

- [_nms_gpu_post_py](#)
- [build](#)
- [non_maximum_suppression](#)

7.24 model.utils.nms._nms_gpu_post_py 네임스페이스 참조

함수

- [def _nms_gpu_post](#) ([mask](#), [n_bbox](#), [threads_per_block](#), [col_blocks](#))

7.24.1 함수 문서화

7.24.1.1 `_nms_gpu_post()`

```
def model.utils.nms._nms_gpu_post_py._nms_gpu_post (
    mask,
    n_bbox,
    threads_per_block,
    col_blocks ) [private]
```

`_nms_gpu_post_py.py` 파일의 4 번째 라인에서 정의되었습니다.

```
4 def _nms_gpu_post ( mask,
5     n_bbox,
6     threads_per_block,
7     col_blocks
8 ):
9     n_selection = 0
10    one_u11 = np.array([1], dtype=np.uint64)
11    selection = np.zeros((n_bbox,), dtype=np.int32)
12    remv = np.zeros((col_blocks,), dtype=np.uint64)
13
14    for i in range(n_bbox):
15        nblock = i // threads_per_block
16        inblock = i % threads_per_block
17
18        if not (remv[nblock] & one_u11 << inblock):
19            selection[n_selection] = i
20            n_selection += 1
21
22        index = i * col_blocks
23        for j in range(nblock, col_blocks):
24            remv[j] |= mask[index + j]
25    return selection, n_selection
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



7.25 `model.utils.nms.build` 네임스페이스 참조

변수

- list `ext_modules` = `[Extension("_nms_gpu_post", ["_nms_gpu_post.pyx"])]`
- `name`
- `cmdclass`

7.25.1 변수 문서화

7.25.1.1 `cmdclass`

`model.utils.nms.build.cmdclass`

`build.py` 파일의 8 번째 라인에서 정의되었습니다.

7.25.1.2 ext_modules

```
model.utils.nms.build.ext_modules = [Extension("_nms_gpu_post", ["_nms_gpu_post.pyx"])]
```

build.py 파일의 5 번째 라인에서 정의되었습니다.

7.25.1.3 name

```
model.utils.nms.build.name
```

build.py 파일의 7 번째 라인에서 정의되었습니다.

7.26 model.utils.nms.non_maximum_suppression 네임스페이스 참조

함수

- `def __load_kernel (kernel_name, code, options=())`
- `def non_maximum_suppression (bbox, thresh, score=None, limit=None)`
- `def __non_maximum_suppression_gpu (bbox, thresh, score=None, limit=None)`
- `def __call_nms_kernel (bbox, thresh)`

변수

- `for_each_device`
- `string __nms_gpu_code`

7.26.1 함수 문서화

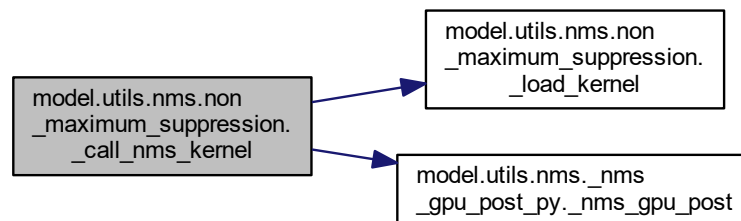
7.26.1.1 `_call_nms_kernel()`

```
def model.utils.nms.non_maximum_suppression._call_nms_kernel (
    bbox,
    thresh ) [private]
```

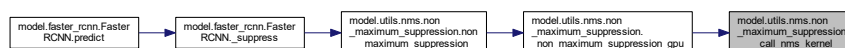
`non_maximum_suppression.py` 파일의 155 번째 라인에서 정의되었습니다.

```
155 def _call_nms_kernel(bbox, thresh):
156     # PyTorch does not support unsigned long Tensor.
157     # Doesn't matter,since it returns ndarray finally.
158     # So I'll keep it unmodified.
159     n_bbox = bbox.shape[0]
160     threads_per_block = 64
161     col_blocks = np.ceil(n_bbox / threads_per_block).astype(np.int32)
162     blocks = (col_blocks, col_blocks, 1)
163     threads = (threads_per_block, 1, 1)
164
165     mask_dev = cp.zeros((n_bbox * col_blocks,), dtype=np.uint64)
166     bbox = cp.ascontiguousarray(bbox, dtype=np.float32) # NOTE: ㄱ成ㄱ의
167     kern = _load_kernel('nms_kernel', _nms_gpu_code)
168     kern(blocks, threads, args=(cp.int32(n_bbox), cp.float32(thresh),
169                               bbox, mask_dev))
170
171     mask_host = mask_dev.get()
172     selection, n_selec = _nms_gpu_post(
173         mask_host, n_bbox, threads_per_block, col_blocks)
174     return selection, n_selec
```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



이 함수를 호출하는 함수들에 대한 그래프입니다.:



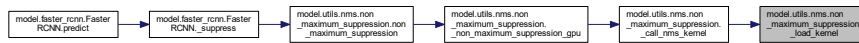
7.26.1.2 `_load_kernel()`

```
def model.utils.nms.non_maximum_suppression._load_kernel (
    kernel_name,
    code,
    options = () ) [private]
```

non_maximum_suppression.py 파일의 13 번째 라인에서 정의되었습니다.

```
13 def _load_kernel(kernel_name, code, options=()):
14     cp.cuda.runtime.free(0)
15     assert isinstance(options, tuple)
16     kernel_code = cp.cuda.compile_with_cache(code, options=options)
17     return kernel_code.get_function(kernel_name)
18
19
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



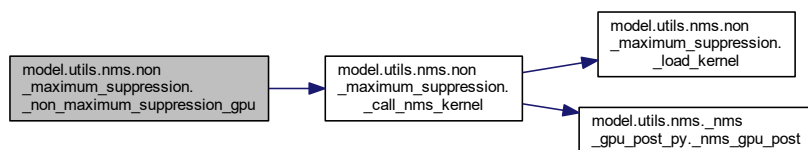
7.26.1.3 _non_maximum_suppression_gpu()

```
def model.utils.nms.non_maximum_suppression._non_maximum_suppression_gpu (
    bbox,
    thresh,
    score = None,
    limit = None ) [private]
```

non_maximum_suppression.py 파일의 69 번째 라인에서 정의되었습니다.

```
69 def _non_maximum_suppression_gpu(bbox, thresh, score=None, limit=None):
70     if len(bbox) == 0:
71         return cp.zeros((0,), dtype=np.int32)
72
73     n_bbox = bbox.shape[0]
74
75     if score is not None:
76         order = score.argsort()[::-1].astype(np.int32)
77     else:
78         order = cp.arange(n_bbox, dtype=np.int32)
79
80     sorted_bbox = bbox[order, :]
81     selec, n_selec = _call_nms_kernel(
82         sorted_bbox, thresh)
83     selec = selec[:n_selec]
84     selec = order[selec]
85     if limit is not None:
86         selec = selec[:limit]
87     return cp.asnumpy(selec)
88
89
```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



이 함수를 호출하는 함수들에 대한 그래프입니다.:



7.26.1.4 non_maximum_suppression()

```
def model.utils.nms.non_maximum_suppression.non_maximum_suppression (
    bbox,
    thresh,
    score = None,
    limit = None )
```

Suppress bounding boxes according to their IoUs.

This method checks each bounding box sequentially and selects the bounding box if the Intersection over Unions (IoUs) between the bounding box and the previously selected bounding boxes is less than `:obj:'thresh'`. This method is mainly used as postprocessing of object detection. The bounding boxes are selected from ones with higher scores. If `:obj:'score'` is not provided as an argument, the bounding box is ordered by its index in ascending order.

The bounding boxes are expected to be packed into a two dimensional tensor of shape `:math:'(R, 4)'`, where `:math:'R'` is the number of bounding boxes in the image. The second axis represents attributes of the bounding box. They are `:math:'(y_{min}, x_{min}, y_{max}, x_{max})'`, where the four attributes are coordinates of the top left and the bottom right vertices.

`:obj:'score'` is a float array of shape `:math:'(R,)'`. Each score indicates confidence of prediction.

This function accepts both `:obj:'numpy.ndarray'` and `:obj:'cupy.ndarray'` as an input. Please note that both `:obj:'bbox'` and `:obj:'score'` need to be the same type.

The type of the output is the same as the input.

Args:

`bbox (array)`: Bounding boxes to be transformed. The shape is `:math:'(R, 4)'`. `:math:'R'` is the number of bounding boxes.
`thresh (float)`: Threshold of IoUs.
`score (array)`: An array of confidences whose shape is `:math:'(R,)'`.
`limit (int)`: The upper bound of the number of the output bounding boxes. If it is not specified, this method selects as many bounding boxes as possible.

Returns:

`array`:
 An array with indices of bounding boxes that are selected. \
 They are sorted by the scores of bounding boxes in descending \
 order. \
 The shape of this array is `:math:'(K,)'` and its dtype is \
`:obj:'numpy.int32'`. Note that `:math:'K \leq R'`.

`non_maximum_suppression.py` 파일의 20 번째 라인에서 정의되었습니다.

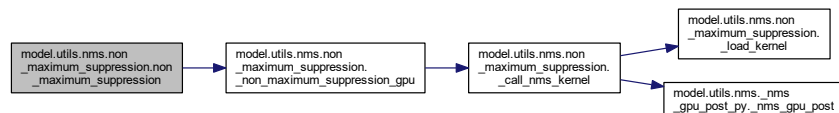
```
20 def non_maximum_suppression(bbox, thresh, score=None,
```

```

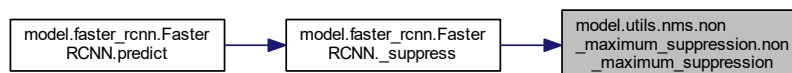
21         limit=None):
22     """Suppress bounding boxes according to their IoUs.
23
24     This method checks each bounding box sequentially and selects the bounding
25     box if the Intersection over Unions (IoUs) between the bounding box and the
26     previously selected bounding boxes is less than :obj:'thresh'. This method
27     is mainly used as postprocessing of object detection.
28     The bounding boxes are selected from ones with higher scores.
29     If :obj:'score' is not provided as an argument, the bounding box
30     is ordered by its index in ascending order.
31
32     The bounding boxes are expected to be packed into a two dimensional
33     tensor of shape :math:'(R, 4)'\text{'}, where :math:'R'\text{' is the number of
34     bounding boxes in the image. The second axis represents attributes of
35     the bounding box. They are :math:'(y_{\min}, x_{\min}, y_{\max}, x_{\max})'\text{'},
36     where the four attributes are coordinates of the top left and the
37     bottom right vertices.
38
39     :obj:'score' is a float array of shape :math:'(R,)\text{'}. Each score indicates
40     confidence of prediction.
41
42     This function accepts both :obj:'numpy.ndarray' and :obj:'cupy.ndarray' as
43     an input. Please note that both :obj:'bbox' and :obj:'score' need to be
44     the same type.
45     The type of the output is the same as the input.
46
47     Args:
48         bbox (array): Bounding boxes to be transformed. The shape is
49             :math:'(R, 4)'\text{'}. :math:'R'\text{' is the number of bounding boxes.
50         thresh (float): Threshold of IoUs.
51         score (array): An array of confidences whose shape is :math:'(R,)\text{'}.
52         limit (int): The upper bound of the number of the output bounding
53             boxes. If it is not specified, this method selects as many
54             bounding boxes as possible.
55
56     Returns:
57         array:
58             An array with indices of bounding boxes that are selected. \
59             They are sorted by the scores of bounding boxes in descending \
60             order. \
61             The shape of this array is :math:'(K,)\text{' and its dtype is \
62             :obj:'numpy.int32'. Note that :math:'K \leq R'\text{'}.
63
64     """
65
66     return _non_maximum_suppression_gpu(bbox, thresh, score, limit)
67
68

```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



이 함수를 호출하는 함수들에 대한 그래프입니다.:



7.26.2 변수 문서화

7.26.2.1 `_nms_gpu_code`

`string model.utils.nms.non_maximum_suppression._nms_gpu_code` [private]

`non_maximum_suppression.py` 파일의 90 번째 라인에서 정의되었습니다.

7.26.2.2 `for_each_device`

`model.utils.nms.non_maximum_suppression.for_each_device`

`non_maximum_suppression.py` 파일의 12 번째 라인에서 정의되었습니다.

7.27 `model.utils.roi_sample` 네임스페이스 참조

클래스

- class [ProposalTargetCreator](#)

7.28 `model.utils.rpn_gt_loc_label` 네임스페이스 참조

클래스

- class [AnchorTargetCreator](#)

함수

- def [_unmap](#) (data, count, index, fill=0)
- def [_get_inside_index](#) (anchor, H, W)
이미지 안에있는 anchor 박스의 index 반환

7.28.1 함수 문서화

7.28.1.1 `_get_inside_index()`

```
def model.utils.rpn_gt_loc_label._get_inside_index (
    anchor,
    H,
    W ) [private]
```

이미지 안에있는 anchor 박스의 index 반환

매개변수

anchor	F.M 한 픽셀당 9개의 anchor 좌표 초기값, shape = (H * W * A, 4)
H	image height
W	image width

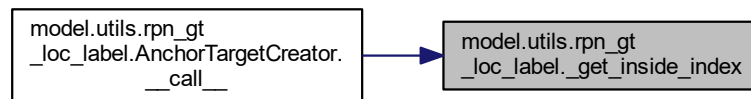
반환값

index_inside : 이미지 size 안에 있는 anchor의 index

rpn_gt_loc_label.py 파일의 177 번째 라인에서 정의되었습니다.

```
177 def _get_inside_index(anchor, H, W):
178     # Calc indicies of anchors which are located completely inside of the image
179     # whose size is speficied.
180     index_inside = np.where(
181         (anchor[:, 0] >= 0) &
182         (anchor[:, 1] >= 0) &
183         (anchor[:, 2] <= H) &
184         (anchor[:, 3] <= W)
185     )[0]
186     return index_inside
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



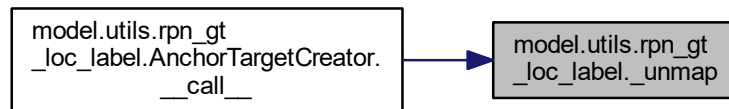
7.28.1.2 _unmap()

```
def model.utils.rpn_gt_loc_label._unmap (
    data,
    count,
    index,
    fill = 0 ) [private]
```

rpn_gt_loc_label.py 파일의 155 번째 라인에서 정의되었습니다.

```
155 def _unmap(data, count, index, fill=0):
156     # Unmap a subset of item (data) back to the original set of items (of size count)
157
158     if len(data.shape) == 1:
159         ret = np.empty((count,), dtype=data.dtype)
160         ret.fill(fill)
161         ret[index] = data
162     else:
163         ret = np.empty((count,) + data.shape[1:], dtype=data.dtype)
164         ret.fill(fill)
165         ret[index, :] = data
166     return ret
167
168
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



7.29 model.vgg16 네임스페이스 참조

함수

- `def decom_vgg16()`
imagenet pretrain vgg16모델 정의 및 상위 4개의 layer 파라미터 고정.

7.29.1 함수 문서화

7.29.1.1 decom_vgg16()

```
def model.vgg16.decom_vgg16 ( )
```

imagenet pretrain vgg16모델 정의 및 상위 4개의 layer 파라미터 고정.

반환값

`nn.Sequential(*features)` : conv layer (마지막 maxpooling layer 제거)
`nn.Sequential(*classifier)` : FC layer

vgg16.py 파일의 9 번째 라인에서 정의되었습니다.

```

9 def decom_vgg16():
10     model = vgg16(pretrained=True)
11     features = list(model.features)[:30]
12     classifier = list(model.classifier)
13     # @brief remove last layer and dropout layer
14     del classifier[6]
15     del classifier[5]
16     del classifier[2]
17
18     # @brief top layer params 고정
19     # @brief Resnet 적용시에도 상위 4개의 layer 고정. (imagenet pretrain 사용)
20     for layer in features[:10]:
21         for p in layer.parameters():
22             p.requires_grad = False
23     return nn.Sequential(*features), nn.Sequential(*classifier)
  
```


이 함수를 호출하는 함수들에 대한 그래프입니다.:



7.30 Train 네임스페이스 참조

함수

- `def eval (dataloader, faster_rcnn, test_num=1000)`
"MAP"를 구하기 위한 함수

변수

- `dataset = Dataset(opt)`
Main Train 함수
- `dataloader`
- `testset = TestDataset(opt)`
- `test_dataloader`
- `faster_rcnn = FasterRCNN()`
- `trainer = FasterRCNNTrainer(faster_rcnn).cuda()`
- `lr_ = opt.lr`
- `int best_map = 0`
- `list loss_list_roi_cls = []`
- `list loss_list_roi_loc = []`
- `list loss_list_rpn_cls = []`
- `list loss_list_rpn_loc = []`
- `scale = at.scalar(scale)`
- `img`
- `bbox`
- `label`
- `loss_list = trainer.train_step(img, bbox, label, scale)`
- `save_optimizer`
- `True`
- `def eval_result = eval(test_dataloader, faster_rcnn, test_num=opt.test_num)`

7.30.1 함수 문서화

7.30.1.1 eval()

```
def Train.eval (
    dataloader,
    faster_rcnn,
    test_num = 1000 )
```

”MAP”를 구하기 위한 함수

매개변수

dataloader	평가할 이미지에 대한 option
faster_rcnn	학습된 model
test_num	평가할 image 개수

반환값

result 'ap' 와 'map' 으로 구성

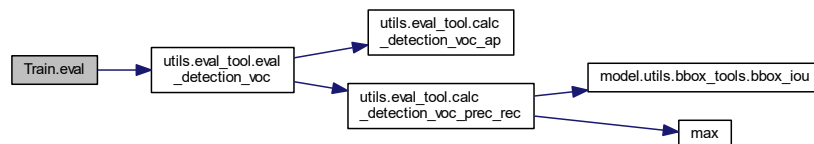
Train.py 파일의 35 번째 라인에서 정의되었습니다.

```

35 def eval(dataloader, faster_rcnn, test_num=1000):
36     pred_bboxes, pred_labels, pred_scores = list(), list(), list()
37     gt_bboxes, gt_labels, gt_difficults = list(), list(), list()
38     for ii, (imgs, sizes, gt_bboxes_, gt_labels_, gt_difficults_) in tqdm(enumerate(dataloader)):
39         sizes = [sizes[0][0], sizes[1][0]]
40         pred_bboxes_, pred_labels_, pred_scores_ = faster_rcnn.predict(imgs, [sizes])
41         gt_bboxes += list(gt_bboxes_.numpy())
42         gt_labels += list(gt_labels_.numpy())
43         gt_difficults += list(gt_difficults_.numpy())
44         pred_bboxes += pred_bboxes_
45         pred_labels += pred_labels_
46         pred_scores += pred_scores_
47         if ii == test_num: break
48
49     result = eval_detection_voc(
50         pred_bboxes, pred_labels, pred_scores,
51         gt_bboxes, gt_labels, gt_difficults,
52         use_07_metric=True)
53     return result
54

```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



7.30.2 변수 문서화

7.30.2.1 bbox

Train.bbox

Train.py 파일의 92 번째 라인에서 정의되었습니다.

7.30.2.2 best_map

```
def Train.best_map = 0
```

Train.py 파일의 83 번째 라인에서 정의되었습니다.

7.30.2.3 dataloader

Train.dataloader

초기값:

```
1 = data_.DataLoader(dataset,
2                               batch_size=1,
3                               shuffle=True,
4                               num_workers=opt.num_workers)
```

Train.py 파일의 60 번째 라인에서 정의되었습니다.

7.30.2.4 dataset

Train.dataset = Dataset(opt)

Main Train 함수

Train.py 파일의 59 번째 라인에서 정의되었습니다.

7.30.2.5 eval_result

```
def Train.eval_result = eval(test_dataloader, faster_rcnn, test_num=opt.test_num)
```

Train.py 파일의 117 번째 라인에서 정의되었습니다.

7.30.2.6 faster_rcnn

Train.faster_rcnn = FasterRCNN()

Train.py 파일의 74 번째 라인에서 정의되었습니다.

7.30.2.7 img

Train.img

Train.py 파일의 92 번째 라인에서 정의되었습니다.

7.30.2.8 label

Train.label

Train.py 파일의 92 번째 라인에서 정의되었습니다.

7.30.2.9 loss_list

```
Train.loss_list = trainer.train_step(img, bbox, label, scale)
```

Train.py 파일의 95 번째 라인에서 정의되었습니다.

7.30.2.10 loss_list_roi_cls

```
list Train.loss_list_roi_cls = []
```

Train.py 파일의 85 번째 라인에서 정의되었습니다.

7.30.2.11 loss_list_roi_loc

```
list Train.loss_list_roi_loc = []
```

Train.py 파일의 86 번째 라인에서 정의되었습니다.

7.30.2.12 loss_list_rpn_cls

```
list Train.loss_list_rpn_cls = []
```

Train.py 파일의 87 번째 라인에서 정의되었습니다.

7.30.2.13 loss_list_rpn_loc

```
list Train.loss_list_rpn_loc = []
```

Train.py 파일의 88 번째 라인에서 정의되었습니다.

7.30.2.14 lr_

```
Train.lr_ = opt.lr
```

Train.py 파일의 82 번째 라인에서 정의되었습니다.

7.30.2.15 save_optimizer

```
Train.save_optimizer
```

Train.py 파일의 109 번째 라인에서 정의되었습니다.

7.30.2.16 scale

```
Train.scale = at.scalar(scale)
```

Train.py 파일의 91 번째 라인에서 정의되었습니다.

7.30.2.17 test_dataloader

```
Train.test_dataloader
```

초기값:

```
1 = data_.DataLoader(testset,
2
3                               batch_size=1,
4                               num_workers=opt.test_num_workers,
5                               shuffle=False, \
6                               pin_memory=True
)
```

Train.py 파일의 66 번째 라인에서 정의되었습니다.

7.30.2.18 testset

```
Train.testset = TestDataset(opt)
```

Train.py 파일의 65 번째 라인에서 정의되었습니다.

7.30.2.19 trainer

```
Train.trainer = FasterRCNNTrainer(faster_rcnn).cuda()
```

Train.py 파일의 75 번째 라인에서 정의되었습니다.

7.30.2.20 True

```
Train.True
```

Train.py 파일의 109 번째 라인에서 정의되었습니다.

7.31 trainer 네임스페이스 참조

네임스페이스

- [trainer](#)

7.32 trainer.trainer 네임스페이스 참조

클래스

- class [FasterRCNNTrainer](#)

함수

- def [_smooth_l1_loss](#) (x, t, in_weight, sigma)
- def [_fast_rcnn_loc_loss](#) (pred_loc, gt_loc, gt_label, sigma)

변수

- [LossTuple](#)

7.32.1 함수 문서화

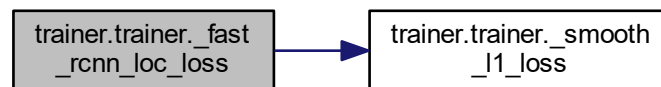
7.32.1.1 `_fast_rcnn_loc_loss()`

```
def trainer.trainer._fast_rcnn_loc_loss (
    pred_loc,
    gt_loc,
    gt_label,
    sigma ) [private]
```

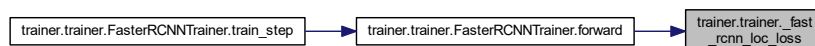
trainer.py 파일의 271 번째 라인에서 정의되었습니다.

```
271 def _fast_rcnn_loc_loss(pred_loc, gt_loc, gt_label, sigma):
272     in_weight = t.zeros(gt_loc.shape).cuda()
273     # Localization loss is calculated only for positive rois.
274     # NOTE: unlike origin implementation,
275     # we don't need inside_weight and outside_weight, they can calculate by gt_label
276     in_weight[(gt_label > 0).view(-1, 1).expand_as(in_weight).cuda()] = 1
277     loc_loss = _smooth_l1_loss(pred_loc, gt_loc, Variable(in_weight), sigma)
278     # Normalize by total number of negtive and positive rois.
279     loc_loss /= (gt_label >= 0).sum() # ignore gt_label==-1 for rpn_loss
280     return loc_loss
```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



이 함수를 호출하는 함수들에 대한 그래프입니다.:



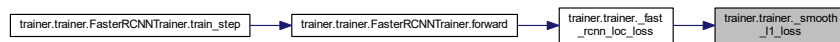
7.32.1.2 `_smooth_l1_loss()`

```
def trainer.trainer._smooth_l1_loss (
    x,
    t,
    in_weight,
    sigma ) [private]
```


trainer.py 파일의 259 번째 라인에서 정의되었습니다.

```
259 def _smooth_l1_loss(x, t, in_weight, sigma):
260     sigma2 = sigma ** 2
261     t = t.float()
262     diff = in_weight * (x - t)
263     abs_diff = diff.abs()
264     flag = (abs_diff.data < (1. / sigma2)).float()
265     flag = Variable(flag)
266     y = (flag * (sigma2 / 2.) * (diff ** 2) +
267          (1 - flag) * (abs_diff - 0.5 / sigma2))
268     return y.sum()
269
270
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



7.32.2 변수 문서화

7.32.2.1 LossTuple

trainer.trainer.LossTuple

초기값:

```
1 = namedtuple('LossTuple',
2             ['rpn_loc_loss',
3              'rpn_cls_loss',
4              'roi_loc_loss',
5              'roi_cls_loss',
6              'total_loss'
7             ])

```

trainer.py 파일의 15 번째 라인에서 정의되었습니다.

7.33 utils 네임스페이스 참조

네임스페이스

- [anchors](#)
- [array_tool](#)
- [config](#)
- [data_load](#)
- [eval_tool](#)
- [py_nms](#)
- [vis_tool](#)

7.34 utils.anchors 네임스페이스 참조

함수

- def `generate_anchor_base` (side_length=16, ratios=[0.5, 1, 2], scales=[0.5, 1, 2], strides=16)
(input image size / 16) * (input image size / 16) * 9 개의 기본 anchor 생성
- def `get_anchors` (anchor_base, feat_stride, height, width)
F.M 한 픽셀에 대응하는 3개의 크기 * 3개의 비율을 가지는 9개의 기본 anchor 생성
- def `get_rois_from_loc_anchors` (anchors, rpn_locs)
RPN 네트워크를 통해 나온 anchor 좌표(상대좌표) 값을 input image에 매칭

7.34.1 함수 문서화

7.34.1.1 generate_anchor_base()

```
def utils.anchors.generate_anchor_base (
    side_length = 16,
    ratios = [0.5, 1, 2],
    scales = [0.5, 1, 2],
    strides = 16 )
```

(input image size / 16) * (input image size / 16) * 9 개의 기본 anchor 생성
base anchor of the original image

Generate anchors for a single 16*16 block. Then transform the anchors to the original image space.

anchors.py 파일의 7 번째 라인에서 정의되었습니다.

```
7 def generate_anchor_base(side_length=16, ratios=[0.5, 1, 2],
8                           scales=[0.5, 1, 2], strides=16):
9     """
10     Generate anchors for a single 16*16 block. Then transform the anchors to the original image space.
11     """
12     py = side_length / 2.
13     px = side_length / 2.
14     # anchor_base shape: <class 'tuple'> (9,4)
15     anchor_base = np.zeros((len(ratios) * len(scales), 4),
16                           dtype=np.float32)
17
18     for i in range(len(ratios)):
19         for j in range(len(scales)):
20             h = side_length * strides * scales[j] * np.sqrt(ratios[i])
21             w = side_length * strides * scales[j] * np.sqrt(1. / ratios[i])
22
23             index = i * len(scales) + j
24             anchor_base[index, 0] = py - h / 2.
25             anchor_base[index, 1] = px - w / 2.
26             anchor_base[index, 2] = py + h / 2.
27             anchor_base[index, 3] = px + w / 2.
28     return anchor_base
29
```

7.34.1.2 get_anchors()

```
def utils.anchors.get_anchors (
    anchor_base,
    feat_stride,
    height,
    width )
```

F.M 한 픽셀에 대응하는 3개의 크기 * 3개의 비율을 가지는 9개의 기본 anchor 생성

매개변수

anchor_base	
feat_stride	16(input image 에서 16개 픽셀에 대응)
height	F.M height
width	F.M width

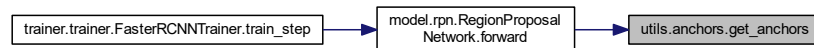
반환값

anchors : F.M 모든 픽셀에 대한 초기 anchor 위치, shape = (F.M height * width * 9 * 4)

anchors.py 파일의 38 번째 라인에서 정의되었습니다.

```
38 def get_anchors(anchor_base, feat_stride, height, width):
39
40     # if height = 512 * width = 512 일 경우
41     # anchors_y = 0, 16, 32 ...
42     anchors_y = np.arange(height) * feat_stride
43     anchors_x = np.arange(width) * feat_stride
44
45     # meshgrid는 사각형의 조합을 의미
46     # anchor_x = shape 512 * 512 / value 0, 16, 32 ...
47     # anchor_y = shape 512 * 512 / value 0, 0, 0 / 1, 1, 1 / 2, 2, 2
48     anchors_x, anchors_y = np.meshgrid(anchors_x, anchors_y)
49
50     # shift shape: (feature map height * feature map width) * 4 (좌표)
51     shift = np.stack((anchors_y.ravel(), anchors_x.ravel(),
52                      anchors_y.ravel(), anchors_x.ravel()), axis=1)
53     anchors = np.repeat(shift, repeats=len(anchor_base), axis=0) + \
54         np.tile(anchor_base, [len(shift), 1])
55     return anchors
56
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



7.34.1.3 get_rois_from_loc_anchors()

```
def utils.anchors.get_rois_from_loc_anchors (
    anchors,
    rpn_locs )
```

RPN 네트워크를 통해 나온 anchor 좌표(상대좌표) 값을 input image에 매칭

매개변수

anchors	F.M 한 픽셀의 9개의 anchor 좌표 (p_{ymin}, p_{xmin}, p_{ymax}, p_{xmax})
rpn_locs	RPN을 통해서 예측한 anchor 좌표, (feature height * feature width * 9 * 4[t_y, t_x, t_h, t_w])

반환값

dst_bbox : 예측된 anchor를 input image 좌표로 변경한 것

주의

RPN을 통해 예측되어 나오는 anchor 박스의 (t_y , t_x , t_h , t_w) 값들은 상대좌표 이다.

Decode bounding boxes from bounding box offsets and scales.

Given bounding box offsets and scales computed by :meth:`bbox2loc`, this function decodes the representation to coordinates in 2D image coordinates.

Given scales and offsets :math:t_y, t_x, t_h, t_w and a bounding box whose center is :math:(y, x) = p_y, p_x and size :math:p_h, p_w, the decoded bounding box's center :math:\hat{g}_y, \hat{g}_x and size :math:\hat{g}_h, \hat{g}_w are calculated by the following formulas.

```
* :math:\hat{g}_y = p_h t_y + p_y
```

```
* :math:\hat{g}_x = p_w t_x + p_x
```

```
* :math:\hat{g}_h = p_h \exp(t_h)
```

```
* :math:\hat{g}_w = p_w \exp(t_w)
```

Args:

anchors (array): A coordinates of bounding boxes.
Its shape is :math:(R, 4). These coordinates are :math:p_{ymin}, p_{xmin}, p_{ymax}, p_{xmax}.
rpn_locs (array): An array with offsets and scales.
The shapes of :obj:`anchors` and :obj:`rpn_locs` should be same.
This contains values :math:t_y, t_x, t_h, t_w.

Returns:

array:
Decoded bounding box coordinates. Its shape is :math:(R, 4).
The second axis contains four values
:math:\hat{g}_{ymin}, \hat{g}_{xmin}, \hat{g}_{ymax}, \hat{g}_{xmax}.

anchors.py 파일의 64 번째 라인에서 정의되었습니다.

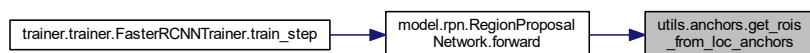
```
64 def get_rois_from_loc_anchors(anchors, rpn_locs):
65     """Decode bounding boxes from bounding box offsets and scales.
66
67     Given bounding box offsets and scales computed by
68     :meth:`bbox2loc`, this function decodes the representation to
69     coordinates in 2D image coordinates.
70
71     Given scales and offsets :math:t_y, t_x, t_h, t_w and a bounding
72     box whose center is :math:(y, x) = p_y, p_x and size :math:p_h, p_w,
73     the decoded bounding box's center :math:\hat{g}_y, \hat{g}_x
74     and size :math:\hat{g}_h, \hat{g}_w are calculated
75     by the following formulas.
76
77     * :math:\hat{g}_y = p_h t_y + p_y
78     * :math:\hat{g}_x = p_w t_x + p_x
79     * :math:\hat{g}_h = p_h \exp(t_h)
80     * :math:\hat{g}_w = p_w \exp(t_w)
81
82     Args:
83         anchors (array): A coordinates of bounding boxes.
84             Its shape is :math:(R, 4). These coordinates are
85             :math:p_{ymin}, p_{xmin}, p_{ymax}, p_{xmax}.
86         rpn_locs (array): An array with offsets and scales.
87             The shapes of :obj:`anchors` and :obj:`rpn_locs` should be same.
88             This contains values :math:t_y, t_x, t_h, t_w.
89
90     Returns:
91         array:
92             Decoded bounding box coordinates. Its shape is :math:(R, 4).
93             The second axis contains four values
94             :math:\hat{g}_{ymin}, \hat{g}_{xmin}, \hat{g}_{ymax}, \hat{g}_{xmax}.
95
96     """
97
```

```

98     src_bbox = anchors
99     src_bbox = src_bbox.astype(src_bbox.dtype, copy=False)
100
101     src_height = src_bbox[:, 2] - src_bbox[:, 0]
102     src_width = src_bbox[:, 3] - src_bbox[:, 1]
103     src_ctr_y = src_bbox[:, 0] + 0.5 * src_height
104     src_ctr_x = src_bbox[:, 1] + 0.5 * src_width
105
106     # 상대적인 좌표 예측
107     dy = rpn_locs[:, 0]
108     dx = rpn_locs[:, 1]
109     dh = rpn_locs[:, 2]
110     dw = rpn_locs[:, 3]
111
112     dst_y = dy * src_height + src_ctr_y
113     dst_x = dx * src_width + src_ctr_x
114     dst_h = np.exp(dh) * src_height
115     dst_w = np.exp(dw) * src_width
116
117     dst_bbox = np.zeros(rpn_locs.shape, dtype=rpn_locs.dtype)
118     dst_bbox[:, 0] = dst_y - 0.5 * dst_h
119     dst_bbox[:, 1] = dst_x - 0.5 * dst_w
120     dst_bbox[:, 2] = dst_y + 0.5 * dst_h
121     dst_bbox[:, 3] = dst_x + 0.5 * dst_w
122
123     return dst_bbox

```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



7.35 utils.array_tool 네임스페이스 참조

함수

- def [tonumpy](#) (data)
- def [totensor](#) (data, cuda=True)
- def [tovariable](#) (data)
- def [scalar](#) (data)

7.35.1 상세한 설명

tools to convert specified type

7.35.2 함수 문서화

7.35.2.1 scalar()

```
def utils.array_tool.scalar (
    data )
```

array_tool.py 파일의 41 번째 라인에서 정의되었습니다.

```
41 def scalar(data):
42     if isinstance(data, np.ndarray):
43         return data.reshape(1) [0]
44     if isinstance(data, t._C._TensorBase):
45         return data.view(1) [0]
46     if isinstance(data, t.autograd.Variable):
47         return data.data.view(1) [0]
```

7.35.2.2 tonumpy()

```
def utils.array_tool.tonumpy (
    data )
```

array_tool.py 파일의 9 번째 라인에서 정의되었습니다.

```
9 def tonumpy(data):
10     if isinstance(data, np.ndarray):
11         return data
12     if isinstance(data, t._C._TensorBase):
13         return data.cpu().numpy()
14     if isinstance(data, t.autograd.Variable):
15         return tonumpy(data.data)
16
17
```

7.35.2.3 totensor()

```
def utils.array_tool.totensor (
    data,
    cuda = True )
```

array_tool.py 파일의 18 번째 라인에서 정의되었습니다.

```
18 def totensor(data, cuda=True):
19     if isinstance(data, np.ndarray):
20         tensor = t.from_numpy(data)
21     if isinstance(data, t._C._TensorBase):
22         tensor = data
23     if isinstance(data, t.autograd.Variable):
24         tensor = data.data
25     if cuda:
26         tensor = tensor.cuda()
27     return tensor
28
29
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



7.35.2.4 tovariable()

```
def utils.array_tool.tovariable (
    data )
```

array_tool.py 파일의 30 번째 라인에서 정의되었습니다.

```
30 def tovariable(data):
31     if isinstance(data, np.ndarray):
32         return tovariable(totensor(data))
33     if isinstance(data, t._C._TensorBase):
34         return t.autograd.Variable(data)
35     if isinstance(data, t.autograd.Variable):
36         return data
37     else:
38         raise ValueError("UnKnow data type: %s, input should be {np.ndarray, Tensor, Variable}"
39                             %type(data))
40
```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



7.36 utils.config 네임스페이스 참조

클래스

- class [Config](#)

변수

- `opt = Config\(\)`

7.36.1 변수 문서화

7.36.1.1 opt

```
utils.config.opt = Config\(\)
```

config.py 파일의 67 번째 라인에서 정의되었습니다.

7.37 utils.data_load 네임스페이스 참조

클래스

- class [VOCBboxDataset](#)
- class [VOCDataset](#)

함수

- def [save_pkl](#) (filename, f)
- def [load_pkl](#) (filename)

변수

- tuple [VOC_BBOX_LABEL_NAMES](#)

7.37.1 함수 문서화

7.37.1.1 load_pkl()

```
def utils.data_load.load_pkl (
    filename )
```

data_load.py 파일의 21 번째 라인에서 정의되었습니다.

```
21 def load_pkl(filename):
22     with open(filename, 'rb') as handle:
23         b = pickle.load(handle)
24         return b
25
26
27 # -----
28 # VOC Objection Datasets
```

7.37.1.2 save_pkl()

```
def utils.data_load.save_pkl (
    filename,
    f )
```

data_load.py 파일의 17 번째 라인에서 정의되었습니다.

```
17 def save_pkl(filename, f):
18     with open(filename, 'wb') as handle:
19         pickle.dump(f, handle, protocol=pickle.HIGHEST_PROTOCOL)
20
```


7.37.2 변수 문서화

7.37.2.1 VOC_BBOX_LABEL_NAMES

tuple utils.data_load.VOC_BBOX_LABEL_NAMES

초기값:

```
1 = (
2     'aeroplane',
3     'bicycle',
4     'bird',
5     'boat',
6     'bottle',
7     'bus',
8     'car',
9     'cat',
10    'chair',
11    'cow',
12    'diningtable',
13    'dog',
14    'horse',
15    'motorbike',
16    'person',
17    'pottedplant',
18    'sheep',
19    'sofa',
20    'train',
21    'tvmonitor')
```

data_load.py 파일의 152 번째 라인에서 정의되었습니다.

7.38 utils.eval_tool 네임스페이스 참조

함수

- def `eval_detection_voc` (pred_bboxes, pred_labels, pred_scores, gt_bboxes, gt_labels, gt_difficults=None, iou_thresh=0.5, use_07_metric=False)
- def `calc_detection_voc_prec_rec` (pred_bboxes, pred_labels, pred_scores, gt_bboxes, gt_labels, gt_difficults=None, iou_thresh=0.5)
- def `calc_detection_voc_ap` (prec, rec, use_07_metric=False)

7.38.1 함수 문서화

7.38.1.1 calc_detection_voc_ap()

```
def utils.eval_tool.calc_detection_voc_ap (
    prec,
    rec,
    use_07_metric = False )
```

Calculate average precisions based on evaluation code of PASCAL VOC.
This function calculates average precisions
from given precisions and recalls.
The code is based on the evaluation code used in PASCAL VOC Challenge.

Args:

```
prec (list of numpy.array): A list of arrays.
    :obj:'prec[l]' indicates precision for class :math:'l'.
    If :obj:'prec[l]' is :obj:'None', this function returns
    :obj:'numpy.nan' for class :math:'l'.
rec (list of numpy.array): A list of arrays.
    :obj:'rec[l]' indicates recall for class :math:'l'.
    If :obj:'rec[l]' is :obj:'None', this function returns
    :obj:'numpy.nan' for class :math:'l'.
use_07_metric (bool): Whether to use PASCAL VOC 2007 evaluation metric
    for calculating average precision. The default value is
    :obj:'False'.
```

Returns:

```
~numpy.ndarray:
    This function returns an array of average precisions.
    The :math:'l'-th value corresponds to the average precision
    for class :math:'l'. If :obj:'prec[l]' or :obj:'rec[l]' is
    :obj:'None', the corresponding value is set to :obj:'numpy.nan'.
```

eval_tool.py 파일의 233 번째 라인에서 정의되었습니다.

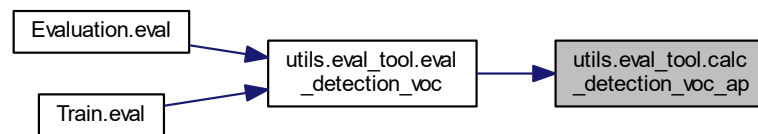
```
233 def calc_detection_voc_ap(prec, rec, use_07_metric=False):
234     """Calculate average precisions based on evaluation code of PASCAL VOC.
235     This function calculates average precisions
236     from given precisions and recalls.
237     The code is based on the evaluation code used in PASCAL VOC Challenge.
238     Args:
239         prec (list of numpy.array): A list of arrays.
240         :obj:'prec[l]' indicates precision for class :math:'l'.
241         If :obj:'prec[l]' is :obj:'None', this function returns
242         :obj:'numpy.nan' for class :math:'l'.
243         rec (list of numpy.array): A list of arrays.
244         :obj:'rec[l]' indicates recall for class :math:'l'.
245         If :obj:'rec[l]' is :obj:'None', this function returns
246         :obj:'numpy.nan' for class :math:'l'.
247         use_07_metric (bool): Whether to use PASCAL VOC 2007 evaluation metric
248         for calculating average precision. The default value is
249         :obj:'False'.
250     Returns:
251         numpy.ndarray:
252         This function returns an array of average precisions.
253         The :math:'l'-th value corresponds to the average precision
254         for class :math:'l'. If :obj:'prec[l]' or :obj:'rec[l]' is
255         :obj:'None', the corresponding value is set to :obj:'numpy.nan'.
256     """
257
258     n_fg_class = len(prec)
259     ap = np.empty(n_fg_class)
260     for l in six.moves.range(n_fg_class):
261         if prec[l] is None or rec[l] is None:
262             ap[l] = np.nan
263             continue
264
265         if use_07_metric:
266             # 11 point metric
267             ap[l] = 0
268             for t in np.arange(0., 1.1, 0.1):
269                 if np.sum(rec[l] >= t) == 0:
270                     p = 0
271                 else:
272                     p = np.max(np.nan_to_num(prec[l])[rec[l] >= t])
273                 ap[l] += p / 11
274         else:
275             # correct AP calculation
276             # first append sentinel values at the end
277             mpre = np.concatenate(([0], np.nan_to_num(prec[l]), [0]))
```

```

278         mrec = np.concatenate(([0], rec[1], [1]))
279
280         mpre = np.maximum.accumulate(mpre[:-1])[:-1]
281
282         # to calculate area under PR curve, look for points
283         # where X axis (recall) changes value
284         i = np.where(mrec[1:] != mrec[:-1])[0]
285
286         # and sum (\Delta recall) * prec
287         ap[1] = np.sum((mrec[i + 1] - mrec[i]) * mpre[i + 1])
288
289     return ap

```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



7.38.1.2 calc_detection_voc_prec_rec()

```

def utils.eval_tool.calc_detection_voc_prec_rec (
    pred_bboxes,
    pred_labels,
    pred_scores,
    gt_bboxes,
    gt_labels,
    gt_difficults = None,
    iou_thresh = 0.5 )

```

Calculate precision and recall based on evaluation code of PASCAL VOC. This function calculates precision and recall of predicted bounding boxes obtained from a dataset which has :math:N' images.

The code is based on the evaluation code used in PASCAL VOC Challenge. Args:

`pred_bboxes` (iterable of numpy.ndarray): An iterable of :math:N' sets of bounding boxes. Its index corresponds to an index for the base dataset. Each element of :code:`obj:'pred_bboxes'` is a set of coordinates of bounding boxes. This is an array whose shape is :math:(R, 4)' , where :math:'R' corresponds to the number of bounding boxes, which may vary among boxes. The second axis corresponds to :math:'y_{min}, x_{min}, y_{max}, x_{max}' of a bounding box.

`pred_labels` (iterable of numpy.ndarray): An iterable of labels. Similar to :code:`obj:'pred_bboxes'`, its index corresponds to an index for the base dataset. Its length is :math:N'.

`pred_scores` (iterable of numpy.ndarray): An iterable of confidence scores for predicted bounding boxes. Similar to :code:`obj:'pred_bboxes'`, its index corresponds to an index for the base dataset. Its length is :math:N'.

`gt_bboxes` (iterable of numpy.ndarray): An iterable of ground truth bounding boxes

whose length is `:math:'N'`. An element of `:obj:'gt_bboxes'` is a bounding box whose shape is `:math:'(R, 4)'`. Note that the number of bounding boxes in each image does not need to be same as the number of corresponding predicted boxes.

`gt_labels` (iterable of `numpy.ndarray`): An iterable of ground truth labels which are organized similarly to `:obj:'gt_bboxes'`.

`gt_difficults` (iterable of `numpy.ndarray`): An iterable of boolean arrays which is organized similarly to `:obj:'gt_bboxes'`. This tells whether the corresponding ground truth bounding box is difficult or not. By default, this is `:obj:'None'`. In that case, this function considers all bounding boxes to be not difficult.

`iou_thresh` (float): A prediction is correct if its Intersection over Union with the ground truth is above this value..

Returns:

tuple of two lists:

This function returns two lists: `:obj:'prec'` and `:obj:'rec'`.

- * `:obj:'prec'`: A list of arrays. `:obj:'prec[l]'` is precision \ for class `:math:'l'`. If class `:math:'l'` does not exist in \ either `:obj:'pred_labels'` or `:obj:'gt_labels'`, `:obj:'prec[l]'` is \ set to `:obj:'None'`.
- * `:obj:'rec'`: A list of arrays. `:obj:'rec[l]'` is recall \ for class `:math:'l'`. If class `:math:'l'` that is not marked as \ difficult does not exist in \ `:obj:'gt_labels'`, `:obj:'rec[l]'` is \ set to `:obj:'None'`.

`eval_tool.py` 파일의 77 번째 라인에서 정의되었습니다.

```
77 def calc_detection_voc_prec_rec(
78     pred_bboxes, pred_labels, pred_scores, gt_bboxes, gt_labels,
79     gt_difficults=None,
80     iou_thresh=0.5):
81     """Calculate precision and recall based on evaluation code of PASCAL VOC.
82     This function calculates precision and recall of
83     predicted bounding boxes obtained from a dataset which has :math:'N'
84     images.
85     The code is based on the evaluation code used in PASCAL VOC Challenge.
86     Args:
87         pred_bboxes (iterable of numpy.ndarray): An iterable of :math:'N'
88             sets of bounding boxes.
89             Its index corresponds to an index for the base dataset.
90             Each element of :obj:'pred_bboxes' is a set of coordinates
91             of bounding boxes. This is an array whose shape is :math:'(R, 4)'\
92             where :math:'R' corresponds
93             to the number of bounding boxes, which may vary among boxes.
94             The second axis corresponds to
95             :math:'y_{min}, x_{min}, y_{max}, x_{max}' of a bounding box.
96         pred_labels (iterable of numpy.ndarray): An iterable of labels.
97             Similar to :obj:'pred_bboxes', its index corresponds to an
98             index for the base dataset. Its length is :math:'N'.
99         pred_scores (iterable of numpy.ndarray): An iterable of confidence
100             scores for predicted bounding boxes. Similar to :obj:'pred_bboxes',
101             its index corresponds to an index for the base dataset.
102             Its length is :math:'N'.
103         gt_bboxes (iterable of numpy.ndarray): An iterable of ground truth
104             bounding boxes
105             whose length is :math:'N'. An element of :obj:'gt_bboxes' is a
106             bounding box whose shape is :math:'(R, 4)'. Note that the number of
107             bounding boxes in each image does not need to be same as the number
108             of corresponding predicted boxes.
109         gt_labels (iterable of numpy.ndarray): An iterable of ground truth
110             labels which are organized similarly to :obj:'gt_bboxes'.
111         gt_difficults (iterable of numpy.ndarray): An iterable of boolean
112             arrays which is organized similarly to :obj:'gt_bboxes'.
113             This tells whether the
114             corresponding ground truth bounding box is difficult or not.
115             By default, this is :obj:'None'. In that case, this function
116             considers all bounding boxes to be not difficult.
117         iou_thresh (float): A prediction is correct if its Intersection over
118             Union with the ground truth is above this value..
119     Returns:
120         tuple of two lists:
121         This function returns two lists: :obj:'prec' and :obj:'rec'.
122         * :obj:'prec': A list of arrays. :obj:'prec[l]' is precision \
123             for class :math:'l'. If class :math:'l' does not exist in \
124             either :obj:'pred_labels' or :obj:'gt_labels', :obj:'prec[l]' is \
125             set to :obj:'None'.
126         * :obj:'rec': A list of arrays. :obj:'rec[l]' is recall \
127             for class :math:'l'. If class :math:'l' that is not marked as \
128             difficult does not exist in \
129             :obj:'gt_labels', :obj:'rec[l]' is \
```

```

130         set to :obj:'None'.
131     """
132
133     pred_bboxes = iter(pred_bboxes)
134     pred_labels = iter(pred_labels)
135     pred_scores = iter(pred_scores)
136     gt_bboxes = iter(gt_bboxes)
137     gt_labels = iter(gt_labels)
138     if gt_difficults is None:
139         gt_difficults = itertools.repeat(None)
140     else:
141         gt_difficults = iter(gt_difficults)
142
143     n_pos = defaultdict(int)
144     score = defaultdict(list)
145     match = defaultdict(list)
146
147     for pred_bbox, pred_label, pred_score, gt_bbox, gt_label, gt_difficult in \
148         six.moves.zip(
149             pred_bboxes, pred_labels, pred_scores,
150             gt_bboxes, gt_labels, gt_difficults):
151
152         if gt_difficult is None:
153             gt_difficult = np.zeros(gt_bbox.shape[0], dtype=bool)
154
155         for l in np.unique(np.concatenate((pred_label, gt_label)).astype(int)):
156             pred_mask_l = pred_label == l
157             pred_bbox_l = pred_bbox[pred_mask_l]
158             pred_score_l = pred_score[pred_mask_l]
159             # sort by score
160             order = pred_score_l.argsort()[::-1]
161             pred_bbox_l = pred_bbox_l[order]
162             pred_score_l = pred_score_l[order]
163
164             gt_mask_l = gt_label == l
165             gt_bbox_l = gt_bbox[gt_mask_l]
166             gt_difficult_l = gt_difficult[gt_mask_l]
167
168             n_pos[l] += np.logical_not(gt_difficult_l).sum()
169             score[l].extend(pred_score_l)
170
171             if len(pred_bbox_l) == 0:
172                 continue
173             if len(gt_bbox_l) == 0:
174                 match[l].extend((0,) * pred_bbox_l.shape[0])
175                 continue
176
177             # VOC evaluation follows integer typed bounding boxes.
178             pred_bbox_l = pred_bbox_l.copy()
179             pred_bbox_l[:, 2:] += 1
180             gt_bbox_l = gt_bbox_l.copy()
181             gt_bbox_l[:, 2:] += 1
182
183             iou = bbox_iou(pred_bbox_l, gt_bbox_l)
184             gt_index = iou.argmax(axis=1)
185             # set -1 if there is no matching ground truth
186             gt_index[iou.max(axis=1) < iou_thresh] = -1
187             del iou
188
189             selec = np.zeros(gt_bbox_l.shape[0], dtype=bool)
190             for gt_idx in gt_index:
191                 if gt_idx >= 0:
192                     if gt_difficult_l[gt_idx]:
193                         match[l].append(-1)
194                     else:
195                         if not selec[gt_idx]:
196                             match[l].append(1)
197                         else:
198                             match[l].append(0)
199                         selec[gt_idx] = True
200                 else:
201                     match[l].append(0)
202
203     for iter_ in (
204         pred_bboxes, pred_labels, pred_scores,
205         gt_bboxes, gt_labels, gt_difficults):
206         if next(iter_, None) is not None:
207             raise ValueError('Length of input iterables need to be same.')
208
209     n_fg_class = max(n_pos.keys()) + 1
210     prec = [None] * n_fg_class
211     rec = [None] * n_fg_class
212
213     for l in n_pos.keys():
214         score_l = np.array(score[l])
215         match_l = np.array(match[l], dtype=np.int8)
216

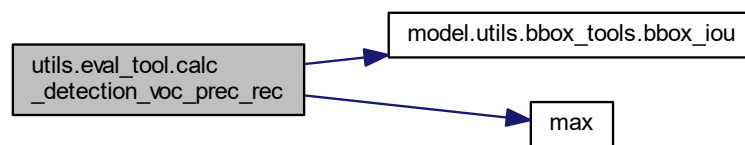
```

```

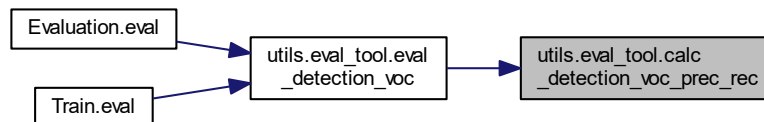
217         order = score_l.argsort()[::-1]
218         match_l = match_l[order]
219
220         tp = np.cumsum(match_l == 1)
221         fp = np.cumsum(match_l == 0)
222
223         # If an element of fp + tp is 0,
224         # the corresponding element of prec[l] is nan.
225         prec[l] = tp / (fp + tp)
226         # If n_pos[l] is 0, rec[l] is None.
227         if n_pos[l] > 0:
228             rec[l] = tp / n_pos[l]
229
230     return prec, rec
231
232

```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



이 함수를 호출하는 함수들에 대한 그래프입니다.:



7.38.1.3 eval_detection_voc()

```

def utils.eval_tool.eval_detection_voc (
    pred_bboxes,
    pred_labels,
    pred_scores,
    gt_bboxes,
    gt_labels,
    gt_difficults = None,
    iou_thresh = 0.5,
    use_07_metric = False )

```

Calculate average precisions based on evaluation code of PASCAL VOC.
 This function evaluates predicted bounding boxes obtained from a dataset
 which has :math:'N' images by using average precision for each class.
 The code is based on the evaluation code used in PASCAL VOC Challenge.
 Args:

`pred_bboxes` (iterable of `numpy.ndarray`): An iterable of :math:'N'
 sets of bounding boxes.
 Its index corresponds to an index for the base dataset.
 Each element of :obj:'pred_bboxes' is a set of coordinates
 of bounding boxes. This is an array whose shape is :math:'(R, 4)',
 where :math:'R' corresponds
 to the number of bounding boxes, which may vary among boxes.
 The second axis corresponds to
 :math:'y_{\min}, x_{\min}, y_{\max}, x_{\max}' of a bounding box.
`pred_labels` (iterable of `numpy.ndarray`): An iterable of labels.
 Similar to :obj:'pred_bboxes', its index corresponds to an
 index for the base dataset. Its length is :math:'N'.
`pred_scores` (iterable of `numpy.ndarray`): An iterable of confidence
 scores for predicted bounding boxes. Similar to :obj:'pred_bboxes',
 its index corresponds to an index for the base dataset.
 Its length is :math:'N'.
`gt_bboxes` (iterable of `numpy.ndarray`): An iterable of ground truth
 bounding boxes
 whose length is :math:'N'. An element of :obj:'gt_bboxes' is a
 bounding box whose shape is :math:'(R, 4)'. Note that the number of
 bounding boxes in each image does not need to be same as the number
 of corresponding predicted boxes.
`gt_labels` (iterable of `numpy.ndarray`): An iterable of ground truth
 labels which are organized similarly to :obj:'gt_bboxes'.
`gt_difficults` (iterable of `numpy.ndarray`): An iterable of boolean
 arrays which is organized similarly to :obj:'gt_bboxes'.
 This tells whether the
 corresponding ground truth bounding box is difficult or not.
 By default, this is :obj:'None'. In that case, this function
 considers all bounding boxes to be not difficult.
`iou_thresh` (float): A prediction is correct if its Intersection over
 Union with the ground truth is above this value.
`use_07_metric` (bool): Whether to use PASCAL VOC 2007 evaluation metric
 for calculating average precision. The default value is
 :obj:'False'.

Returns:

dict:
 The keys, value-types and the description of the values are listed
 below.
 * **ap** (*`numpy.ndarray`): An array of average precisions. \
 The :math:'l'-th value corresponds to the average precision \
 for class :math:'l'. If class :math:'l' does not exist in \
 either :obj:'pred_labels' or :obj:'gt_labels', the corresponding \
 value is set to :obj:'numpy.nan'.
 * **map** (*float*): The average of Average Precisions over classes.

`eval_tool.py` 파일의 11 번째 라인에서 정의되었습니다.

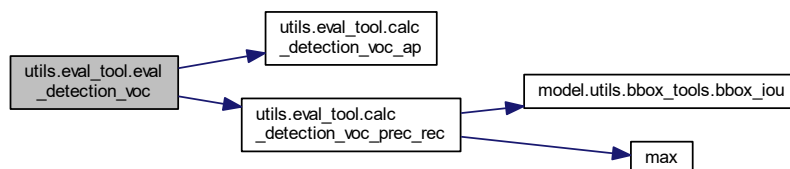
```
11 def eval_detection_voc(
12     pred_bboxes, pred_labels, pred_scores, gt_bboxes, gt_labels,
13     gt_difficults=None,
14     iou_thresh=0.5, use_07_metric=False):
15     """Calculate average precisions based on evaluation code of PASCAL VOC.
16     This function evaluates predicted bounding boxes obtained from a dataset
17     which has :math:'N' images by using average precision for each class.
18     The code is based on the evaluation code used in PASCAL VOC Challenge.
19     Args:
20         pred_bboxes (iterable of numpy.ndarray): An iterable of :math:'N'
21             sets of bounding boxes.
22             Its index corresponds to an index for the base dataset.
23             Each element of :obj:'pred_bboxes' is a set of coordinates
24             of bounding boxes. This is an array whose shape is :math:'(R, 4)',
25             where :math:'R' corresponds
26             to the number of bounding boxes, which may vary among boxes.
27             The second axis corresponds to
28             :math:'y_{\min}, x_{\min}, y_{\max}, x_{\max}' of a bounding box.
29         pred_labels (iterable of numpy.ndarray): An iterable of labels.
30             Similar to :obj:'pred_bboxes', its index corresponds to an
31             index for the base dataset. Its length is :math:'N'.
32         pred_scores (iterable of numpy.ndarray): An iterable of confidence
33             scores for predicted bounding boxes. Similar to :obj:'pred_bboxes',
34             its index corresponds to an index for the base dataset.
```

```

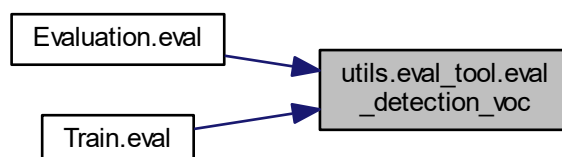
35     Its length is :math:'N'.
36     gt_bboxes (iterable of numpy.ndarray): An iterable of ground truth
37     bounding boxes
38     whose length is :math:'N'. An element of :obj:'gt_bboxes' is a
39     bounding box whose shape is :math:'(R, 4)'. Note that the number of
40     bounding boxes in each image does not need to be same as the number
41     of corresponding predicted boxes.
42     gt_labels (iterable of numpy.ndarray): An iterable of ground truth
43     labels which are organized similarly to :obj:'gt_bboxes'.
44     gt_difficults (iterable of numpy.ndarray): An iterable of boolean
45     arrays which is organized similarly to :obj:'gt_bboxes'.
46     This tells whether the
47     corresponding ground truth bounding box is difficult or not.
48     By default, this is :obj:'None'. In that case, this function
49     considers all bounding boxes to be not difficult.
50     iou_thresh (float): A prediction is correct if its Intersection over
51     Union with the ground truth is above this value.
52     use_07_metric (bool): Whether to use PASCAL VOC 2007 evaluation metric
53     for calculating average precision. The default value is
54     :obj:'False'.
55 Returns:
56     dict:
57     The keys, value-types and the description of the values are listed
58     below.
59     * **ap** (*numpy.ndarray*): An array of average precisions. \
60     The :math:'l'-th value corresponds to the average precision \
61     for class :math:'l'. If class :math:'l' does not exist in \
62     either :obj:'pred_labels' or :obj:'gt_labels', the corresponding \
63     value is set to :obj:'numpy.nan'.
64     * **map** (*float*): The average of Average Precisions over classes.
65 """
66
67 prec, rec = calc_detection_voc_prec_rec(
68     pred_bboxes, pred_labels, pred_scores,
69     gt_bboxes, gt_labels, gt_difficults,
70     iou_thresh=iou_thresh)
71
72 ap = calc_detection_voc_ap(prec, rec, use_07_metric=use_07_metric)
73
74 return {'ap': ap, 'map': np.nanmean(ap)}
75
76

```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



이 함수를 호출하는 함수들에 대한 그래프입니다.:



7.39 utils.py_nms 네임스페이스 참조

함수

- `def py_cpu_nms (rois, thresh)`
Non-Maximum Suppression(NMS): ground truth box와 IoU가 0.7이상 겹치는 anchor 값이 많이 나오기 때문에 하나만 남기고 나머지는 지운다.

7.39.1 함수 문서화

7.39.1.1 py_cpu_nms()

```
def utils.py_nms.py_cpu_nms (
    rois,
    thresh )
```

Non-Maximum Suppression(NMS): ground truth box와 IoU가 0.7이상 겹치는 anchor 값이 많이 나오기 때문에 하나만 남기고 나머지는 지운다.

매개변수

rois	object가 존재할 상위 12000개의 예측된 anchor 좌표
thresh	0.7

반환값

keep : 12000개의 anchor box 중 IoU가 0.7이상인 index return

Pure Python NMS baseline.
Already Sorted

return:
keep: roi keep indice

py_nms.py 파일의 9 번째 라인에서 정의되었습니다.

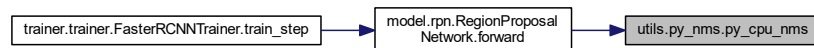
```
9 def py_cpu_nms(rois, thresh):
10     """
11     Pure Python NMS baseline.
12     Already Sorted
13
14     return:
15     keep: roi keep indice
16     """
17     y1 = rois[:, 0]
18     x1 = rois[:, 1]
19     y2 = rois[:, 2]
20     x2 = rois[:, 3]
21     areas = (x2 - x1 + 1) * (y2 - y1 + 1)
22
23     N = len(rois)
24     order = np.array(range(N))
25
26     keep = []
27     while order.size > 0:
28         i = order[0]
29         # 겹치는 anchor 중 첫번째 anchor만 저장
```

```

30         keep.append(i)
31
32         # x1 기준으로 x1좌표가 더 큰 좌표만 계산한다.(a,b의 계산이 b,a에서 다시 계산되는 것 방지)
33         xx1 = np.maximum(x1[i], x1[order[1:]])
34         yy1 = np.maximum(y1[i], y1[order[1:]])
35         xx2 = np.minimum(x2[i], x2[order[1:]])
36         yy2 = np.minimum(y2[i], y2[order[1:]])
37
38         # x1,x2(y1, y2)가 서로 역전되지 않는 경우를 생각한다.
39         w = np.maximum(0.0, xx2 - xx1 + 1)
40         h = np.maximum(0.0, yy2 - yy1 + 1)
41         inter = w * h
42         # ovr: anchor끼리의 IoU
43         ovr = inter / (areas[i] + areas[order[1:]] - inter)
44
45         inds = np.where(ovr <= thresh)[0]
46         order = order[inds + 1]
47
48     return keep

```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



7.40 utils.vis_tool 네임스페이스 참조

클래스

- class [Visualizer](#)

함수

- def [vis_image](#) (img, ax=None)
- def [vis_bbox](#) (img, [bbox](#), [label](#)=None, score=None, ax=None)
- def [fig2data](#) (fig)
- def [fig4vis](#) (fig)
- def [visdom_bbox](#) (*args, **kwargs)

변수

- tuple [VOC_BBOX_LABEL_NAMES](#)

7.40.1 함수 문서화

7.40.1.1 fig2data()

```
def utils.vis_tool.fig2data (
    fig )
```

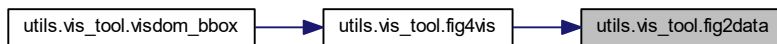
brief Convert a Matplotlib figure to a 4D numpy array with RGBA channels and return it

@param fig: a matplotlib figure
@return a numpy 3D array of RGBA values

vis_tool.py 파일의 115 번째 라인에서 정의되었습니다.

```
115 def fig2data(fig):
116     """
117     brief Convert a Matplotlib figure to a 4D numpy array with RGBA
118     channels and return it
119
120     @param fig: a matplotlib figure
121     @return a numpy 3D array of RGBA values
122     """
123     # draw the renderer
124     fig.canvas.draw()
125
126     # Get the RGBA buffer from the figure
127     w, h = fig.canvas.get_width_height()
128     buf = np.fromstring(fig.canvas.tostring_argb(), dtype=np.uint8)
129     buf.shape = (w, h, 4)
130
131     # canvas.tostring_argb give pixmap in ARGB mode. Roll the ALPHA channel to have it in RGBA mode
132     buf = np.roll(buf, 3, axis=2)
133     return buf.reshape(h, w, 4)
134
135
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



7.40.1.2 fig4vis()

```
def utils.vis_tool.fig4vis (
    fig )
```

convert figure to ndarray

vis_tool.py 파일의 136 번째 라인에서 정의되었습니다.

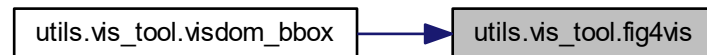
```
136 def fig4vis(fig):
137     """
138     convert figure to ndarray
139     """
140     ax = fig.get_figure()
141     img_data = fig2data(ax).astype(np.int32)
142     plot.close()
143     # HWC->CHW
144     return img_data[:, :, :3].transpose((2, 0, 1)) / 255.
145
```

146

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



이 함수를 호출하는 함수들에 대한 그래프입니다.:



7.40.1.3 vis_bbox()

```
def utils.vis_tool.vis_bbox (
    img,
    bbox,
    label = None,
    score = None,
    ax = None )
```

Visualize bounding boxes inside image.

Args:

`img` (~numpy.ndarray): An array of shape :math:`(3, \text{height}, \text{width})`.
This is in RGB format and the range of its value is :math:`[0, 255]`.

`bbox` (~numpy.ndarray): An array of shape :math:`(R, 4)` , where :math:`R` is the number of bounding boxes in the image.
Each element is organized by :math:`(y_{\min}, x_{\min}, y_{\max}, x_{\max})` in the second axis.

`label` (~numpy.ndarray): An integer array of shape :math:`(R,)` .
The values correspond to id for label names stored in :code:`label_names` . This is optional.

`score` (~numpy.ndarray): A float array of shape :math:`(R,)` .
Each value indicates how confident the prediction is.
This is optional.

`label_names` (iterable of strings): Name of labels ordered according to label ids. If this is :code:`None` , labels will be skipped.

`ax` (matplotlib.axes.Axis): The visualization is displayed on this axis. If this is :code:`None` (default), a new axis is created.

Returns:

~matplotlib.axes.Axes:
Returns the Axes object with the plot for further tweaking.

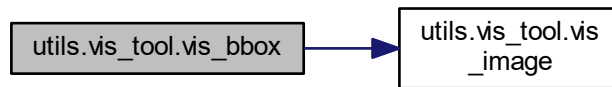
vis_tool.py 파일의 46 번째 라인에서 정의되었습니다.

```

46 def vis_bbox(img, bbox, label=None, score=None, ax=None):
47     """Visualize bounding boxes inside image.
48
49     Args:
50         img ( numpy.ndarray ): An array of shape :math:`(3, height, width)`'.
51         This is in RGB format and the range of its value is
52         :math:`[0, 255]`'.
53         bbox ( numpy.ndarray ): An array of shape :math:`(R, 4)`', where
54         :math:`R`' is the number of bounding boxes in the image.
55         Each element is organized
56         by :math:`(y_{min}, x_{min}, y_{max}, x_{max})`' in the second axis.
57         label ( numpy.ndarray ): An integer array of shape :math:`(R,)`'`.
58         The values correspond to id for label names stored in
59         :obj:`label_names`. This is optional.
60         score ( numpy.ndarray ): A float array of shape :math:`(R,)`'`.
61         Each value indicates how confident the prediction is.
62         This is optional.
63         label_names ( iterable of strings ): Name of labels ordered according
64         to label ids. If this is :obj:`None`, labels will be skipped.
65         ax ( matplotlib.axes.Axes ): The visualization is displayed on this
66         axis. If this is :obj:`None` (default), a new axis is created.
67
68     Returns:
69         matplotlib.axes.Axes:
70         Returns the Axes object with the plot for further tweaking.
71
72     """
73
74     label_names = list(VOC_BBOX_LABEL_NAMES) + ['bg']
75     # add for index '-1'
76     if label is not None and not len(bbox) == len(label):
77         raise ValueError('The length of label must be same as that of bbox')
78     if score is not None and not len(bbox) == len(score):
79         raise ValueError('The length of score must be same as that of bbox')
80
81     # Returns newly instantiated matplotlib.axes.Axes object if ax is None
82     ax = vis_image(img, ax=ax)
83
84     # If there is no bounding box to display, visualize the image and exit.
85     if len(bbox) == 0:
86         return ax
87
88     for i, bb in enumerate(bbox):
89         xy = (bb[1], bb[0])
90         height = bb[2] - bb[0]
91         width = bb[3] - bb[1]
92         ax.add_patch(plot.Rectangle(
93             xy, width, height, fill=False, edgecolor='red', linewidth=2))
94
95         caption = list()
96
97         if label is not None and label_names is not None:
98             lb = label[i]
99             if not (-1 <= lb < len(label_names)): # modify here to add background
100                 raise ValueError('No corresponding name is given')
101             caption.append(label_names[lb])
102         if score is not None:
103             sc = score[i]
104             caption.append('{:.2f}'.format(sc))
105
106         if len(caption) > 0:
107             ax.text(bb[1], bb[0],
108                     ': '.join(caption),
109                     style='italic',
110                     bbox={'facecolor': 'white', 'alpha': 0.5, 'pad': 0})
111
112     return ax
113
114

```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



이 함수를 호출하는 함수들에 대한 그래프입니다.:



7.40.1.4 vis_image()

```
def utils.vis_tool.vis_image (
    img,
    ax = None )
```

Visualize a color image.

Args:

`img` (~numpy.ndarray): An array of shape `:math:`(3, height, width)``. This is in RGB format and the range of its value is `:math:`[0, 255]``.
`ax` (matplotlib.axes.Axis): The visualization is displayed on this axis. If this is `:obj:`None`` (default), a new axis is created.

Returns:

~matplotlib.axes.Axes:
 Returns the Axes object with the plot for further tweaking.

vis_tool.py 파일의 21 번째 라인에서 정의되었습니다.

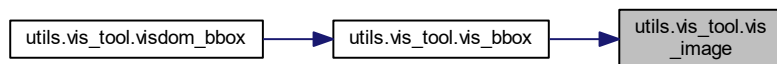
```
21 def vis_image(img, ax=None):
22     """Visualize a color image.
23
24     Args:
25         img ( numpy.ndarray): An array of shape :math:`(3, height, width)` .
26         This is in RGB format and the range of its value is
27         :math:`[0, 255]` .
28         ax (matplotlib.axes.Axis): The visualization is displayed on this
29         axis. If this is :obj:`None` (default), a new axis is created.
30
31     Returns:
32         matplotlib.axes.Axes:
```

```

33     Returns the Axes object with the plot for further tweaking.
34
35     """
36
37     if ax is None:
38         fig = plot.figure()
39         ax = fig.add_subplot(1, 1, 1)
40     # CHW -> HWC
41     img = img.transpose((1, 2, 0))
42     ax.imshow(img.astype(np.uint8))
43     return ax
44
45

```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



7.40.1.5 visdom_bbox()

```

def utils.vis_tool.visdom_bbox (
    * args,
    ** kwargs )

```

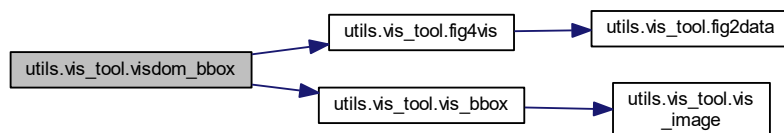
vis_tool.py 파일의 147 번째 라인에서 정의되었습니다.

```

147 def visdom_bbox(*args, **kwargs):
148     fig = vis_bbox(*args, **kwargs)
149     data = fig4vis(fig)
150     return data
151
152

```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



7.40.2 변수 문서화

7.40.2.1 VOC_BBOX_LABEL_NAMES

tuple utils.vis_tool.VOC_BBOX_LABEL_NAMES

초기값:

```
1 = (  
2     'do_not_wash',  
3     'do_not_bleach',  
4     'do_not_iron',  
5     'do_not_dry_clean'  
6 )
```

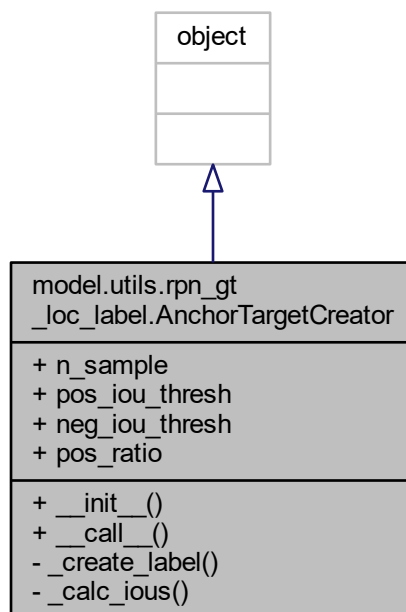
vis_tool.py 파일의 13 번째 라인에서 정의되었습니다.

Chapter 8

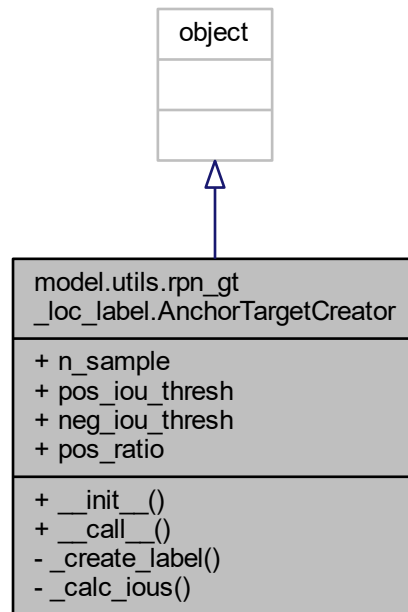
클래스 문서화

8.1 model.utils.rpn_gt_loc_label.AnchorTargetCreator 클래스 참조

model.utils.rpn_gt_loc_label.AnchorTargetCreator에 대한 상속 다이어그램 :



model.utils.rpn_gt_loc_label.AnchorTargetCreator에 대한 협력 다이어그램:



Public 멤버 함수

- def `__init__` (self, `n_sample`=256, `pos_iou_thresh`=0.7, `neg_iou_thresh`=0.3, `pos_ratio`=0.5)
- def `__call__` (self, `bbox`, anchor, `img_size`)

RPN을 통해서 예측한

Public 속성

- `n_sample`
- `pos_iou_thresh`
- `neg_iou_thresh`
- `pos_ratio`

Private 멤버 함수

- def `_create_label` (self, `inside_index`, anchor, `bbox`)
 - 1.
- def `_calc_iious` (self, anchor, `bbox`, `inside_index`)

8.1.1 상세한 설명

Assign the ground truth bounding boxes to anchors.

Assigns the ground truth bounding boxes to anchors for training Region Proposal Networks introduced in Faster R-CNN [#]_.

Offsets and scales to match anchors to the ground truth are calculated using the encoding scheme of
:func:`model.utils.bbox_tools.bbox2loc`.

.. [#] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun. \ Faster R-CNN: Towards Real-Time Object Detection with \ Region Proposal Networks. NIPS 2015.

```
@param    n_sample (int): The number of regions to produce.

@param    pos_iou_thresh (float): Anchors with IoU above this threshold will be assigned as positive.

@param    neg_iou_thresh (float): Anchors with IoU below this threshold will be assigned as negative.

@param    pos_ratio (float): Ratio of positive regions in the sampled regions.
```

rpn_gt_loc_label.py 파일의 5 번째 라인에서 정의되었습니다.

8.1.2 생성자 & 소멸자 문서화

8.1.2.1 __init__()

```
def model.utils.rpn_gt_loc_label.AnchorTargetCreator.__init__(
    self,
    n_sample = 256,
    pos_iou_thresh = 0.7,
    neg_iou_thresh = 0.3,
    pos_ratio = 0.5 )
```

rpn_gt_loc_label.py 파일의 30 번째 라인에서 정의되었습니다.

```
30 def __init__(self,
31             n_sample=256,
32             pos_iou_thresh=0.7, neg_iou_thresh=0.3,
33             pos_ratio=0.5):
34     self.n_sample = n_sample
35     self.pos_iou_thresh = pos_iou_thresh
36     self.neg_iou_thresh = neg_iou_thresh
37     self.pos_ratio = pos_ratio
38
```

8.1.3 멤버 함수 문서화

8.1.3.1 __call__()

```
def model.utils.rpn_gt_loc_label.AnchorTargetCreator.__call__(
    self,
    bbox,
    anchor,
    img_size )
```

RPN을 통해서 예측한

매개변수

bbox	B.B GT
anchor	F.M 한 픽셀당 9개의 anchor 좌표 초기값, shape = (F.M H * W * A, 4)
img_size	이미지 크기

반환값

loc : 전체 anchor 좌표와 B.B와의 상대좌표를 기록한 배열 --> shape = (F.M H * W * A, 4) 이지만
안에는 값은 inside_index 안에만 기록되어 있음

label : anchor에 대한 label을 기록한 배열 --> shape = (F.M H * W * A, 4)

Assign ground truth supervision to sampled subset of anchors.

Types of input arrays and output arrays are same.

Here are notations.

* :math:`S` is the number of anchors.
* :math:`R` is the number of bounding boxes.

Args:

bbox (array): Coordinates of bounding boxes. Its shape is :math:`(R, 4)`.
anchor (array): Coordinates of anchors. Its shape is :math:`(S, 4)`.
img_size (tuple of ints): A tuple :obj:`H, W`, which is a tuple of height and width of an image.

Returns:

(array, array):

#NOTE: it's scale not only offset
* **loc**: Offsets and scales to match the anchors to \ the ground truth bounding boxes. Its shape is :math:`(S, 4)`.
* **label**: Labels of anchors with values \ :obj:`(l=positive, 0=negative, -1=ignore)`. Its shape \ is :math:`(S,)`.

rpn_gt_loc_label.py 파일의 47 번째 라인에서 정의되었습니다.

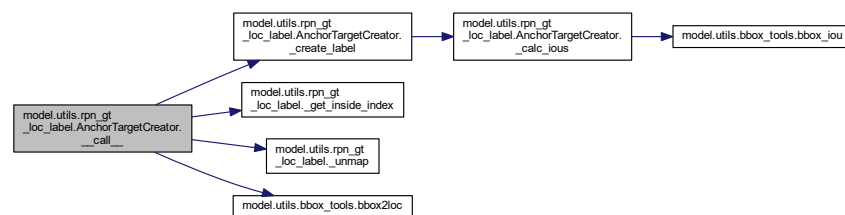
```
47 def __call__(self, bbox, anchor, img_size):
48     """Assign ground truth supervision to sampled subset of anchors.
49
50     Types of input arrays and output arrays are same.
51
52     Here are notations.
53
54     * :math:`S` is the number of anchors.
55     * :math:`R` is the number of bounding boxes.
56
57     Args:
58         bbox (array): Coordinates of bounding boxes. Its shape is
59             :math:`(R, 4)`'.
60         anchor (array): Coordinates of anchors. Its shape is
61             :math:`(S, 4)`'.
62         img_size (tuple of ints): A tuple :obj:`H, W`, which
63             is a tuple of height and width of an image.
64
65     Returns:
66         (array, array):
67
68         #NOTE: it's scale not only offset
69         * **loc**: Offsets and scales to match the anchors to \
70             the ground truth bounding boxes. Its shape is :math:`(S, 4)`'.
71         * **label**: Labels of anchors with values \
72             :obj:`(l=positive, 0=negative, -1=ignore)`. Its shape \
73             is :math:`(S,)`'.
74
75     """
76
77     img_H, img_W = img_size
```

```

78
79     # 전체 base anchor 사이즈
80     n_anchor = len(anchor)
81     inside_index = _get_inside_index(anchor, img_H, img_W)
82     anchor = anchor[inside_index]
83     argmax_iious, label = self._create_label(
84         inside_index, anchor, bbox)
85     # compute bounding box regression targets
86     # anchor 와 실제 B.B 의 위치와의 차이를 나타낸다. Y,X, height, width 로 구성되었음
87     loc = bbox2loc(anchor, bbox[argmax_ious])
88
89     # map up to original set of anchors
90     # labeling을 한 index의 label값만 가져온다.
91     label = _unmap(label, n_anchor, inside_index, fill=-1)
92     # 위에서 선택된 anchor들에서 실제 B.B.와의 상대적인 거리를 loc에 저장. 배열은 전체 base anchor size 만큼 설정, inside_index 만 값을 가진다.
93     loc = _unmap(loc, n_anchor, inside_index, fill=0)
94     return loc, label
95

```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



8.1.3.2 _calc_iious()

```

def model.utils.rpn_gt_loc_label.AnchorTargetCreator._calc_iious (
    self,
    anchor,
    bbox,
    inside_index ) [private]

```

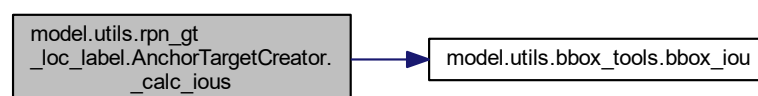
rpn_gt_loc_label.py 파일의 143 번째 라인에서 정의되었습니다.

```

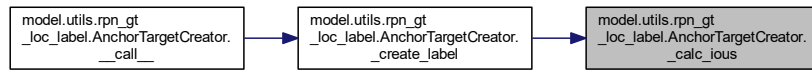
143 def _calc_iious(self, anchor, bbox, inside_index):
144     # ious between the anchors and the gt boxes
145     ious = bbox_iou(anchor, bbox)
146     argmax_iious = ious.argmax(axis=1)
147     max_iious = ious[np.arange(len(inside_index)), argmax_iious]
148     gt_argmax_iious = ious.argmax(axis=0)
149     gt_max_iious = ious[gt_argmax_iious, np.arange(ious.shape[1])]
150     gt_argmax_iious = np.where(ious == gt_max_iious)[0]
151
152     return argmax_iious, max_iious, gt_argmax_iious
153
154

```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



이 함수를 호출하는 함수들에 대한 그래프입니다.:



8.1.3.3 _create_label()

```
def model.utils.rpn_gt_loc_label.AnchorTargetCreator._create_label (
    self,
    inside_index,
    anchor,
    bbox ) [private]
```

- 1.

각 anchor마다 최대 iou가 0.3 미만이면 해당 anchor의 label은 0이다.

- 2. 각 anchor마다 최대 iou가 0.7 이상이면 해당 anchor의 label은 1이다.
- 3. 각 gt box마다 최대 iou를 가지는 anchor의 label은 1이다.
- (anchor label = 0: background, 1: object, -1: 애매하므로 무시)

매개변수

inside_index	이미지 size 안에 존재하는 anchor index
anchor	anchor box
bbox	B.B GT

반환값

argmax_ious anchor 박스에 대한 IoU
label argmax_ious에 대한 labeling 값

rpn_gt_loc_label.py 파일의 107 번째 라인에서 정의되었습니다.

```

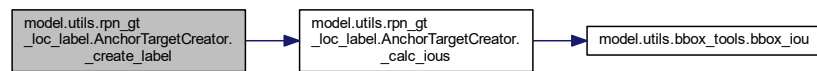
107 def _create_label(self, inside_index, anchor, bbox):
108     # label: 1 is positive, 0 is negative, -1 is dont care
109     label = np.empty((len(inside_index)), dtype=np.int32)
110     label.fill(-1)
111
112     argmax_ious, max_ious, gt_argmax_ious = \
113         self._calc_ious(anchor, bbox, inside_index)
114
115     # assign negative labels first so that positive labels can clobber them
116     label[max_ious < self.neg_iou_thresh] = 0
117
118     # positive label: for each gt, anchor with highest iou
119     label[gt_argmax_ious] = 1
120
```

```

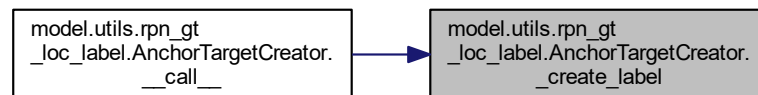
121         # positive label: above threshold IOU
122         label[max_ioues >= self.pos_iou_thresh] = 1
123
124         # subsample positive labels if we have too many
125         n_pos = int(self.pos_ratio * self.n_sample)
126         pos_index = np.where(label == 1)[0]
127         if len(pos_index) > n_pos:
128             disable_index = np.random.choice(
129                 pos_index, size=(len(pos_index) - n_pos), replace=False)
130             label[disable_index] = -1
131
132         # subsample negative labels if we have too many
133         n_neg = self.n_sample - np.sum(label == 1)
134         neg_index = np.where(label == 0)[0]
135         if len(neg_index) > n_neg:
136             disable_index = np.random.choice(
137                 neg_index, size=(len(neg_index) - n_neg), replace=False)
138             label[disable_index] = -1
139
140         return argmax_ioues, label
141

```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



이 함수를 호출하는 함수들에 대한 그래프입니다.:



8.1.4 멤버 데이터 문서화

8.1.4.1 n_sample

model.utils.rpn_gt_loc_label.AnchorTargetCreator.n_sample

rpn_gt_loc_label.py 파일의 31 번째 라인에서 정의되었습니다.

8.1.4.2 neg_iou_thresh

model.utils.rpn_gt_loc_label.AnchorTargetCreator.neg_iou_thresh

rpn_gt_loc_label.py 파일의 33 번째 라인에서 정의되었습니다.

8.1.4.3 pos_iou_thresh

model.utils.rpn_gt_loc_label.AnchorTargetCreator.pos_iou_thresh

rpn_gt_loc_label.py 파일의 32 번째 라인에서 정의되었습니다.

8.1.4.4 pos_ratio

model.utils.rpn_gt_loc_label.AnchorTargetCreator.pos_ratio

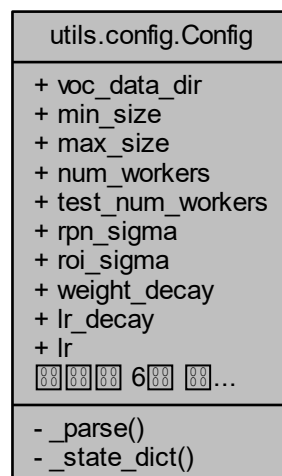
rpn_gt_loc_label.py 파일의 34 번째 라인에서 정의되었습니다.

이 클래스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- model/utils/[rpn_gt_loc_label.py](#)

8.2 utils.config.Config 클래스 참조

utils.config.Config에 대한 협력 다이어그램:



정적 Public 속성

- string `voc_data_dir` = './data/VOCdevkit/VOC2007'
- int `min_size` = 600
- int `max_size` = 1000
- int `num_workers` = 4
- int `test_num_workers` = 4
- int `rpn_sigma` = 3.
- int `roi_sigma` = 1.
- float `weight_decay` = 0.0005
- float `lr_decay` = 0.1
- int `lr` = 1e-3
- int `epoch` = 20
- bool `use_adam` = False
- int `test_num` = 1000
- string `load_path` = 'C:/Users/user/PycharmProjects/Faster-RCNN-Pytorch_6/checkpoints/faster_rcnn_11072246_7.pth'
- bool `caffe_pretrain` = False
- string `caffe_pretrain_path` = 'checkpoints/vgg16-caffe.pth'

Private 멤버 함수

- def `__parse` (self, kwargs)
- def `__state_dict` (self)

8.2.1 상세한 설명

`config.py` 파일의 4 번째 라인에서 정의되었습니다.

8.2.2 멤버 함수 문서화

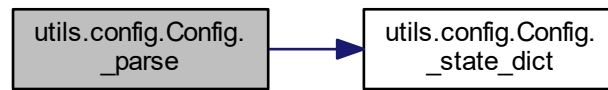
8.2.2.1 `__parse()`

```
def utils.config.Config.__parse (
    self,
    kwargs ) [private]
```

`config.py` 파일의 51 번째 라인에서 정의되었습니다.

```
51     def __parse(self, kwargs):
52         state_dict = self._state_dict()
53         for k, v in kwargs.items():
54             if k not in state_dict:
55                 raise ValueError('Unknown Option: "%s"' % k)
56             setattr(self, k, v)
57
58         print('====user config====')
59         pprint(self._state_dict())
60         print('=====end=====')
61
```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



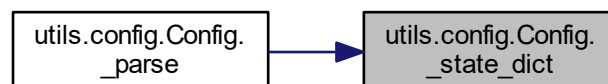
8.2.2.2 _state_dict()

```
def utils.config.Config._state_dict (
    self ) [private]
```

config.py 파일의 62 번째 라인에서 정의되었습니다.

```
62     def _state_dict(self):
63         return {k: getattr(self, k) for k, _ in Config.__dict__.items() \
64                 if not k.startswith('_')}
65
66
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



8.2.3 멤버 데이터 문서화

8.2.3.1 caffe_pretrain

```
bool utils.config.Config.caffe_pretrain = False [static]
```

config.py 파일의 48 번째 라인에서 정의되었습니다.

8.2.3.2 caffe_pretrain_path

```
string utils.config.Config.caffe_pretrain_path = 'checkpoints/vgg16-caffe.pth' [static]
```

config.py 파일의 49 번째 라인에서 정의되었습니다.

8.2.3.3 epoch

```
int utils.config.Config.epoch = 20 [static]
```

config.py 파일의 33 번째 라인에서 정의되었습니다.

8.2.3.4 load_path

```
string utils.config.Config.load_path = 'C:/Users/user/PycharmProjects/Faster-RCNN-Pytorch_6/checkpoints/fasterrcnn↵_11072246_7.pth' [static]
```

config.py 파일의 45 번째 라인에서 정의되었습니다.

8.2.3.5 lr

```
int utils.config.Config.lr = 1e-3 [static]
```

config.py 파일의 20 번째 라인에서 정의되었습니다.

8.2.3.6 lr_decay

```
float utils.config.Config.lr_decay = 0.1 [static]
```

config.py 파일의 19 번째 라인에서 정의되었습니다.

8.2.3.7 max_size

```
int utils.config.Config.max_size = 1000 [static]
```

config.py 파일의 8 번째 라인에서 정의되었습니다.

8.2.3.8 min_size

```
int utils.config.Config.min_size = 600 [static]
```

config.py 파일의 7 번째 라인에서 정의되었습니다.

8.2.3.9 num_workers

```
int utils.config.Config.num_workers = 4 [static]
```

config.py 파일의 9 번째 라인에서 정의되었습니다.

8.2.3.10 roi_sigma

```
int utils.config.Config.roi_sigma = 1. [static]
```

config.py 파일의 14 번째 라인에서 정의되었습니다.

8.2.3.11 rpn_sigma

```
int utils.config.Config.rpn_sigma = 3. [static]
```

config.py 파일의 13 번째 라인에서 정의되었습니다.

8.2.3.12 test_num

```
int utils.config.Config.test_num = 1000 [static]
```

config.py 파일의 42 번째 라인에서 정의되었습니다.

8.2.3.13 test_num_workers

```
int utils.config.Config.test_num_workers = 4 [static]
```

config.py 파일의 10 번째 라인에서 정의되었습니다.

8.2.3.14 use_adam

```
bool utils.config.Config.use_adam = False [static]
```

config.py 파일의 36 번째 라인에서 정의되었습니다.

8.2.3.15 voc_data_dir

```
string utils.config.Config.voc_data_dir = './data/VOCdevkit/VOC2007' [static]
```

config.py 파일의 6 번째 라인에서 정의되었습니다.

8.2.3.16 weight_decay

```
float utils.config.Config.weight_decay = 0.0005 [static]
```

config.py 파일의 18 번째 라인에서 정의되었습니다.

이 클래스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- [utils/config.py](#)

8.3 data.dataset.Dataset 클래스 참조

data.dataset.Dataset에 대한 협력 다이어그램:

data.dataset.Dataset
+ opt + db + tsf
+ __init__() + __getitem__() + __len__()

Public 멤버 함수

- `def __init__ (self, opt)`
- `def __getitem__ (self, idx)`
- `def __len__ (self)`

Public 속성

- `opt`
- `db`
- `tsf`

8.3.1 상세한 설명

dataset.py 파일의 153 번째 라인에서 정의되었습니다.

8.3.2 생성자 & 소멸자 문서화

8.3.2.1 __init__()

```
def data.dataset.Dataset.__init__ (
    self,
    opt )
```

dataset.py 파일의 154 번째 라인에서 정의되었습니다.

```
154     def __init__(self, opt):
155         self.opt = opt
156         self.db = VOCBboxDataset(opt.voc_data_dir)
157         self.tsf = Transform(opt.min_size, opt.max_size)
158
```

8.3.3 멤버 함수 문서화

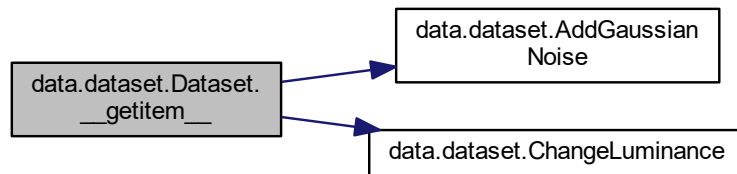
8.3.3.1 __getitem__()

```
def data.dataset.Dataset.__getitem__(
    self,
    idx )
```

dataset.py 파일의 159 번째 라인에서 정의되었습니다.

```
159 def __getitem__(self, idx):
160     # voc_dataset.py --> get_example
161     ori_img, bbox, label, difficult = self.db.get_example(idx)
162     # data augmentation
163     ori_img = AddGaussianNoise(ori_img, 0.05)
164     ori_img = ChangeLuminance(ori_img, 10, 20)
165
166     img, bbox, label, scale = self.tsf((ori_img, bbox, label))
167     # TODO: check whose stride is negative to fix this instead copy all
168     # some of the strides of a given numpy array are negative.
169     return img.copy(), bbox.copy(), label.copy(), scale
170
```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



8.3.3.2 __len__()

```
def data.dataset.Dataset.__len__(
    self )
```

dataset.py 파일의 171 번째 라인에서 정의되었습니다.

```
171 def __len__(self):
172     return len(self.db)
173
174
```

8.3.4 멤버 데이터 문서화

8.3.4.1 db

data.dataset.Dataset.db

dataset.py 파일의 156 번째 라인에서 정의되었습니다.

8.3.4.2 opt

`data.dataset.Dataset.opt`

`dataset.py` 파일의 155 번째 라인에서 정의되었습니다.

8.3.4.3 tsf

`data.dataset.Dataset.tsf`

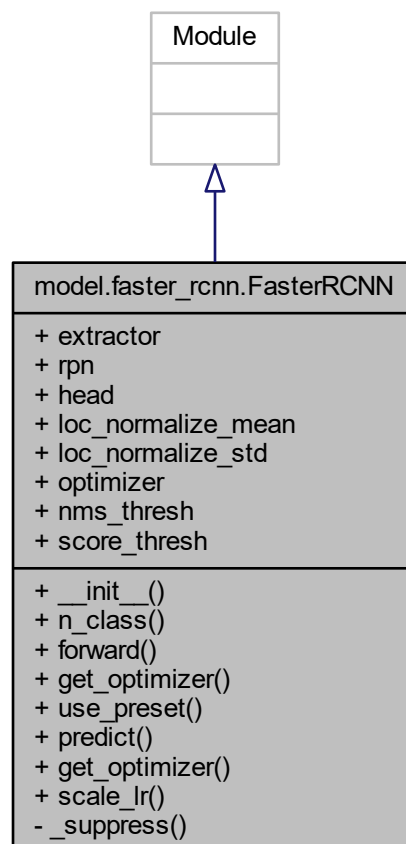
`dataset.py` 파일의 157 번째 라인에서 정의되었습니다.

이 클래스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

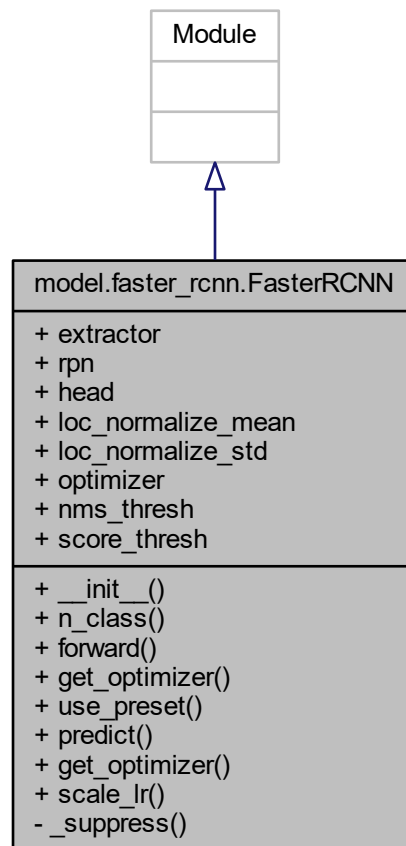
- `data/dataset.py`

8.4 model.faster_rcnn.FasterRCNN 클래스 참조

`model.faster_rcnn.FasterRCNN`에 대한 상속 다이어그램 :



model.faster_rcnn.FasterRCNN에 대한 협력 다이어그램:



Public 멤버 함수

- def `__init__` (self, ratios=[0.5, 1, 2], anchor_scales=[0.5, 1, 2], `loc_normalize_mean`=(0., 0., 0., 0.), `loc_normalize_std`=(0.1, 0.1, 0.2, 0.2))
model 정의 및 RPN & ROI network 초기화 설정
- def `n_class` (self)
- def `forward` (self, x, scale=1.)
- def `get_optimizer` (self)
- def `use_preset` (self, preset)
- def `predict` (self, imgs, sizes=None, visualize=False)
- def `get_optimizer` (self)
- def `scale_lr` (self, decay=0.1)

Public 속성

- `extractor`
- `rpn`

- `head`
- `loc_normalize_mean`
- `loc_normalize_std`
- `optimizer`
- `nms_thresh`
- `score_thresh`

Private 멤버 함수

- `def __suppress` (self, raw_cls_bbox, raw_prob)

8.4.1 상세한 설명

`faster_rcnn.py` 파일의 17 번째 라인에서 정의되었습니다.

8.4.2 생성자 & 소멸자 문서화

8.4.2.1 `__init__()`

```
def model.faster_rcnn.FasterRCNN.__init__(
    self,
    ratios = [0.5, 1, 2],
    anchor_scales = [0.5, 1, 2],
    loc_normalize_mean = (0., 0., 0., 0.),
    loc_normalize_std = (0.1, 0.1, 0.2, 0.2) )
```

model 정의 및 RPN & ROI network 초기화 설정

매개변수

<code>ratios</code>	: anchor 박스의 비율 설정
<code>anchor_scales</code>	: anchor 박스의 크기 설정
<code>loc_normalize_mean</code>	: (0., 0., 0., 0.)
<code>loc_normalize_std</code>	: (0.1, 0.1, 0.2, 0.2)

`faster_rcnn.py` 파일의 25 번째 라인에서 정의되었습니다.

```
25 def __init__(self, ratios=[0.5, 1, 2], anchor_scales=[0.5, 1, 2], \
26               loc_normalize_mean = (0., 0., 0., 0.), \
27               loc_normalize_std = (0.1, 0.1, 0.2, 0.2)):
28     super(FasterRCNN, self).__init__()
29
30     # prepare
31     # extractor & classifier type : <class 'torch.nn.modules.container.Sequential'>
32     # VGG16.py 에서 parameter 고정하기 & 모델 변경
33     extractor, classifier = decom_vgg16()
34
35     # 네트워크 변경시 네트워크 마지막 layer의 output size에 맞도록 숫자 변경(VGG 기준 512 * 512)
36     # todo: resnet50 & resnet101 적용
37     rpn = RegionProposalNetwork(
38         512, 512,
```

```

39         ratios=ratios,
40         scales=anchor_scales,
41         feat_stride=16
42     )
43
44     head = VGG16RoIHead(
45         n_class=4 + 1,
46         roi_size=7,
47         spatial_scale=(1. / 16),
48         classifier=classifier.cuda()
49     )
50     self.extractor = extractor.cuda()
51     self.rpn = rpn.cuda()
52     self.head = head
53
54     # mean and std
55     self.loc_normalize_mean = loc_normalize_mean
56     self.loc_normalize_std = loc_normalize_std
57
58

```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



8.4.3 멤버 함수 문서화

8.4.3.1 _suppress()

```

def model.faster_rcnn.FasterRCNN._suppress (
    self,
    raw_cls_bbox,
    raw_prob ) [private]

```

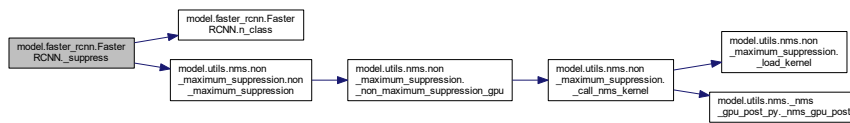
faster_rcnn.py 파일의 123 번째 라인에서 정의되었습니다.

```

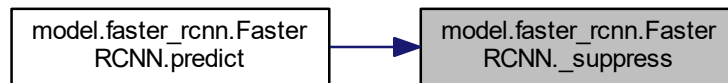
123 def _suppress(self, raw_cls_bbox, raw_prob):
124     bbox = list()
125     label = list()
126     score = list()
127     # skip cls_id = 0 because it is the background class
128     for l in range(1, self.n_class):
129         cls_bbox_l = raw_cls_bbox.reshape((-1, self.n_class, 4))[:, l, :]
130         prob_l = raw_prob[:, l]
131         mask = prob_l > self.score_thresh
132         cls_bbox_l = cls_bbox_l[mask]
133         prob_l = prob_l[mask]
134         keep = non_maximum_suppression(
135             cp.array(cls_bbox_l), self.nms_thresh, prob_l)
136         keep = cp.asnumpy(keep)
137         bbox.append(cls_bbox_l[keep])
138         # The labels are in [0, self.n_class - 2].
139         label.append((l - 1) * np.ones((len(keep)),))
140         score.append(prob_l[keep])
141     bbox = np.concatenate(bbox, axis=0).astype(np.float32)
142     label = np.concatenate(label, axis=0).astype(np.int32)
143     score = np.concatenate(score, axis=0).astype(np.float32)
144     return bbox, label, score
145

```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



이 함수를 호출하는 함수들에 대한 그래프입니다.:



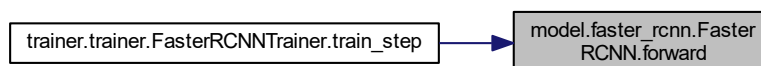
8.4.3.2 forward()

```
def model.faster_rcnn.FasterRCNN.forward (
    self,
    x,
    scale = 1. )
```

faster_rcnn.py 파일의 64 번째 라인에서 정의되었습니다.

```
64 def forward(self, x, scale=1.):
65
66     img_size = x.shape[2:]
67
68     h = self.extractor(x)
69     rpn_locs, rpn_scores, rois, roi_indices, anchor = \
70         self.rpn(h, img_size, scale)
71     roi_cls_locs, roi_scores = self.head(
72         h.cuda(), rois, np.array(roi_indices))
73     return roi_cls_locs, roi_scores, rois, roi_indices
74
75
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



8.4.3.3 get_optimizer() [1/2]

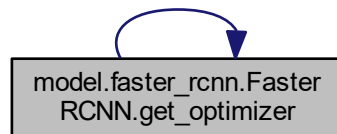
```
def model.faster_rcnn.FasterRCNN.get_optimizer (
    self )

return optimizer, It could be overwritten if you want to specify
special optimizer
```

faster_rcnn.py 파일의 76 번째 라인에서 정의되었습니다.

```
76 def get_optimizer(self):
77     """
78     return optimizer, It could be overwritten if you want to specify
79     special optimizer
80     """
81     lr = opt.lr
82     params = []
83     for key, value in dict(self.named_parameters()).items():
84         if value.requires_grad:
85             if 'bias' in key:
86                 params += [{'params': [value], 'lr': lr * 2, 'weight_decay': 0}]
87             else:
88                 params += [{'params': [value], 'lr': lr, 'weight_decay': opt.weight_decay}]
89     # if opt.use_adam:
90     #     self.optimizer = t.optim.Adam(params)
91     # else:
92     self.optimizer = t.optim.SGD(params, momentum=0.9)
93     return self.optimizer
94
95
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



8.4.3.4 get_optimizer() [2/2]

```
def model.faster_rcnn.FasterRCNN.get_optimizer (
    self )

return optimizer, It could be overwritten if you want to specify
special optimizer
```

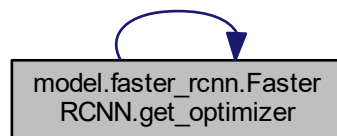
faster_rcnn.py 파일의 237 번째 라인에서 정의되었습니다.

```

237     def get_optimizer(self):
238         """
239         return optimizer, It could be overwritten if you want to specify
240         special optimizer
241         """
242         lr = opt.lr
243         params = []
244         for key, value in dict(self.named_parameters()).items():
245             if value.requires_grad:
246                 if 'bias' in key:
247                     params += [{'params': [value], 'lr': lr * 2, 'weight_decay': 0}]
248                 else:
249                     params += [{'params': [value], 'lr': lr, 'weight_decay': opt.weight_decay}]
250         if opt.use_adam:
251             self.optimizer = t.optim.Adam(params)
252         else:
253             self.optimizer = t.optim.SGD(params, momentum=0.9)
254         return self.optimizer
255

```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



8.4.3.5 n_class()

```

def model.faster_rcnn.FasterRCNN.n_class (
    self )

```

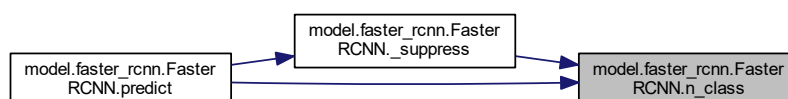
faster_rcnn.py 파일의 60 번째 라인에서 정의되었습니다.

```

60     def n_class(self):
61         # Total number of classes including the background.
62         return self.head.n_class
63

```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



8.4.3.6 predict()

```
def model.faster_rcnn.FasterRCNN.predict (
    self,
    imgs,
    sizes = None,
    visualize = False )
```

Detect objects from images.

This method predicts objects for each image.

Args:

imgs (iterable of numpy.ndarray): Arrays holding images.
All images are in CHW and RGB format
and the range of their value is :math:[0, 255].

Returns:

tuple of lists:
This method returns a tuple of three lists,
:obj:`(bboxes, labels, scores)`.

* **bboxes**: A list of float arrays of shape :math:(R, 4), \
where :math:'R' is the number of bounding boxes in a image. \
Each bounding box is organized by \
:math:(y_{min}, x_{min}, y_{max}, x_{max}) \
in the second axis.

* **labels**: A list of integer arrays of shape :math:(R,). \
Each value indicates the class of the bounding box. \
Values are in range :math:[0, L - 1], where :math:'L' is the \
number of the foreground classes.

* **scores**: A list of float arrays of shape :math:(R,). \
Each value indicates how confident the prediction is.

faster_rcnn.py 파일의 146 번째 라인에서 정의되었습니다.

```
146 def predict(self, imgs, sizes=None, visualize=False):
147     """Detect objects from images.
148
149     This method predicts objects for each image.
150
151     Args:
152         imgs (iterable of numpy.ndarray): Arrays holding images.
153         All images are in CHW and RGB format
154         and the range of their value is :math:[0, 255].
155
156     Returns:
157         tuple of lists:
158         This method returns a tuple of three lists,
159         :obj:`(bboxes, labels, scores)`.


160



161         * bboxes: A list of float arrays of shape :math:(R, 4), \
162           where :math:'R' is the number of bounding boxes in a image. \
163           Each bounding box is organized by \
164           :math:(y_{min}, x_{min}, y_{max}, x_{max}) \
165           in the second axis.



166         * labels: A list of integer arrays of shape :math:(R,). \
167           Each value indicates the class of the bounding box. \
168           Values are in range :math:[0, L - 1], where :math:'L' is the \
169           number of the foreground classes.



170         * scores: A list of float arrays of shape :math:(R,). \
171           Each value indicates how confident the prediction is.



172



173     """



174     self.eval()



175     self.use_preset('evaluate')



176     if visualize:



177         self.use_preset('visualize')



178         prepared_imgs = list()



179         sizes = list()



180         for img in imgs:



181             size = img.shape[1:]



182             img = preprocess(at.tonumpy(img))



183             prepared_imgs.append(img)



184             sizes.append(size)



185     else:

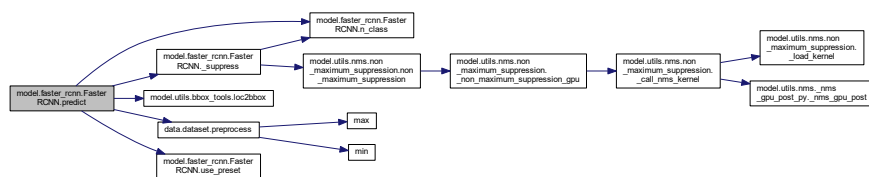

```

```

186         prepared_imgs = imgs
187         bboxes = list()
188         labels = list()
189         scores = list()
190         for img, size in zip(prepared_imgs, sizes):
191             with t.no_grad():
192                 img = t.autograd.Variable(at.totensor(img).float() [None])
193                 print("good")
194                 # img = t.autograd.Variable(at.totensor(img).float() [None], volatile=True)
195                 scale = img.shape[3] / size[1]
196                 # scale tensor 만들기 위해 추가
197                 scale_torch = t.ones([1], dtype=t.float32)
198                 scale_torch[0] = scale
199                 roi_cls_loc, roi_scores, rois, _ = self(img, scale=scale_torch)
200                 # We are assuming that batch size is 1.
201                 roi_score = roi_scores.data
202                 roi_cls_loc = roi_cls_loc.data
203                 roi = at.totensor(rois) / scale
204
205                 # Convert predictions to bounding boxes in image coordinates.
206                 # Bounding boxes are scaled to the scale of the input images.
207                 mean = t.Tensor(self.loc_normalize_mean).cuda(). \
208                     repeat(self.n_class) [None]
209                 std = t.Tensor(self.loc_normalize_std).cuda(). \
210                     repeat(self.n_class) [None]
211
212                 roi_cls_loc = (roi_cls_loc * std + mean)
213                 roi_cls_loc = roi_cls_loc.view(-1, self.n_class, 4)
214                 roi = roi.view(-1, 1, 4).expand_as(roi_cls_loc)
215                 cls_bbox = loc2bbox(at.tonumpy(roi).reshape((-1, 4)),
216                                     at.tonumpy(roi_cls_loc).reshape((-1, 4)))
217                 cls_bbox = at.totensor(cls_bbox)
218                 cls_bbox = cls_bbox.view(-1, self.n_class * 4)
219                 # clip bounding box
220                 cls_bbox[:, 0::2] = (cls_bbox[:, 0::2]).clamp(min=0, max=size[0])
221                 cls_bbox[:, 1::2] = (cls_bbox[:, 1::2]).clamp(min=0, max=size[1])
222
223                 prob = at.tonumpy(F.softmax(at.tovariable(roi_score), dim=1))
224
225                 raw_cls_bbox = at.tonumpy(cls_bbox)
226                 raw_prob = at.tonumpy(prob)
227
228                 bbox, label, score = self._suppress(raw_cls_bbox, raw_prob)
229                 bboxes.append(bbox)
230                 labels.append(label)
231                 scores.append(score)
232
233         self.use_preset('evaluate')
234         self.train()
235         return bboxes, labels, scores
236

```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



8.4.3.7 scale_lr()

```

def model.faster_rcnn.FasterRCNN.scale_lr (
    self,
    decay = 0.1 )

```

faster_rcnn.py 파일의 256 번째 라인에서 정의되었습니다.


```

256     def scale_lr(self, decay=0.1):
257         for param_group in self.optimizer.param_groups:
258             param_group['lr'] *= decay
259         return self.optimizer

```

8.4.3.8 use_preset()

```

def model.faster_rcnn.FasterRCNN.use_preset (
    self,
    preset )

```

Use the given preset during prediction.

This method changes values of :obj:`self.nms_thresh` and :obj:`self.score_thresh`. These values are a threshold value used for non maximum suppression and a threshold value to discard low confidence proposals in :meth:`predict`, respectively.

If the attributes need to be changed to something other than the values provided in the presets, please modify them by directly accessing the public attributes.

Args:

 preset ({'visualize', 'evaluate'}): A string to determine the preset to use.

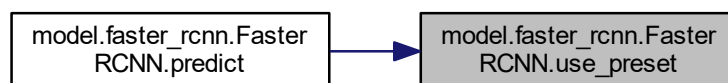
faster_rcnn.py 파일의 96 번째 라인에서 정의되었습니다.

```

96     def use_preset(self, preset):
97         """Use the given preset during prediction.
98
99         This method changes values of :obj:`self.nms_thresh` and
100         :obj:`self.score_thresh`. These values are a threshold value
101         used for non maximum suppression and a threshold value
102         to discard low confidence proposals in :meth:`predict`,
103         respectively.
104
105         If the attributes need to be changed to something
106         other than the values provided in the presets, please modify
107         them by directly accessing the public attributes.
108
109         Args:
110             preset ({'visualize', 'evaluate'}): A string to determine the
111             preset to use.
112
113         """
114         if preset == 'visualize':
115             self.nms_thresh = 0.3
116             self.score_thresh = 0.7
117         elif preset == 'evaluate':
118             self.nms_thresh = 0.3
119             self.score_thresh = 0.05
120         else:
121             raise ValueError('preset must be visualize or evaluate')
122

```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



8.4.4 멤버 데이터 문서화

8.4.4.1 extractor

`model.faster_rcnn.FasterRCNN.extractor`

`faster_rcnn.py` 파일의 48 번째 라인에서 정의되었습니다.

8.4.4.2 head

`model.faster_rcnn.FasterRCNN.head`

`faster_rcnn.py` 파일의 50 번째 라인에서 정의되었습니다.

8.4.4.3 loc_normalize_mean

`model.faster_rcnn.FasterRCNN.loc_normalize_mean`

`faster_rcnn.py` 파일의 53 번째 라인에서 정의되었습니다.

8.4.4.4 loc_normalize_std

`model.faster_rcnn.FasterRCNN.loc_normalize_std`

`faster_rcnn.py` 파일의 54 번째 라인에서 정의되었습니다.

8.4.4.5 nms_thresh

`model.faster_rcnn.FasterRCNN.nms_thresh`

`faster_rcnn.py` 파일의 115 번째 라인에서 정의되었습니다.

8.4.4.6 optimizer

`model.faster_rcnn.FasterRCNN.optimizer`

`faster_rcnn.py` 파일의 92 번째 라인에서 정의되었습니다.

8.4.4.7 rpn

`model.faster_rcnn.FasterRCNN.rpn`

`faster_rcnn.py` 파일의 49 번째 라인에서 정의되었습니다.

8.4.4.8 score_thresh

`model.faster_rcnn.FasterRCNN.score_thresh`

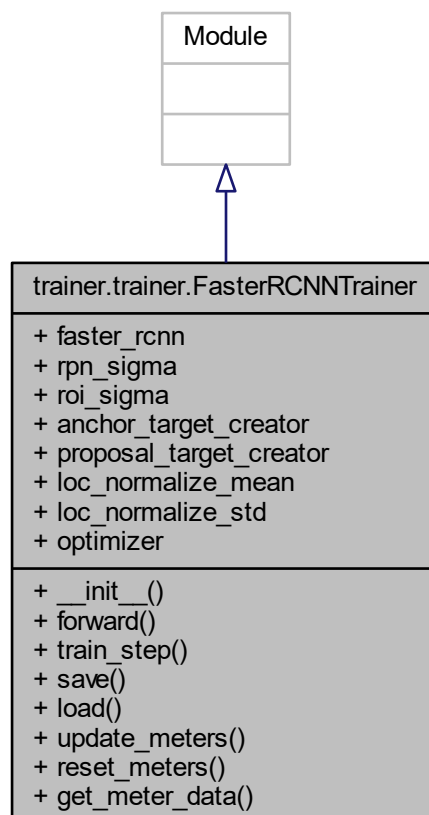
`faster_rcnn.py` 파일의 116 번째 라인에서 정의되었습니다.

이 클래스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

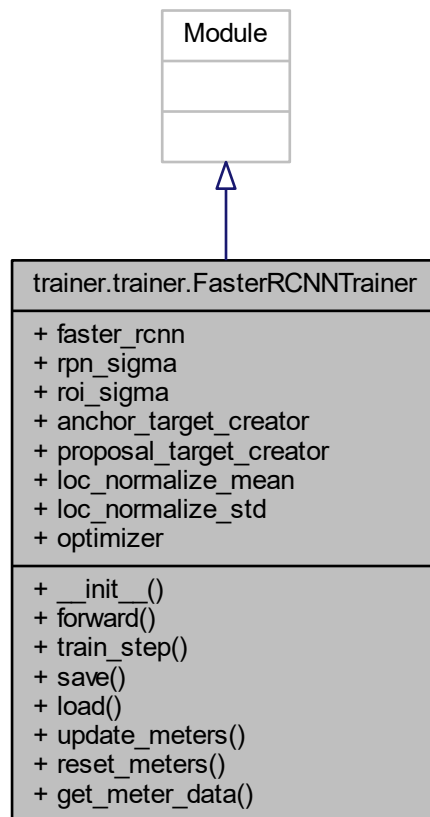
- `model/faster_rcnn.py`

8.5 trainer.trainer.FasterRCNNTrainer 클래스 참조

trainer.trainer.FasterRCNNTrainer에 대한 상속 다이어그램 :



trainer.trainer.FasterRCNNTrainer에 대한 협력 다이어그램:



Public 멤버 함수

- def `__init__` (self, `faster_rcnn`)
- def `forward` (self, imgs, bboxes, labels, scale)
Train forward code
- def `train_step` (self, imgs, bboxes, labels, scale)
Train step code
- def `save` (self, save_optimizer=False, save_path=None, **kwargs)
- def `load` (self, path, load_optimizer=True, parse_opt=False)
- def `update_meters` (self, losses)
- def `reset_meters` (self)
- def `get_meter_data` (self)

Public 속성

- `faster_rcnn`
- `rpn_sigma`
- `roi_sigma`

- [anchor_target_creator](#)
- [proposal_target_creator](#)
- [loc_normalize_mean](#)
- [loc_normalize_std](#)
- [optimizer](#)

8.5.1 상세한 설명

wrapper for conveniently training. return losses

The losses include:

```
* :obj:`rpn_loc_loss`: The localization loss for \
    Region Proposal Network (RPN).
* :obj:`rpn_cls_loss`: The classification loss for RPN.
* :obj:`roi_loc_loss`: The localization loss for the head module.
* :obj:`roi_cls_loss`: The classification loss for the head module.
* :obj:`total_loss`: The sum of 4 loss above.
```

Args:

```
faster_rcnn (model.FasterRCNN):
    A Faster R-CNN model that is going to be trained.
```

trainer.py 파일의 24 번째 라인에서 정의되었습니다.

8.5.2 생성자 & 소멸자 문서화

8.5.2.1 `__init__()`

```
def trainer.trainer.FasterRCNNTrainer.__init__(
    self,
    faster_rcnn )
```

trainer.py 파일의 41 번째 라인에서 정의되었습니다.

```
41     def __init__(self, faster_rcnn):
42         super(FasterRCNNTrainer, self).__init__()
43
44         self.faster_rcnn = faster_rcnn
45         self.rpn_sigma = opt.rpn_sigma
46         self.roi_sigma = opt.roi_sigma
47
48         # target creator create gt_bbox gt_label etc as training targets.
49         self.anchor_target_creator = AnchorTargetCreator()
50         self.proposal_target_creator = ProposalTargetCreator()
51
52         self.loc_normalize_mean = faster_rcnn.loc_normalize_mean
53         self.loc_normalize_std = faster_rcnn.loc_normalize_std
54
55         self.optimizer = self.faster_rcnn.get_optimizer()
56
```

8.5.3 멤버 함수 문서화

8.5.3.1 forward()

```
def trainer.trainer.FasterRCNNTrainer.forward (
    self,
    imgs,
    bboxes,
    labels,
    scale )
```

[Train](#) forward code

매개변수

imgs	input image
bboxes	B.B 정보, shape = (y1,x1,y2,x2)
labels	label 정보
scale	이미지 resize 비율

반환값

losses

Forward Faster R-CNN and calculate losses.

Here are notations used.

* :math:`N` is the batch size.
 * :math:`R` is the number of bounding boxes per image.

Currently, only :math:`N=1` is supported.

Args:

imgs (~torch.autograd.Variable): A variable with a batch of images.
 --> shape: 1 * 3 * height * width

bboxes (~torch.autograd.Variable): A batch of bounding boxes.
 Its shape is :math:`(N, R, 4)`.

labels (~torch.autograd.Variable): A batch of labels.
 Its shape is :math:`(N, R)` . The background is excluded from the definition, which means that the range of the value is :math:`[0, L - 1]` . :math:`L` is the number of foreground classes.

scale (float): Amount of scaling applied to the raw image during preprocessing.

Returns:

namedtuple of 5 losses

trainer.py 파일의 65 번째 라인에서 정의되었습니다.

```
65 def forward(self, imgs, bboxes, labels, scale):
66     """Forward Faster R-CNN and calculate losses.
67
68     Here are notations used.
69
70     * :math:`N` is the batch size.
71     * :math:`R` is the number of bounding boxes per image.
72
73     Currently, only :math:`N=1` is supported.
74
75     Args:
76         imgs ( torch.autograd.Variable): A variable with a batch of images.
77         --> shape: 1 * 3 * height * width
78
79         bboxes ( torch.autograd.Variable): A batch of bounding boxes.
80         Its shape is :math:`(N, R, 4)`.
```

```

81         labels ( torch.autograd.Variable): A batch of labels.
82         Its shape is :math:(N, R)'. The background is excluded from
83         the definition, which means that the range of the value
84         is :math:[0, L - 1]'. :math:'L' is the number of foreground
85         classes.
86         scale (float): Amount of scaling applied to
87         the raw image during preprocessing.
88
89     Returns:
90         namedtuple of 5 losses
91     """
92     n = bboxes.shape[0]
93     if n != 1:
94         raise ValueError('Currently only batch size 1 is supported.')
95
96     _, _, H, W = imgs.shape
97     img_size = (H, W)
98
99     # F.M shape = (1 * 512 * height(input height/16) * width(input width/16))
100    features = self.faster_rcnn.extractor(imgs)
101
102    # ----- RPN forward -----#
103    # rpn.py
104    rpn_locs, rpn_scores, rois, roi_indices, anchor = \
105        self.faster_rcnn.rpn(features, img_size, scale)
106
107    # Since batch size is one, convert variables to singular form ex) 1 * 1984 * 4 이면 1984 * 4로 변경
108    bbox = bboxes[0]
109    label = labels[0]
110    rpn_score = rpn_scores[0]
111    rpn_loc = rpn_locs[0]
112    roi = rois
113
114    # ----- RPN losses -----#
115    # rpn_gt_loc_label.py
116    gt_rpn_loc, gt_rpn_label = self.anchor_target_creator(
117        at.tonumpy(bbox),
118        anchor,
119        img_size)
120
121    # 전체 anchor 좌표와 B.B와의 상대좌표를 기록한 배열 --> shape = (F.M H * W * A, 4) 이지만 안에는 값
122    # 은 inside_index 안에만 기록되어 있음
123    gt_rpn_label = at.tovariable(gt_rpn_label).long()
124    # anchor에 대한 label을 기록한 배열 --> shape = (F.M H * W * A, 4(ctr_y, ctr_X, h, w))
125    gt_rpn_loc = at.tovariable(gt_rpn_loc)
126
127    # RPN regression loss
128    rpn_loc_loss = _fast_rcnn_loc_loss(
129        rpn_loc, # rpn_locs --> F.M의 하나의 픽셀당 9개의 anchor들 = pred location 즉 layer를 통해서 나온 값이다.
130        gt_rpn_loc,
131        gt_rpn_label.data,
132        self.rpn_sigma)
133
134    # RPN classification loss
135    # TODO focal loss 적용 해보기
136    rpn_cls_loss = F.cross_entropy(rpn_score, gt_rpn_label.cuda(), ignore_index=-1)
137
138    # ----- Sample RoIs and forward -----#
139    # roi_sample.py
140    sample_roi, gt_roi_loc, gt_roi_label = self.proposal_target_creator(
141        roi,
142        at.tonumpy(bbox),
143        at.tonumpy(label),
144        self.loc_normalize_mean,
145        self.loc_normalize_std)
146
147    # @note it's all zero because now it only support for batch=1 now
148    sample_roi_index = t.zeros(len(sample_roi))
149
150    # roi_module.py
151    # TODO network 수정 시 pooling layer input 변경
152    roi_cls_loc, roi_score = self.faster_rcnn.head(
153        features,
154        sample_roi,
155        sample_roi_index)
156
157    # ----- ROI losses (fast rcnn loss) -----#
158    n_sample = roi_cls_loc.shape[0]
159    roi_cls_loc = roi_cls_loc.view(n_sample, -1, 4)
160    roi_loc = roi_cls_loc[t.arange(0, n_sample).long().cuda(), \
161        at.totensor(gt_roi_label).long()]
162    gt_roi_label = at.tovariable(gt_roi_label).long()
163    gt_roi_loc = at.tovariable(gt_roi_loc)
164
165    # ROI regression loss
166    roi_loc_loss = _fast_rcnn_loc_loss(
167        roi_loc.contiguous(), # layer 거쳐서 나온 B.B

```

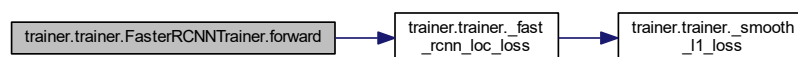


```

167         gt_roi_loc.float(), # 실제 B.B 와의 상대좌표 결과값.
168         gt_roi_label.data, # label값. pos와 neg 섞여 있음
169         self.roi_sigma)
170
171         # ROI classification loss
172         roi_cls_loss = nn.CrossEntropyLoss()(roi_score, gt_roi_label.cuda())
173
174         # -----#
175         losses = [rpn_loc_loss, rpn_cls_loss, roi_loc_loss, roi_cls_loss]
176         losses = losses + [sum(losses)] # loss[4]째에 전체다 더한 것을 보냄.
177         print(losses)
178         return LossTuple(*losses)
179

```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



이 함수를 호출하는 함수들에 대한 그래프입니다.:



8.5.3.2 get_meter_data()

```

def trainer.trainer.FasterRCNNTrainer.get_meter_data (
    self )

```

trainer.py 파일의 256 번째 라인에서 정의되었습니다.

```

256     def get_meter_data(self):
257         return {k: v.value()[0] for k, v in self.meters.items()}
258

```

8.5.3.3 load()

```

def trainer.trainer.FasterRCNNTrainer.load (
    self,
    path,
    load_optimizer = True,
    parse_opt = False )

```

trainer.py 파일의 232 번째 라인에서 정의되었습니다.

```

232     def load(self, path, load_optimizer=True, parse_opt=False, ):
233         state_dict = t.load(path)

```

```

234         if 'model' in state_dict:
235             self.faster_rcnn.load_state_dict(state_dict['model'])
236         else: # legacy way, for backward compatibility
237             self.faster_rcnn.load_state_dict(state_dict)
238         return self
239     if parse_opt:
240         opt._parse(state_dict['config'])
241     if 'optimizer' in state_dict and load_optimizer:
242         self.optimizer.load_state_dict(state_dict['optimizer'])
243     return self
244 
```

8.5.3.4 reset_meters()

```
def trainer.trainer.FasterRCNNTrainer.reset_meters (
    self )
```

trainer.py 파일의 250 번째 라인에서 정의되었습니다.

```

250     def reset_meters(self):
251         for key, meter in self.meters.items():
252             meter.reset()
253         self.roi_cm.reset()
254         self.rpn_cm.reset()
255 
```

8.5.3.5 save()

```
def trainer.trainer.FasterRCNNTrainer.save (
    self,
    save_optimizer = False,
    save_path = None,
    **kwargs )
```

serialize models include optimizer and other info

return path where the model-file is stored.

Args:

save_optimizer (bool): whether save optimizer.state_dict().

save_path (string): where to save model, if it's None, save_path is generate using time str and info from kwargs.

Returns:

save_path(str): the path to save models.

trainer.py 파일의 197 번째 라인에서 정의되었습니다.

```

197     def save(self, save_optimizer=False, save_path=None, **kwargs):
198         """serialize models include optimizer and other info
199         return path where the model-file is stored.
200         Args:
201             save_optimizer (bool): whether save optimizer.state_dict().
202             save_path (string): where to save model, if it's None, save_path
203             is generate using time str and info from kwargs.
204
205         Returns:
206             save_path(str): the path to save models.
207         """
208         save_dict = dict()
209
210         save_dict['model'] = self.faster_rcnn.state_dict()
211         save_dict['config'] = opt._state_dict()
212         save_dict['other_info'] = kwargs
213         # save_dict['vis_info'] = self.vis.state_dict()
214
215         if save_optimizer:

```

```

216         save_dict['optimizer'] = self.optimizer.state_dict()
217
218         if save_path is None:
219             timestr = time.strftime('%m%d%H%M')
220             save_path = './checkpoints/fasterrcnn_%s' % timestr
221             for k_, v_ in kwargs.items():
222                 save_path += '_%.4s' % v_
223             save_path += '.pth'
224             save_dir = os.path.dirname(save_path)
225             if not os.path.exists(save_dir):
226                 os.makedirs(save_dir)
227
228             t.save(save_dict, save_path)
229             # self.vis.save([self.vis.env])
230             return save_path
231

```

8.5.3.6 train_step()

```

def trainer.trainer.FasterRCNNTrainer.train_step (
    self,
    imgs,
    bboxes,
    labels,
    scale )

```

Train step code

매개변수

imgs	input image
bboxes	B.B 정보
labels	label 정보
scale	이미지 resize 비율

반환값

losses

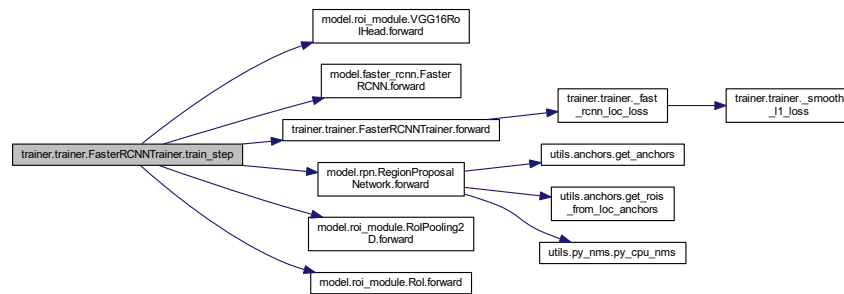
trainer.py 파일의 188 번째 라인에서 정의되었습니다.

```

188     def train_step(self, imgs, bboxes, labels, scale):
189         self.optimizer.zero_grad()
190         losses = self.forward(imgs, bboxes, labels, scale)
191         # TODO: 단계별로 학습
192         # 현재는 RPN & ROI 합친 loss backward
193         losses.total_loss.backward()
194         self.optimizer.step()
195         return losses
196

```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



8.5.3.7 update_meters()

```
def trainer.trainer.FasterRCNNTrainer.update_meters (
    self,
    losses )
```

trainer.py 파일의 245 번째 라인에서 정의되었습니다.

```
245 def update_meters(self, losses):
246     loss_d = {k: at.scalar(v) for k, v in losses.__asdict().items()}
247     for key, meter in self.meters.items():
248         meter.add(loss_d[key])
249
```

8.5.4 멤버 데이터 문서화

8.5.4.1 anchor_target_creator

```
trainer.trainer.FasterRCNNTrainer.anchor_target_creator
```

trainer.py 파일의 49 번째 라인에서 정의되었습니다.

8.5.4.2 faster_rcnn

```
trainer.trainer.FasterRCNNTrainer.faster_rcnn
```

trainer.py 파일의 44 번째 라인에서 정의되었습니다.

8.5.4.3 loc_normalize_mean

trainer.trainer.FasterRCNNTrainer.loc_normalize_mean

trainer.py 파일의 52 번째 라인에서 정의되었습니다.

8.5.4.4 loc_normalize_std

trainer.trainer.FasterRCNNTrainer.loc_normalize_std

trainer.py 파일의 53 번째 라인에서 정의되었습니다.

8.5.4.5 optimizer

trainer.trainer.FasterRCNNTrainer.optimizer

trainer.py 파일의 55 번째 라인에서 정의되었습니다.

8.5.4.6 proposal_target_creator

trainer.trainer.FasterRCNNTrainer.proposal_target_creator

trainer.py 파일의 50 번째 라인에서 정의되었습니다.

8.5.4.7 roi_sigma

trainer.trainer.FasterRCNNTrainer.roi_sigma

trainer.py 파일의 46 번째 라인에서 정의되었습니다.

8.5.4.8 rpn_sigma

trainer.trainer.FasterRCNNTrainer.rpn_sigma

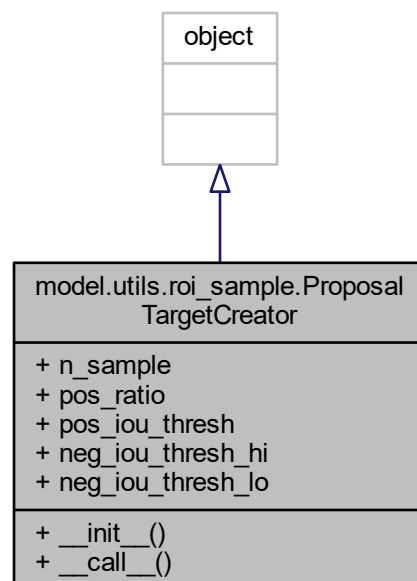
trainer.py 파일의 45 번째 라인에서 정의되었습니다.

이 클래스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

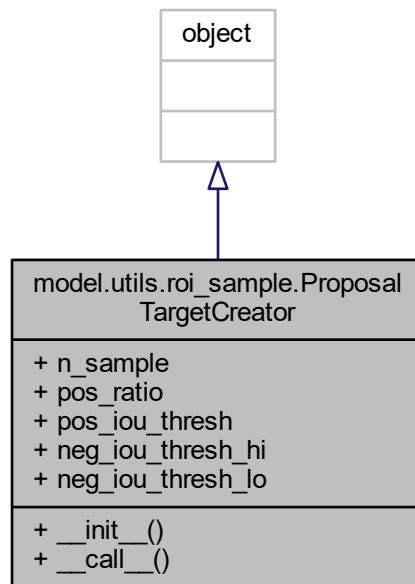
- [trainer/trainer.py](#)

8.6 model.utils.roi_sample.ProposalTargetCreator 클래스 참조

model.utils.roi_sample.ProposalTargetCreator에 대한 상속 다이어그램 :



model.utils.roi_sample.ProposalTargetCreator에 대한 협력 다이어그램:



Public 멤버 함수

- `def __init__(self, n_sample=128, pos_ratio=0.25, pos_iou_thresh=0.5, neg_iou_thresh_hi=0.5, neg_iou_thresh_lo=0.0)`
- `def __call__(self, roi, bbox, label, loc_normalize_mean=(0., 0., 0., 0.), loc_normalize_std=(0.1, 0.1, 0.2, 0.2))`

RPN 네트워크에서 뽑은 sample anchor 중에서 IoU를 기준으로 pos와 negative의 비율을 1:3으로 맞춰

Public 속성

- `n_sample`
- `pos_ratio`
- `pos_iou_thresh`
- `neg_iou_thresh_hi`
- `neg_iou_thresh_lo`

8.6.1 상세한 설명

Assign ground truth bounding boxes to given RoIs.

The :meth:`__call__` of this class generates training targets for each object proposal.
This is used to train Faster RCNN [#].

.. [#] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun. \

Faster R-CNN: Towards Real-Time Object Detection with \ Region Proposal Networks. NIPS 2015.

Args:

```
n_sample (int): The number of sampled regions.
pos_ratio (float): Fraction of regions that is labeled as a
    foreground. # sample에서 pos한 비율
pos_iou_thresh (float): IoU threshold for a RoI to be considered as a
    foreground. #pos로 간주되기 위한 GT와 겹치는 iou
neg_iou_thresh_hi (float): RoI is considered to be the background
    if IoU is in # neg로 간주되기 위한 back ground와 겹치는 비율
    [:obj:'neg_iou_thresh_hi', :obj:'neg_iou_thresh_hi').
neg_iou_thresh_lo (float): See above.
```

roi_sample.py 파일의 4 번째 라인에서 정의되었습니다.

8.6.2 생성자 & 소멸자 문서화

8.6.2.1 __init__()

```
def model.utils.roi_sample.ProposalTargetCreator.__init__(
    self,
    n_sample = 128,
    pos_ratio = 0.25,
    pos_iou_thresh = 0.5,
    neg_iou_thresh_hi = 0.5,
    neg_iou_thresh_lo = 0.0 )
```

roi_sample.py 파일의 28 번째 라인에서 정의되었습니다.

```
28 def __init__(self,
29             n_sample=128,
30             pos_ratio=0.25, pos_iou_thresh=0.5,
31             neg_iou_thresh_hi=0.5, neg_iou_thresh_lo=0.0
32             ):
33     self.n_sample = n_sample
34     self.pos_ratio = pos_ratio
35     self.pos_iou_thresh = pos_iou_thresh
36     self.neg_iou_thresh_hi = neg_iou_thresh_hi
37     self.neg_iou_thresh_lo = neg_iou_thresh_lo
38
```

8.6.3 멤버 함수 문서화

8.6.3.1 __call__()

```
def model.utils.roi_sample.ProposalTargetCreator.__call__(
    self,
    roi,
    bbox,
    label,
    loc_normalize_mean = (0., 0., 0., 0.),
    loc_normalize_std = (0.1, 0.1, 0.2, 0.2) )
```

RPN 네트워크에서 뽑은 sample anchor 중에서 IoU를 기준으로 pos와 negative의 비율을 1:3으로 맞춰 실제 object 가 있을 법한 roi와 background 만 있는 roi를 뽑는다.

매개변수

roi	예측된 sample anchor (최대 2000개)
bbox	B.B 정보
label	B.B label 정보
loc_normalize_mean	(0., 0., 0., 0.)
loc_normalize_std	(0.1, 0.1, 0.2, 0.2)

반환값

sample_roi : sample anchor 중에서 pos 와 neg 비율 대로 뽑은 것

gt_roi_loc : 위에서 뽑은 anchor 와 B.B G.T와의 상대좌표

gt_roi_label : 위에서 뽑은 anchor 의 label

Assigns ground truth to sampled proposals.

This function samples total of :obj:`self.n_sample` RoIs from the combination of :obj:`roi` and :obj:`bbox`. The RoIs are assigned with the ground truth class labels as well as bounding box offsets and scales to match the ground truth bounding boxes. As many as :obj:`pos_ratio * self.n_sample` RoIs are sampled as foregrounds.

Offsets and scales of bounding boxes are calculated using :func:`model.utils.bbox_tools.bbox2loc`. Also, types of input arrays and output arrays are same.

Here are notations.

* :math:`S` is the total number of sampled RoIs, which equals \ :obj:`self.n_sample`.
* :math:`L` is number of object classes possibly including the \ background.

Args:

roi (array): Region of Interests (RoIs) from which we sample. Its shape is :math:`(R, 4)`
bbox (array): The coordinates of ground truth bounding boxes. Its shape is :math:`(R', 4)`
label (array): Ground truth bounding box labels. Its shape is :math:`(R',)` . Its range is :math:`[0, L - 1]` , where :math:`L` is the number of foreground classes.
loc_normalize_mean (tuple of four floats): Mean values to normalize coordinates of bounding boxes.
loc_normalize_std (tupler of four floats): Standard deviation of the coordinates of bounding boxes.

Returns:

(array, array, array):
* **sample_roi**: Regions of interests that are sampled. \ Its shape is :math:`(S, 4)`
* **gt_roi_loc**: Offsets and scales to match \ the sampled RoIs to the ground truth bounding boxes. \ Its shape is :math:`(S, 4)`
* **gt_roi_label**: Labels assigned to sampled RoIs. Its shape is \ :math:`(S,)` . Its range is :math:`[0, L]` . The label with \ value 0 is the background.

roi_sample.py 파일의 51 번째 라인에서 정의되었습니다.

```
51 def __call__(self, roi, bbox, label,
52             loc_normalize_mean=(0., 0., 0., 0.),
53             loc_normalize_std=(0.1, 0.1, 0.2, 0.2)):
54     """Assigns ground truth to sampled proposals.
55
```

```

56     This function samples total of :obj:`self.n_sample` RoIs
57     from the combination of :obj:`roi` and :obj:`bbox`.
58     The RoIs are assigned with the ground truth class labels as well as
59     bounding box offsets and scales to match the ground truth bounding
60     boxes. As many as :obj:`pos_ratio * self.n_sample` RoIs are
61     sampled as foregrounds.
62
63     Offsets and scales of bounding boxes are calculated using
64     :func:`model.utils.bbox_tools.bbox2loc`.
65     Also, types of input arrays and output arrays are same.
66
67     Here are notations.
68
69     * :math:`S` is the total number of sampled RoIs, which equals \
70       :obj:`self.n_sample`.
71     * :math:`L` is number of object classes possibly including the \
72       background.
73
74     Args:
75         roi (array): Region of Interests (RoIs) from which we sample.
76             Its shape is :math:`(R, 4)`
77         bbox (array): The coordinates of ground truth bounding boxes.
78             Its shape is :math:`(R', 4)`
79         label (array): Ground truth bounding box labels. Its shape
80             is :math:`(R', )`. Its range is :math:`[0, L - 1]`, where
81             :math:`L` is the number of foreground classes.
82         loc_normalize_mean (tuple of four floats): Mean values to normalize
83             coordinates of bounding boxes.
84         loc_normalize_std (tuple of four floats): Standard deviation of
85             the coordinates of bounding boxes.
86
87     Returns:
88         (array, array, array):
89
90         * **sample_roi**: Regions of interests that are sampled. \
91           Its shape is :math:`(S, 4)`
92         * **gt_roi_loc**: Offsets and scales to match \
93           the sampled RoIs to the ground truth bounding boxes. \
94           Its shape is :math:`(S, 4)`
95         * **gt_roi_label**: Labels assigned to sampled RoIs. Its shape is \
96           :math:`(S, )`. Its range is :math:`[0, L]`. The label with \
97           value 0 is the background.
98
99     """
100     n_bbox, _ = bbox.shape
101     # sample로 뽑은 roi뒤에 B.B 정보를 붙인다.
102     roi = np.concatenate((roi, bbox), axis=0)
103
104
105     pos_roi_per_image = np.round(self.n_sample * self.pos_ratio) # 뽑을 pos 개수 128 * 0.25 = 32
106
107     iou = bbox_iou(roi, bbox) # sample anchor에 대한 iou
108     gt_assignment = iou.argmax(axis=1) # B.B 카테고리 중에서 iou가 큰 index값 설정
109     max_iou = iou.max(axis=1) # B.B 카테고리 중에서 iou가 큰 값 설정
110
111     # Offset range of classes from [0, n_fg_class - 1] to [1, n_fg_class].
112     # +1 의미: 해주는 이유는 background 가 0으로 붙기 때문이다.
113     gt_roi_label = label[gt_assignment] + 1
114
115     # Select foreground RoIs as those with : roi 중에서 object 인 것만 select
116     pos_index = np.where(max_iou >= self.pos_iou_thresh)[0] # 0.5 보다 큰 것만 select
117     pos_roi_per_this_image = int(min(pos_roi_per_image, pos_index.size))
118     if pos_index.size > 0:
119         pos_index = np.random.choice(
120             pos_index, size=pos_roi_per_this_image, replace=False)
121
122     # Select background RoIs as those within : roi 중에서 background 인 것만 select
123     neg_index = np.where((max_iou < self.neg_iou_thresh_hi) &
124                          (max_iou >= self.neg_iou_thresh_lo))[0]
125     neg_roi_per_this_image = self.n_sample - pos_roi_per_this_image
126     neg_roi_per_this_image = int(min(neg_roi_per_this_image,
127                                     neg_index.size))
128     if neg_index.size > 0:
129         neg_index = np.random.choice(
130             neg_index, size=neg_roi_per_this_image, replace=False)
131
132     # The indices that we're selecting (both positive and negative).
133     keep_index = np.append(pos_index, neg_index) # pos roi index 뒤에 neg roi index를 붙인다.
134     gt_roi_label = gt_roi_label[keep_index]
135     # sample roi 의 negative 즉 background label 0으로 할당.
136     gt_roi_label[pos_roi_per_this_image:] = 0
137     sample_roi = roi[keep_index]
138
139     # Compute offsets and scales to match sampled RoIs to the GTs.
140     @note 뽑힌 sample roi(pos & negative 비율 1 : 3) 와 실제 B.B 와의 상대좌표
141     gt_roi_loc = bbox2loc(sample_roi, bbox[gt_assignment[keep_index]])
142     gt_roi_loc = ((gt_roi_loc - np.array(loc_normalize_mean, np.float32)

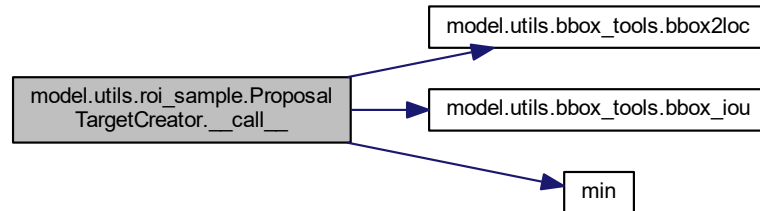
```

```

143         ) / np.array(loc_normalize_std, np.float32))
144
145     return sample_roi, gt_roi_loc, gt_roi_label

```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



8.6.4 멤버 데이터 문서화

8.6.4.1 n_sample

`model.utils.roi_sample.ProposalTargetCreator.n_sample`

`roi_sample.py` 파일의 29 번째 라인에서 정의되었습니다.

8.6.4.2 neg_iou_thresh_hi

`model.utils.roi_sample.ProposalTargetCreator.neg_iou_thresh_hi`

`roi_sample.py` 파일의 32 번째 라인에서 정의되었습니다.

8.6.4.3 neg_iou_thresh_lo

`model.utils.roi_sample.ProposalTargetCreator.neg_iou_thresh_lo`

`roi_sample.py` 파일의 33 번째 라인에서 정의되었습니다.

8.6.4.4 pos_iou_thresh

`model.utils.roi_sample.ProposalTargetCreator.pos_iou_thresh`

`roi_sample.py` 파일의 31 번째 라인에서 정의되었습니다.

8.6.4.5 pos_ratio

`model.utils.roi_sample.ProposalTargetCreator.pos_ratio`

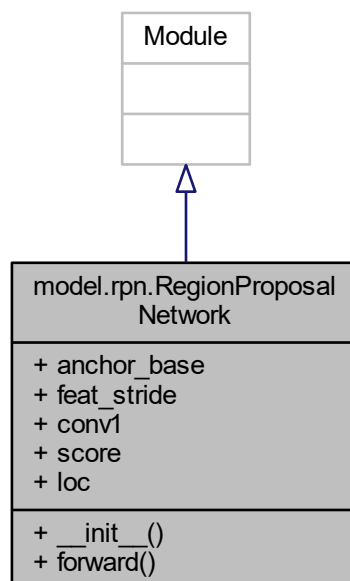
`roi_sample.py` 파일의 30 번째 라인에서 정의되었습니다.

이 클래스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

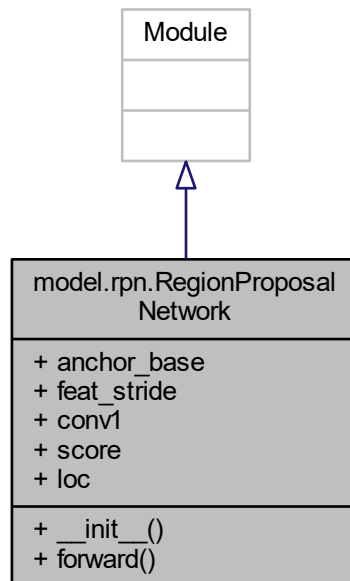
- [model/utils/roi_sample.py](#)

8.7 model.rpn.RegionProposalNetwork 클래스 참조

`model.rpn.RegionProposalNetwork`에 대한 상속 다이어그램 :



model.rpn.RegionProposalNetwork에 대한 협력 다이어그램:



Public 멤버 함수

- `def __init__(self, in_channels=512, mid_channels=512, ratios=[0.5, 1, 2], scales=[0.5, 1, 2], feat_stride=16)`
- `def forward(self, h, img_size, scale=1.)`
Forward Region Proposal Network.

Public 속성

- `anchor_base`
- `feat_stride`
- `conv1`
- `score`
- `loc`

8.7.1 상세한 설명

@brief Region Proposal Network introduced in Faster R-CNN.

@param in_channels (int): The channel size of input.

@param mid_channels (int): The channel size of the intermediate tensor.

@param ratios (list of floats): This is ratios of width to height of the anchors.

@param anchor_scales (list of numbers): This is areas of anchors.

```

    Those areas will be the product of the square of an element in
    :obj:`anchor_scales` and the original area of the reference window.

@param feat_stride (int): Stride size after extracting features from an image.

@param initialW (callable): Initial weight value. If :obj:`None` then this
    function uses Gaussian distribution scaled by 0.1 to initialize weight.
    May also be a callable that takes an array and edits its values.

@param proposal_creator_params (dict): Key valued paramters for
    :class:`model.utils.creator_tools.ProposalCreator`.

.. seealso::
    :class:`~model.utils.creator_tools.ProposalCreator`

```

rpn.py 파일의 8 번째 라인에서 정의되었습니다.

8.7.2 생성자 & 소멸자 문서화

8.7.2.1 __init__()

```

def model.rpn.RegionProposalNetwork.__init__(
    self,
    in_channels = 512,
    mid_channels = 512,
    ratios = [0.5, 1, 2],
    scales = [0.5, 1, 2],
    feat_stride = 16 )

```

rpn.py 파일의 36 번째 라인에서 정의되었습니다.

```

36 def __init__(
37     self, in_channels=512, mid_channels=512, ratios=[0.5, 1, 2],
38     scales=[0.5, 1, 2], feat_stride=16):
39     super(RegionProposalNetwork, self).__init__()
40     # prepare anchor base
41     # side_length 는 16 * 16 의 single box (block side length)
42     self.anchor_base = generate_anchor_base(side_length=16,
43     ratios=ratios, scales=scales, strides=feat_stride)
44     self.feat_stride = feat_stride
45     # network params
46     n_anchor = self.anchor_base.shape[0] # n_anchor = 9 : anchor의
47     self.conv1 = nn.Conv2d(in_channels, mid_channels, 3, 1, 1) # input, output, kernel, stride,
padding : 3 * 3 filter 해준
다.
48     self.score = nn.Conv2d(mid_channels, n_anchor * 2, 1, 1, 0) # B.B 에 대한 점수 object 인지 object 가 아닌
지
49     self.loc = nn.Conv2d(mid_channels, n_anchor * 4, 1, 1, 0) # regression 에 대한 위
치 (regression location)
50     normal_init(self.conv1, 0, 0.01)
51     normal_init(self.score, 0, 0.01)
52     normal_init(self.loc, 0, 0.01)
53
54
55

```

8.7.3 멤버 함수 문서화

8.7.3.1 forward()

```
def model.rpn.RegionProposalNetwork.forward (
    self,
    h,
    img_size,
    scale = 1. )
```

Forward Region Proposal Network.

매개변수

h	F.M 의미, shape = (N, C, H, W)
img_size	Resize 된 크기의 input image
scale	Resize의 비율

반환값

```
rpn_locs : F.M의 한 픽셀에 해당하는 9개의 anchor 좌표, shape = (N, H W A, 4)
rpn_scores : F.M의 한 픽셀에 해당하는 9개의 anchor 각각에 대한 object 존재 확률값 shape = (N, H W A, 2)
rois : 예측된 sample anchor
[0]*len(rois) : sample anchor 개수 만큼의 list
anchors : F.M 한 픽셀당 9개의 anchor 좌표 초기값, shape = (H W A, 4)

- :math:'N' is batch size.
- :math:'C' channel size of the input.
- :math:'H' and :math:'W' are height and width of the input feature.
- :math:'A' is number of anchors assigned to each pixel.

Args:
    x (~torch.autograd.Variable): The Features extracted from images.
    Its shape is :math:'(N, C, H, W)'.

    img_size (tuple of ints): A tuple :obj:'height, width', which contains image size after scaling.

    scale (float): The amount of scaling done to the input images after reading them from files.

Returns:
    (~torch.autograd.Variable, ~torch.autograd.Variable, array, array, array):

    This is a tuple of five following values.

    * **rpn_locs**: Predicted bounding box offsets and scales for \
anchors. Its shape is :math:'(N, H W A, 4)'.
    * **rpn_scores**: Predicted foreground scores for \
anchors. Its shape is :math:'(N, H W A, 2)'.
    * **rois**: A bounding box array containing coordinates of \
proposal boxes. This is a concatenation of bounding box \
arrays from multiple images in the batch. \
Its shape is :math:'(R', 4)'. Given :math:'R_i' predicted \
bounding boxes from the :math:'i' th image, \
:math:'R' = \sum_{i=1}^N R_i'.
    * **roi_indices**: An array containing indices of images to \
which RoIs correspond to. Its shape is :math:'(R',)'.
    * **anchor**: Coordinates of enumerated shifted anchors. \
Its shape is :math:'(H W A, 4)'.
```

rpn.py 파일의 67 번째 라인에서 정의되었습니다.

```
67 def forward(self, h, img_size, scale=1.):
68     """
```

```

69     - :math:`N` is batch size.
70     - :math:`C` channel size of the input.
71     - :math:`H` and :math:`W` are height and width of the input feature.
72     - :math:`A` is number of anchors assigned to each pixel.
73
74     Args:
75         x ( torch.autograd.Variable): The Features extracted from images.
76             Its shape is :math:`(N, C, H, W)` .
77
78         img_size (tuple of ints): A tuple :obj:`height, width`, which contains image size after
scaling.
79
80         scale (float): The amount of scaling done to the input images after reading them from files.
81
82     Returns:
83         ( torch.autograd.Variable, torch.autograd.Variable, array, array, array):
84
85         This is a tuple of five following values.
86
87         * **rpn_locs**: Predicted bounding box offsets and scales for \
anchors. Its shape is :math:`(N, H W A, 4)` .
88         * **rpn_scores**: Predicted foreground scores for \
anchors. Its shape is :math:`(N, H W A, 2)` .
89         * **rois**: A bounding box array containing coordinates of \
proposal boxes. This is a concatenation of bounding box \
arrays from multiple images in the batch. \
Its shape is :math:`(R', 4)` . Given :math:`R_i` predicted \
bounding boxes from the :math:`i` th image, \
:math:`R' = \sum_{i=1}^N R_i` .
90         * **roi_indices**: An array containing indices of images to
which RoIs correspond to. Its shape is :math:`(R',)` .
91         * **anchor**: Coordinates of enumerated shifted anchors. \
Its shape is :math:`(H W A, 4)` .
92
93     """
94     # NMS에 넣기 전 뽑는 objectness가 높은 순서의 anchor 개수
95     n_pre_nms = 12000
96     # NMS로 뽑을 anchor 수 (최대 2000개, 2000개보다 적을 수 있음)
97     n_post_nms = 2000
98     # NMS G.T 와 겹치는 anchor의 IoU threshold
99     nms_thresh = 0.7
100
101     # feature map의 shape
102     n, _, hh, ww = h.shape
103
104     # utils --> anchors.py
105     anchors = get_anchors(self.anchor_base, self.feat_stride, hh, ww)
106
107     hidd = F.relu(self.conv1(h))
108     # rpn_locs shape = 1 * 36(9 * 4) * feature map height * feature map width
109     rpn_locs = self.loc(hidd)
110     # rpn_scores shape = 1 * 18(9 * 2) * feature map height * feature map width
111     rpn_scores = self.score(hidd)
112
113     # shape : 1 * (feature height * feature width * 9) * 4
114     rpn_locs = rpn_locs.permute(0, 2, 3, 1).contiguous().view(n, -1, 4)
115     # shape : 1 * (feature height * feature width * 9) * 2
116     rpn_scores = rpn_scores.permute(0, 2, 3, 1).contiguous().view(n, -1, 2)
117     # 0 or 1 중 1(object에 대한 확률 값)
118     scores = rpn_scores[:, :, 1].data.cpu().numpy()[0]
119
120     # RPN을 통해서 나온 anchor값(상대좌표)을 input image 크기의 anchor 좌표로 변환
121     rois = get_rois_from_loc_anchors(anchors, rpn_locs[0].data.cpu().numpy())
122
123     # rois의 좌표 값이 이미지 크기 밖에 있을 경우 np.clip을 통해 보정
124     rois[:, :2] = np.clip(rois[:, :2], 0, img_size[0])
125     rois[:, 1:2] = np.clip(rois[:, 1:2], 0, img_size[1])
126
127     # scale: change type (tensor -> array)
128     scale = scale.numpy()
129
130     # 이미지를 resize 하기 때문에 최소 픽셀값 "16" 값에도 scale 곱함
131     min_size = 16.
132     min_size = min_size * scale
133
134     hs = rois[:, 2] - rois[:, 0] # height
135     ws = rois[:, 3] - rois[:, 1] # width
136
137     keep = np.where((hs >= min_size) & (ws >= min_size))[0]
138     rois = rois[keep, :] # 최소 픽셀 이상인 anchor의 select (anchor 박스의 크기가 16 * 16 보다 작으면 탈락)
139     scores = scores[keep]
140
141     # ravel(): 다차원 배열을 1차원으로 펼침
142     # argsort(): 오름차순 정렬된 원소의 index를 반환
143     # [::-1]: 역순으로 반환
144     # order: object 존재 score가 높은 순서대로 index 저장

```

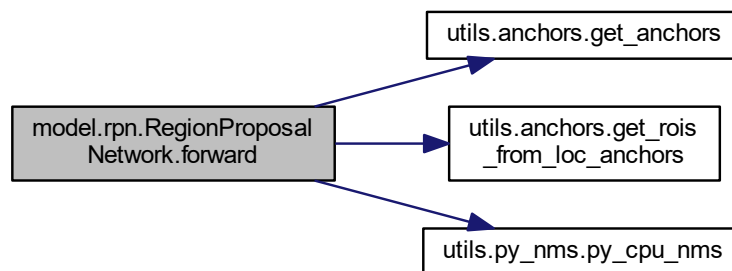


```

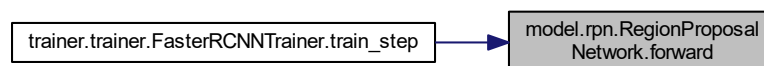
155         order = scores.ravel().argsort()[::-1]
156         if n_pre_nms > 0:
157             order = order[:n_pre_nms] # score 가 높은 순서대로 12000개 뽑음
158             rois = rois[order, :]
159
160         # NMS (py_nms.py)
161         keep = py_cpu_nms(rois, nms_thresh)
162         # IoU 0.7보다 큰 것중에서 상위 2000개를 선택. 꼭 2000개가 아니라 2000개보다 작을 수도 있다
163         keep = keep[:n_post_nms]
164         # 2000개 정도를 (2000보다 작을수도 있음) 뽑은 것의 roi만 저장
165         rois = rois[keep]
166         return rpn_locs, rpn_scores, rois, [0]*len(rois), anchors
167

```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



이 함수를 호출하는 함수들에 대한 그래프입니다.:



8.7.4 멤버 데이터 문서화

8.7.4.1 anchor_base

`model.rpn.RegionProposalNetwork.anchor_base`

`rpn.py` 파일의 40 번째 라인에서 정의되었습니다.

8.7.4.2 conv1

`model.rpn.RegionProposalNetwork.conv1`

rpn.py 파일의 45 번째 라인에서 정의되었습니다.

8.7.4.3 feat_stride

`model.rpn.RegionProposalNetwork.feat_stride`

rpn.py 파일의 42 번째 라인에서 정의되었습니다.

8.7.4.4 loc

`model.rpn.RegionProposalNetwork.loc`

rpn.py 파일의 47 번째 라인에서 정의되었습니다.

8.7.4.5 score

`model.rpn.RegionProposalNetwork.score`

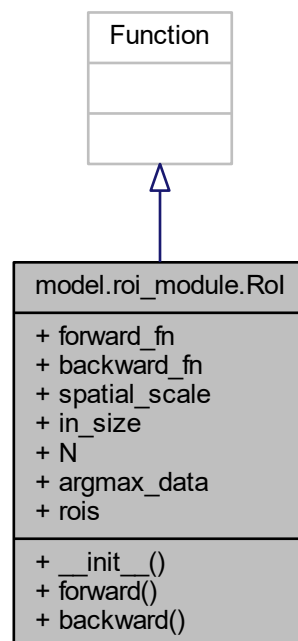
rpn.py 파일의 46 번째 라인에서 정의되었습니다.

이 클래스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

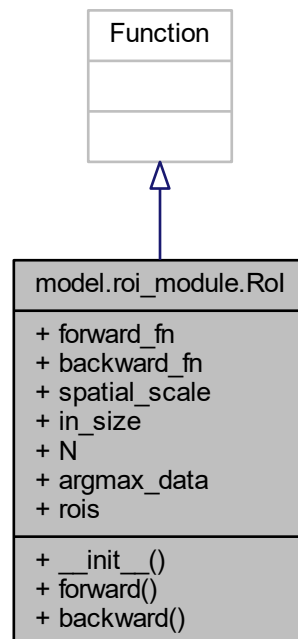
- [model/rpn.py](#)

8.8 model.roi_module.Rol 클래스 참조

model.roi_module.Rol에 대한 상속 다이어그램 :



model.roi_module.RoI에 대한 협력 다이어그램:



Public 멤버 함수

- `def __init__` (self, outh, outw, `spatial_scale`)
- `def forward` (self, x, `rois`)
- `def backward` (self, grad_output)

Public 속성

- `forward_fn`
- `backward_fn`
- `spatial_scale`
- `in_size`
- `N`
- `argmax_data`
- `rois`

8.8.1 상세한 설명

NOTE : only CUDA-compatible

`roi_module.py` 파일의 124 번째 라인에서 정의되었습니다.

8.8.2 생성자 & 소멸자 문서화

8.8.2.1 __init__()

```
def model.roi_module.RoI.__init__(
    self,
    outh,
    outw,
    spatial_scale )
```

roi_module.py 파일의 129 번째 라인에서 정의되었습니다.

```
129     def __init__(self, outh, outw, spatial_scale):
130         self.forward_fn = load_kernel('roi_forward', kernel_forward)
131         self.backward_fn = load_kernel('roi_backward', kernel_backward)
132         self.outh, self.outw, self.spatial_scale = outh, outw, spatial_scale
133
```

8.8.3 멤버 함수 문서화

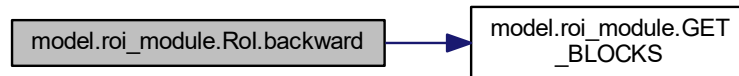
8.8.3.1 backward()

```
def model.roi_module.RoI.backward (
    self,
    grad_output )
```

roi_module.py 파일의 156 번째 라인에서 정의되었습니다.

```
156     def backward(self, grad_output):
157         # NOTE: IMPORTANT CONTIGUOUS
158         # TODO: input
159         grad_output = grad_output.contiguous()
160         B, C, H, W = self.in_size
161         grad_input = t.zeros(self.in_size).cuda()
162         stream = Stream(ptr=torch.cuda.current_stream().cuda_stream)
163         args = [grad_output.data_ptr(),
164                 self.argmax_data.data_ptr(),
165                 self.rois.data_ptr(),
166                 grad_input.data_ptr(),
167                 self.N, self.spatial_scale, C, H, W, self.outh, self.outw,
168                 grad_input.numel()]
169         self.backward_fn(args=args,
170                          block=(CUDA_NUM_THREADS, 1, 1),
171                          grid=(GET_BLOCKS(grad_input.numel()), 1, 1),
172                          stream=stream
173                          )
174         return grad_input, None
175
176
177
178
179
180
```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



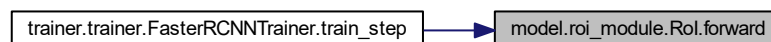
8.8.3.2 forward()

```
def model.roi_module.Rol.forward (
    self,
    x,
    rois )
```

roi_module.py 파일의 134 번째 라인에서 정의되었습니다.

```
134     def forward(self, x, rois):
135         # NOTE: MAKE SURE input is contiguous too
136         x = x.contiguous()
137         rois = rois.contiguous()
138         self.in_size = B, C, H, W = x.size()
139         self.N = N = rois.size(0)
140         output = t.zeros(N, C, self.outh, self.outw).cuda()
141         self.argmax_data = t.zeros(N, C, self.outh, self.outw).int().cuda()
142         self.rois = rois
143         args = [x.data_ptr(), rois.data_ptr(),
144                output.data_ptr(),
145                self.argmax_data.data_ptr(),
146                self.spatial_scale, C, H, W,
147                self.outh, self.outw,
148                output.numel()]
149         stream = Stream(ptr=torch.cuda.current_stream().cuda_stream)
150         self.forward_fn(args=args,
151                        block=(CUDA_NUM_THREADS, 1, 1),
152                        grid=(GET_BLOCKS(output.numel()), 1, 1),
153                        stream=stream)
154         return output
155
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



8.8.4 멤버 데이터 문서화

8.8.4.1 argmax_data

model.roi_module.Roi.argmax_data

roi_module.py 파일의 141 번째 라인에서 정의되었습니다.

8.8.4.2 backward_fn

model.roi_module.Roi.backward_fn

roi_module.py 파일의 131 번째 라인에서 정의되었습니다.

8.8.4.3 forward_fn

model.roi_module.Roi.forward_fn

roi_module.py 파일의 130 번째 라인에서 정의되었습니다.

8.8.4.4 in_size

model.roi_module.Roi.in_size

roi_module.py 파일의 138 번째 라인에서 정의되었습니다.

8.8.4.5 N

model.roi_module.Roi.N

roi_module.py 파일의 139 번째 라인에서 정의되었습니다.

8.8.4.6 rois

model.roi_module.Roi.rois

roi_module.py 파일의 142 번째 라인에서 정의되었습니다.

8.8.4.7 spatial_scale

`model.roi_module.RoI.spatial_scale`

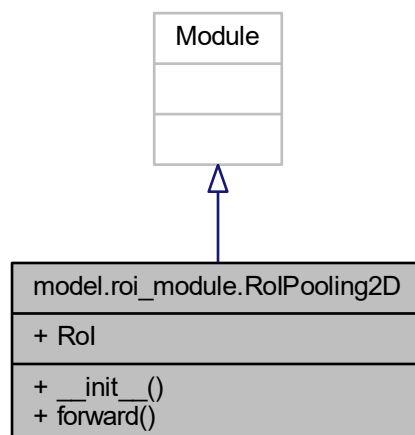
`roi_module.py` 파일의 132 번째 라인에서 정의되었습니다.

이 클래스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

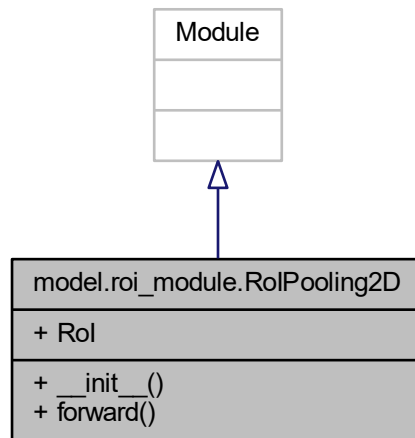
- [model/roi_module.py](#)

8.9 model.roi_module.RoIPooling2D 클래스 참조

`model.roi_module.RoIPooling2D`에 대한 상속 다이어그램 :



model.roi_module.RoIPooling2D에 대한 협력 다이어그램:



Public 멤버 함수

- def `__init__` (self, outh, outw, spatial_scale)
- def `forward` (self, x, rois)

Public 속성

- `RoI`

8.9.1 상세한 설명

roi_module.py 파일의 112 번째 라인에서 정의되었습니다.

8.9.2 생성자 & 소멸자 문서화

8.9.2.1 `__init__`()

```
def model.roi_module.RoIPooling2D.__init__(
    self,
    outh,
    outw,
    spatial_scale )
```

roi_module.py 파일의 114 번째 라인에서 정의되었습니다.

```
114 def __init__(self, outh, outw, spatial_scale):
115     super(RoIPooling2D, self).__init__()
116     self.RoI = RoI(outh, outw, spatial_scale)
117
```

8.9.3 멤버 함수 문서화

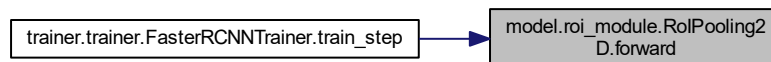
8.9.3.1 forward()

```
def model.roi_module.RoIPooling2D.forward (
    self,
    x,
    rois )
```

roi_module.py 파일의 120 번째 라인에서 정의되었습니다.

```
120     def forward(self, x, rois):
121         return self.RoI(x, rois)
122
123
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



8.9.4 멤버 데이터 문서화

8.9.4.1 RoI

```
model.roi_module.RoIPooling2D.RoI
```

roi_module.py 파일의 116 번째 라인에서 정의되었습니다.

이 클래스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- [model/roi_module.py](#)

8.10 data.dataset.TestDataset 클래스 참조

data.dataset.TestDataset에 대한 협력 다이어그램:

data.dataset.TestDataset
+ opt + db
+ __init__() + __getitem__() + __len__()

Public 멤버 함수

- def `__init__` (self, `opt`, split='test', use_difficult=True)
- def `__getitem__` (self, idx)
- def `__len__` (self)

Public 속성

- `opt`
- `db`

8.10.1 상세한 설명

dataset.py 파일의 175 번째 라인에서 정의되었습니다.

8.10.2 생성자 & 소멸자 문서화

8.10.2.1 `__init__()`

```
def data.dataset.TestDataset.__init__(
    self,
    opt,
    split = 'test',
    use_difficult = True )
```

dataset.py 파일의 176 번째 라인에서 정의되었습니다.

```
176 def __init__(self, opt, split='test', use_difficult=True):
177     self.opt = opt
178     self.db = VOCBboxDataset(opt.voc_data_dir, split=split, use_difficult=use_difficult)
179
```

8.10.3 멤버 함수 문서화

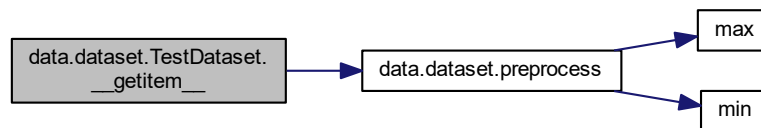
8.10.3.1 __getitem__()

```
def data.dataset.TestDataset.__getitem__(
    self,
    idx )
```

dataset.py 파일의 180 번째 라인에서 정의되었습니다.

```
180     def __getitem__(self, idx):
181         ori_img, bbox, label, difficult = self.db.get_example(idx)
182         img = preprocess(ori_img)
183         return img, ori_img.shape[1:], bbox, label, difficult
184
```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



8.10.3.2 __len__()

```
def data.dataset.TestDataset.__len__(
    self )
```

dataset.py 파일의 185 번째 라인에서 정의되었습니다.

```
185     def __len__(self):
186         return len(self.db)
```

8.10.4 멤버 데이터 문서화

8.10.4.1 db

```
data.dataset.TestDataset.db
```

dataset.py 파일의 178 번째 라인에서 정의되었습니다.

8.10.4.2 opt

data.dataset.TestDataset.opt

dataset.py 파일의 177 번째 라인에서 정의되었습니다.

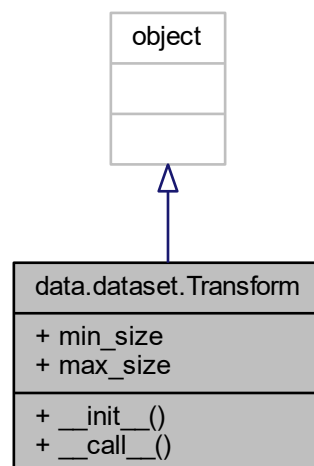
이 클래스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- data/[dataset.py](#)

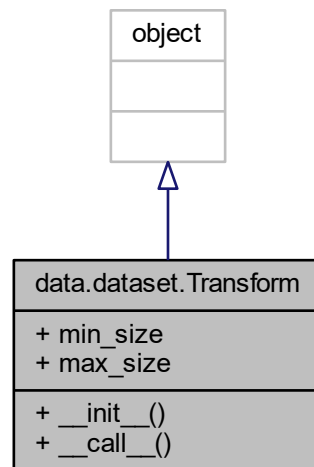
8.11 data.dataset.Transform 클래스 참조

이미지 크기 변경 및 B.B 크기 변경

data.dataset.Transform에 대한 상속 다이어그램 :



data.dataset.Transform에 대한 협력 다이어그램:



Public 멤버 함수

- def `__init__` (self, `min_size`=600, `max_size`=1000)
- def `__call__` (self, in_data)

Public 속성

- `min_size`
- `max_size`

8.11.1 상세한 설명

이미지 크기 변경 및 B.B 크기 변경

dataset.py 파일의 129 번째 라인에서 정의되었습니다.

8.11.2 생성자 & 소멸자 문서화

8.11.2.1 __init__()

```
def data.dataset.Transform.__init__(
    self,
    min_size = 600,
    max_size = 1000 )
```

dataset.py 파일의 131 번째 라인에서 정의되었습니다.

```
131 def __init__(self, min_size=600, max_size=1000):
132     self.min_size = min_size
133     self.max_size = max_size
134
```

8.11.3 멤버 함수 문서화

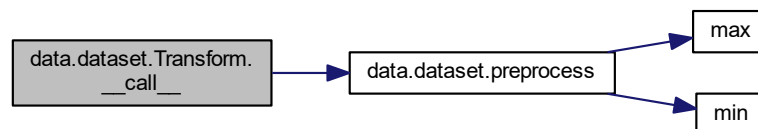
8.11.3.1 __call__()

```
def data.dataset.Transform.__call__(
    self,
    in_data )
```

dataset.py 파일의 135 번째 라인에서 정의되었습니다.

```
135 def __call__(self, in_data):
136     img, bbox, label = in_data
137     _, H, W = img.shape
138     # 이미지 resize
139     img = preprocess(img, self.min_size, self.max_size)
140     _, o_H, o_W = img.shape
141     scale = o_H / H
142     bbox = util.resize_bbox(bbox, (H, W), (o_H, o_W))
143
144     # horizontally flip
145     img, params = util.random_flip(
146         img, x_random=True, return_param=True)
147     bbox = util.flip_bbox(
148         bbox, (o_H, o_W), x_flip=params['x_flip'])
149
150     return img, bbox, label, scale
151
152
```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



8.11.4 멤버 데이터 문서화

8.11.4.1 max_size

`data.dataset.Transform.max_size`

`dataset.py` 파일의 133 번째 라인에서 정의되었습니다.

8.11.4.2 min_size

`data.dataset.Transform.min_size`

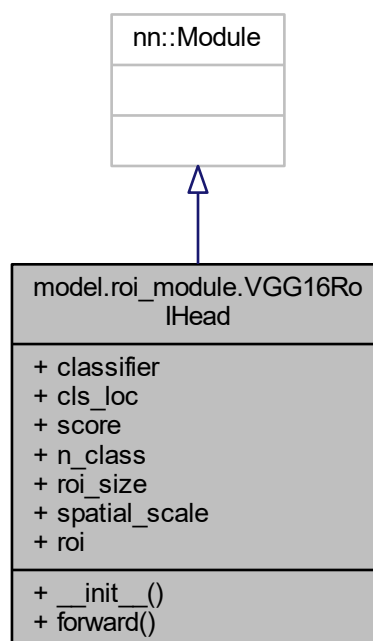
`dataset.py` 파일의 132 번째 라인에서 정의되었습니다.

이 클래스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

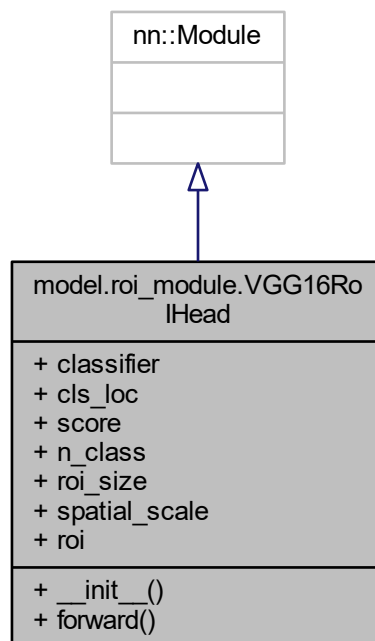
- [data/dataset.py](#)

8.12 model.roi_module.VGG16RoIHead 클래스 참조

`model.roi_module.VGG16RoIHead`에 대한 상속 다이어그램 :



model.roi_module.VGG16RoIHead에 대한 협력 다이어그램:



Public 멤버 함수

- def `__init__` (self, `n_class`, `roi_size`, `spatial_scale`, `classifier`)
- def `forward` (self, `x`, `rois`, `roi_indices`)
RoIPooling network

Public 속성

- `classifier`
- `cls_loc`
- `score`
- `n_class`
- `roi_size`
- `spatial_scale`
- `roi`

8.12.1 상세한 설명

Faster R-CNN Head for VGG-16 based implementation.
 @brief this class is used as a head for Faster R-CNN.
 @brief This outputs class-wise localizations and classification based on feature
 @brief maps in the given RoIs.
 @brief "RPN"에서 나온 ProposalCreator --> ProposalTargetCreator(return sample_rois)를 ROI pooling을 통해 7 * 7 만듬.

Args:

@param n_class (int): The number of classes possibly including the background.

@param roi_size (int): Height and width of the feature maps after RoI-pooling.

@param spatial_scale (float): Scale of the roi is resized.

@param classifier (nn.Module): Two layer Linear ported from vgg16

roi_module.py 파일의 23 번째 라인에서 정의되었습니다.

8.12.2 생성자 & 소멸자 문서화

8.12.2.1 __init__()

```
def model.roi_module.VGG16RoIHead.__init__(
    self,
    n_class,
    roi_size,
    spatial_scale,
    classifier )
```

roi_module.py 파일의 36 번째 라인에서 정의되었습니다.

```
36 def __init__(self, n_class, roi_size, spatial_scale,
37             classifier):
38     # n_class includes the background
39     super(VGG16RoIHead, self).__init__()
40     # Todo 네트워크 변경시 해당 부분도 함께 변경.
41     self.classifier = classifier.cuda()
42     self.cls_loc = nn.Linear(4096, n_class * 4).cuda()
43     self.score = nn.Linear(4096, n_class).cuda()
44
45     normal_init(self.cls_loc, 0, 0.001)
46     normal_init(self.score, 0, 0.01)
47
48     self.n_class = n_class
49     self.roi_size = roi_size
50     self.spatial_scale = spatial_scale
51     self.roi = RoIPooling2D(self.roi_size, self.roi_size, self.spatial_scale)
52
```

8.12.3 멤버 함수 문서화

8.12.3.1 forward()

```
def model.roi_module.VGG16RoIHead.forward (
    self,
    x,
    rois,
    roi_indices )
```

RoIPooling network

매개변수

x	model의 Feature map
roi	sample rois batch 적용 시 배치 크기만큼의 anchor 선택 (현재는 batch를 사용하지 않으므로 0)
roi_indices	roi (현재는 batch를 사용하지 않으므로 0)

반환값

roi_cls_locs : result of pooling classification

roi_scores : result of pooling score

Forward the chain.

We assume that there are N batches.

Args:

x (Variable): 4D image variable.

rois (Tensor): A bounding box array containing coordinates of proposal boxes. This is a concatenation of bounding box arrays from multiple images in the batch.

Its shape is $(R', 4)$. Given R_i proposed

RoIs from the i th image,

$R' = \sum_{i=1}^N R_i$.

roi_indices (Tensor): An array containing indices of images to which bounding boxes correspond to. Its shape is $(R',)$.

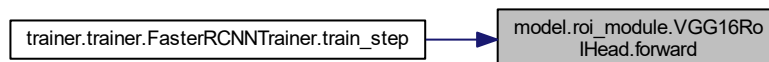
roi_module.py 파일의 61 번째 라인에서 정의되었습니다.

```

61 def forward(self, x, rois, roi_indices):
62     """Forward the chain.
63
64     We assume that there are  $N$  batches.
65
66     Args:
67         x (Variable): 4D image variable.
68         rois (Tensor): A bounding box array containing coordinates of
69         proposal boxes. This is a concatenation of bounding box
70         arrays from multiple images in the batch.
71         Its shape is  $(R', 4)$ . Given  $R_i$  proposed
72         RoIs from the  $i$ th image,
73          $R' = \sum_{i=1}^N R_i$ .
74         roi_indices (Tensor): An array containing indices of images to
75         which bounding boxes correspond to. Its shape is  $(R',)$ .
76
77     """
78     # in case roi_indices is ndarray
79     roi_indices = at.totensor(roi_indices).float()
80     rois = at.totensor(rois).float()
81     indices_and_rois = t.cat([roi_indices[:, None], rois], dim=1) # 128 * (1(배경) + 4)
82     # @NOTE: important: yx->xy
83     xy_indices_and_rois = indices_and_rois[:, [0, 2, 1, 4, 3]]
84     indices_and_rois = t.autograd.Variable(xy_indices_and_rois.contiguous())
85
86     # RoIPooling
87     pool = self.roi(x, indices_and_rois)
88     pool = pool.view(pool.size(0), -1)
89     fc7 = self.classifier(pool)
90     roi_cls_locs = self.cls_loc(fc7)
91     roi_scores = self.score(fc7)
92     return roi_cls_locs, roi_scores
93

```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



8.12.4 멤버 데이터 문서화

8.12.4.1 classifier

`model.roi_module.VGG16RoIHead.classifier`

`roi_module.py` 파일의 40 번째 라인에서 정의되었습니다.

8.12.4.2 cls_loc

`model.roi_module.VGG16RoIHead.cls_loc`

`roi_module.py` 파일의 41 번째 라인에서 정의되었습니다.

8.12.4.3 n_class

`model.roi_module.VGG16RoIHead.n_class`

`roi_module.py` 파일의 47 번째 라인에서 정의되었습니다.

8.12.4.4 roi

`model.roi_module.VGG16RoIHead.roi`

`roi_module.py` 파일의 50 번째 라인에서 정의되었습니다.

8.12.4.5 roi_size

`model.roi_module.VGG16RoIHead.roi_size`

`roi_module.py` 파일의 48 번째 라인에서 정의되었습니다.

8.12.4.6 score

model.roi_module.VGG16RoIHead.score

roi_module.py 파일의 42 번째 라인에서 정의되었습니다.

8.12.4.7 spatial_scale

model.roi_module.VGG16RoIHead.spatial_scale

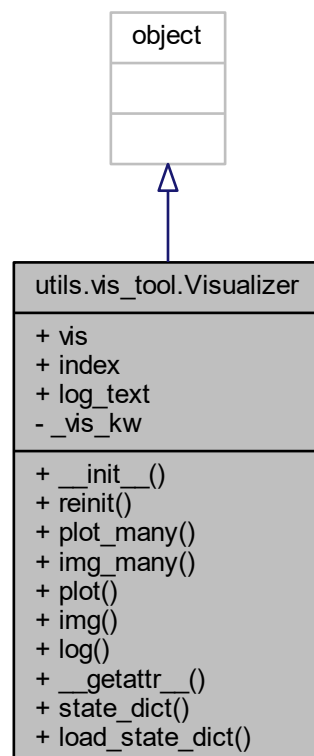
roi_module.py 파일의 49 번째 라인에서 정의되었습니다.

이 클래스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

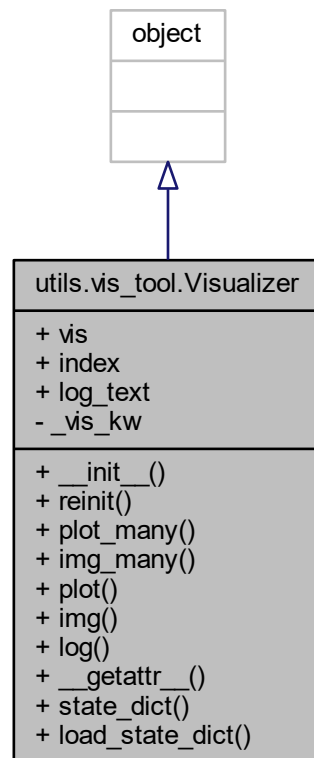
- [model/roi_module.py](#)

8.13 utils.vis_tool.Visualizer 클래스 참조

utils.vis_tool.Visualizer에 대한 상속 다이어그램 :



utils.vis_tool.Visualizer에 대한 협력 다이어그램:



Public 멤버 함수

- def `__init__` (self, env='default', **kwargs)
- def `reinit` (self, env='default', **kwargs)
- def `plot_many` (self, `d`)
- def `img_many` (self, `d`)
- def `plot` (self, name, y, **kwargs)
- def `img` (self, name, img_, **kwargs)
- def `log` (self, info, win='log_text')
- def `__getattr__` (self, name)
- def `state_dict` (self)
- def `load_state_dict` (self, `d`)

Public 속성

- `vis`
- `index`
- `log_text`

Private 속성

- `_vis_kw`

8.13.1 상세한 설명

wrapper for visdom
you can still access naive visdom function by
self.line, self.scatter, self._send, etc.
due to the implementation of `__getattr__`

vis_tool.py 파일의 153 번째 라인에서 정의되었습니다.

8.13.2 생성자 & 소멸자 문서화

8.13.2.1 `__init__()`

```
def utils.vis_tool.Visualizer.__init__(
    self,
    env = 'default',
    **kwargs )
```

vis_tool.py 파일의 161 번째 라인에서 정의되었습니다.

```
161     def __init__(self, env='default', **kwargs):
162         self.vis = visdom.Vision(env=env, use_incoming_socket=False, **kwargs)
163         self._vis_kw = kwargs
164
165         # e.g. ('loss', 23) the 23th value of loss
166         self.index = {}
167         self.log_text = ""
168
```

8.13.3 멤버 함수 문서화

8.13.3.1 `__getattr__()`

```
def utils.vis_tool.Visualizer.__getattr__(
    self,
    name )
```

vis_tool.py 파일의 225 번째 라인에서 정의되었습니다.

```
225     def __getattr__(self, name):
226         return getattr(self.vis, name)
227
```

8.13.3.2 img()

```
def utils.vis_tool.Visualizer.img (
    self,
    name,
    img_,
    ** kwargs )

self.img('input_img', t.Tensor(64, 64))
self.img('input_imgs', t.Tensor(3, 64, 64))
self.img('input_imgs', t.Tensor(100, 1, 64, 64))
self.img('input_imgs', t.Tensor(100, 3, 64, 64), nrows=10)
!!don't ~~self.img('input_imgs', t.Tensor(100, 64, 64), nrows=10)~~!!
```

vis_tool.py 파일의 202 번째 라인에서 정의되었습니다.

```
202 def img(self, name, img_, **kwargs):
203     """
204     self.img('input_img', t.Tensor(64, 64))
205     self.img('input_imgs', t.Tensor(3, 64, 64))
206     self.img('input_imgs', t.Tensor(100, 1, 64, 64))
207     self.img('input_imgs', t.Tensor(100, 3, 64, 64), nrows=10)
208     !!don't self.img('input_imgs', t.Tensor(100, 64, 64), nrows=10) !!
209     """
210     self.vis.images(t.Tensor(img_).cpu().numpy(),
211                    win=name,
212                    opts=dict(title=name),
213                    **kwargs
214                    )
215
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



8.13.3.3 img_many()

```
def utils.vis_tool.Visualizer.img_many (
    self,
    d )
```

vis_tool.py 파일의 185 번째 라인에서 정의되었습니다.

```
185 def img_many(self, d):
186     for k, v in d.items():
187         self.img(k, v)
188
```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



8.13.3.4 load_state_dict()

```
def utils.vis_tool.Visualizer.load_state_dict (
    self,
    d )
```

vis_tool.py 파일의 236 번째 라인에서 정의되었습니다.

```
236     def load_state_dict(self, d):
237         self.vis = visdom.Visdom(env=d.get('env', self.vis.env), **(self.d.get('vis_kw')))
238         self.log_text = d.get('log_text', "")
239         self.index = d.get('index', dict())
240         return self
```

8.13.3.5 log()

```
def utils.vis_tool.Visualizer.log (
    self,
    info,
    win = 'log_text' )
```

```
self.log({'loss':1,'lr':0.0001})
```

vis_tool.py 파일의 216 번째 라인에서 정의되었습니다.

```
216     def log(self, info, win='log_text'):
217         """
218         self.log({'loss':1,'lr':0.0001})
219         """
220         self.log_text += ('[{time}] {info} <br>'.format(
221             time=time.strftime('%m%d_%H%M%S'), \
222             info=info))
223         self.vis.text(self.log_text, win)
224
```

8.13.3.6 plot()

```
def utils.vis_tool.Visualizer.plot (
    self,
    name,
    y,
    ** kwargs )
```

```
self.plot('loss',1.00)
```

vis_tool.py 파일의 189 번째 라인에서 정의되었습니다.

```
189     def plot(self, name, y, **kwargs):
190         """
191         self.plot('loss',1.00)
192         """
193         x = self.index.get(name, 0)
194         self.vis.line(Y=np.array([y]), X=np.array([x]),
195                      win=name,
196                      opts=dict(title=name),
197                      update=None if x == 0 else 'append',
198                      **kwargs
199                      )
200         self.index[name] = x + 1
```

201

이 함수를 호출하는 함수들에 대한 그래프입니다.:



8.13.3.7 plot_many()

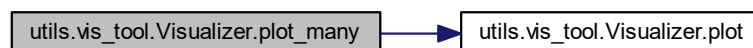
```
def utils.vis_tool.Visualizer.plot_many (
    self,
    d )
```

```
plot multi values
@params d: dict (name,value) i.e. ('loss',0.11)
```

vis_tool.py 파일의 176 번째 라인에서 정의되었습니다.

```
176     def plot_many(self, d):
177         """
178         plot multi values
179         @params d: dict (name,value) i.e. ('loss',0.11)
180         """
181         for k, v in d.items():
182             if v is not None:
183                 self.plot(k, v)
184
```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



8.13.3.8 reinit()

```
def utils.vis_tool.Visualizer.reinit (
    self,
    env = 'default',
    ** kwargs )
```

change the config of visdom

vis_tool.py 파일의 169 번째 라인에서 정의되었습니다.

```
169     def reinit(self, env='default', **kwargs):
170         """
171         change the config of visdom
172         """
173         self.vis = visdom.Visdom(env=env, **kwargs)
174         return self
175
```

8.13.3.9 state_dict()

```
def utils.vis_tool.Visualizer.state_dict (
    self )
```

vis_tool.py 파일의 228 번째 라인에서 정의되었습니다.

```
228     def state_dict(self):
229         return {
230             'index': self.index,
231             'vis_kw': self._vis_kw,
232             'log_text': self.log_text,
233             'env': self.vis.env
234         }
235
```

8.13.4 멤버 데이터 문서화

8.13.4.1 _vis_kw

utils.vis_tool.Visualizer._vis_kw [private]

vis_tool.py 파일의 163 번째 라인에서 정의되었습니다.

8.13.4.2 index

utils.vis_tool.Visualizer.index

vis_tool.py 파일의 166 번째 라인에서 정의되었습니다.

8.13.4.3 log_text

`utils.vis_tool.Visualizer.log_text`

`vis_tool.py` 파일의 167 번째 라인에서 정의되었습니다.

8.13.4.4 vis

`utils.vis_tool.Visualizer.vis`

`vis_tool.py` 파일의 162 번째 라인에서 정의되었습니다.

이 클래스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- [utils/vis_tool.py](#)

8.14 data.voc_dataset.VOCBboxDataset 클래스 참조

`data.voc_dataset.VOCBboxDataset`에 대한 협력 다이어그램:

data.voc_dataset.VOCBbox Dataset
+ ids + data_dir + use_difficult + return_difficult + label_names - __getitem__
+ __init__() + __len__() + get_example()

Public 멤버 함수

- `def __init__(self, data_dir, split='train_laundry', use_difficult=False, return_difficult=False)`
- `def __len__(self)`
- `def get_example(self, i)`
input image에 대한 정보 (B.B, label)를 return 해주는 함수

Public 속성

- [ids](#)
- [data_dir](#)
- [use_difficult](#)
- [return_difficult](#)
- [label_names](#)

정적 Private 속성

- `def __getitem__ = get_example`

8.14.1 상세한 설명

Bounding box dataset for PASCAL 'VOC'.

```
.. _'VOC': http://host.robots.ox.ac.uk/pascal/VOC/voc2012/
```

The index corresponds to each image.

When queried by an index, if `:obj:'return_difficult' == False`, this dataset returns a corresponding `:obj:'img, bbox, label'`, a tuple of an image, bounding boxes and labels. This is the default behaviour. If `:obj:'return_difficult' == True`, this dataset returns corresponding `:obj:'img, bbox, label, difficult'`. `:obj:'difficult'` is a boolean array that indicates whether bounding boxes are labeled as difficult or not.

The bounding boxes are packed into a two dimensional tensor of shape `:math:'(R, 4)'`, where `:math:'R'` is the number of bounding boxes in the image. The second axis represents attributes of the bounding box. They are `:math:'(y_{min}, x_{min}, y_{max}, x_{max})'`, where the four attributes are coordinates of the top left and the bottom right vertices.

The labels are packed into a one dimensional tensor of shape `:math:'(R,)'`. `:math:'R'` is the number of bounding boxes in the image. The class name of the label `:math:'l'` is `:math:'l'`th element of `:obj:'VOC_BBOX_LABEL_NAMES'`.

The array `:obj:'difficult'` is a one dimensional boolean array of shape `:math:'(R,)'`. `:math:'R'` is the number of bounding boxes in the image. If `:obj:'use_difficult'` is `:obj:'False'`, this array is a boolean array with all `:obj:'False'`.

The type of the image, the bounding boxes and the labels are as follows.

```
* :obj:'img.dtype' == numpy.float32`
* :obj:'bbox.dtype' == numpy.float32`
* :obj:'label.dtype' == numpy.int32`
* :obj:'difficult.dtype' == numpy.bool`
```

Args:

```
data_dir (string): Path to the root of the training data.
    i.e. "/data/image/voc/VOCdevkit/VOC2007/"
split ({'train', 'val', 'trainval', 'test'}): Select a split of the
    dataset. :obj:'test' split is only available for
    2007 dataset.
year ({'2007', '2012'}): Use a dataset prepared for a challenge
    held in :obj:'year'.
use_difficult (bool): If :obj:'True', use images that are labeled as
    difficult in the original annotation.
return_difficult (bool): If :obj:'True', this dataset returns
    a boolean array
    that indicates whether bounding boxes are labeled as difficult
    or not. The default value is :obj:'False'.
```

voc_dataset.py 파일의 8 번째 라인에서 정의되었습니다.

8.14.2 생성자 & 소멸자 문서화

8.14.2.1 __init__()

```
def data.voc_dataset.VOCBboxDataset.__init__(
    self,
    data_dir,
    split = 'train_laundry',
    use_difficult = False,
    return_difficult = False )
```

@param data_dir: 데이터 파일 위치
 @param split: 해당 이름으로 학습하고자 하는 파일 이름 선택.
 @param use_difficult: False
 @param return_difficult: False

voc_dataset.py 파일의 64 번째 라인에서 정의되었습니다.

```
64 def __init__(self, data_dir, split='train_laundry',
65             use_difficult=False, return_difficult=False,
66             ):
67
68     """
69     @param data_dir: 데이터 파일 위치
70     @param split: 해당 이름으로 학습하고자 하는 파일 이름 선택.
71     @param use_difficult: False
72     @param return_difficult: False
73     """
74
75     id_list_file = os.path.join(
76         data_dir, 'ImageSets/Main/{0}.txt'.format(split))
77
78     self.ids = [id.strip() for id in open(id_list_file)]
79     self.data_dir = data_dir
80     self.use_difficult = use_difficult
81     self.return_difficult = return_difficult
82     self.label_names = VOC_BBOX_LABEL_NAMES
83
```

8.14.3 멤버 함수 문서화

8.14.3.1 __len__()

```
def data.voc_dataset.VOCBboxDataset.__len__(
    self )
```

voc_dataset.py 파일의 84 번째 라인에서 정의되었습니다.

```
84 def __len__(self):
85     return len(self.ids)
```

8.14.3.2 get_example()

```
def data.voc_dataset.VOCBboxDataset.get_example (
    self,
    i )
```

input image에 대한 정보 (B.B, label)를 return 해주는 함수

매개변수

i	이미지 이름 (숫자로 구성되어 있음 ex)000001)
---	--------------------------------

반환값

img : 이미지

bbox : B.B 정보

label : label 정보

difficult

Returns the i-th example.

Returns a color image and bounding boxes. The image is in CHW format.
The returned image is RGB.

Args:

i (int): The index of the example.

Returns:

tuple of an image and bounding boxes

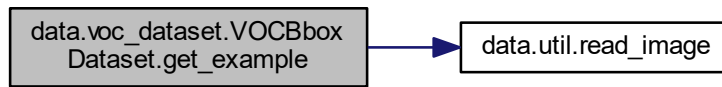
voc_dataset.py 파일의 94 번째 라인에서 정의되었습니다.

```

94     def get_example(self, i):
95         """Returns the i-th example.
96
97         Returns a color image and bounding boxes. The image is in CHW format.
98         The returned image is RGB.
99
100        Args:
101            i (int): The index of the example.
102
103        Returns:
104            tuple of an image and bounding boxes
105
106        """
107        id_ = self.ids[i]
108        anno = ET.parse(
109            os.path.join(self.data_dir, 'dataset_laundry tag_xml', id_ + '.xml'))
110        bbox = list()
111        label = list()
112        difficult = list()
113        for obj in anno.findall('object'):
114            # when in not using difficult split, and the object is
115            # difficult, skip it.
116            if not self.use_difficult and int(obj.find('difficult').text) == 1:
117                continue
118
119            difficult.append(int(obj.find('difficult').text))
120            bndbox_anno = obj.find('bndbox')
121            # subtract 1 to make pixel indexes 0-based
122            bbox.append([
123                int(bndbox_anno.find(tag).text) - 1
124                for tag in ('ymin', 'xmin', 'ymax', 'xmax')])
125            name = obj.find('name').text.lower().strip()
126
127            label.append(VOC_BBOX_LABEL_NAMES.index(name))
128
129        bbox = np.stack(bbox).astype(np.float32)
130        label = np.stack(label).astype(np.int32)
131        # When 'use_difficult==False', all elements in 'difficult' are False.
132        difficult = np.array(difficult, dtype=np.bool).astype(np.uint8) # PyTorch don't support np.bool
133
134        # Load a image
135        img_file = os.path.join(self.data_dir, 'dataset_laundry tag', id_ + '.jpg')
136        img = read_image(img_file, color=True)
137
138        return img, bbox, label, difficult
139

```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



8.14.4 멤버 데이터 문서화

8.14.4.1 `__getitem__`

```
def data.voc_dataset.VOCBboxDataset.__getitem__ = get\_example [static], [private]
```

`voc_dataset.py` 파일의 140 번째 라인에서 정의되었습니다.

8.14.4.2 `data_dir`

```
data.voc_dataset.VOCBboxDataset.data_dir
```

`voc_dataset.py` 파일의 77 번째 라인에서 정의되었습니다.

8.14.4.3 `ids`

```
data.voc_dataset.VOCBboxDataset.ids
```

`voc_dataset.py` 파일의 76 번째 라인에서 정의되었습니다.

8.14.4.4 `label_names`

```
data.voc_dataset.VOCBboxDataset.label_names
```

`voc_dataset.py` 파일의 80 번째 라인에서 정의되었습니다.

8.14.4.5 return_difficult

data.voc_dataset.VOCBboxDataset.return_difficult

voc_dataset.py 파일의 79 번째 라인에서 정의되었습니다.

8.14.4.6 use_difficult

data.voc_dataset.VOCBboxDataset.use_difficult

voc_dataset.py 파일의 78 번째 라인에서 정의되었습니다.

이 클래스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- data/[voc_dataset.py](#)

8.15 utils.data_load.VOCBboxDataset 클래스 참조

utils.data_load.VOCBboxDataset에 대한 협력 다이어그램:

utils.data_load.VOCBboxDataset
+ ids + data_dir + use_difficult + return_difficult + label_names - __getitem__
+ __init__() + __len__() + get_example()

Public 멤버 함수

- def [__init__](#) (self, [data_dir](#), split='trainval', [use_difficult](#)=False, [return_difficult](#)=False)
- def [__len__](#) (self)
- def [get_example](#) (self, i)

Public 속성

- [ids](#)
- [data_dir](#)
- [use_difficult](#)
- [return_difficult](#)
- [label_names](#)

정적 Private 속성

- `def __getitem__ = get_example`

8.15.1 상세한 설명

Bounding box dataset for PASCAL 'VOC'__.

```
.. _'VOC': http://host.robots.ox.ac.uk/pascal/VOC/voc2012/
```

The index corresponds to each image.

When queried by an index, if `:obj:'return_difficult' == False`, this dataset returns a corresponding `:obj:'img, bbox, label'`, a tuple of an image, bounding boxes and labels. This is the default behaviour. If `:obj:'return_difficult' == True`, this dataset returns corresponding `:obj:'img, bbox, label, difficult'`. `:obj:'difficult'` is a boolean array that indicates whether bounding boxes are labeled as difficult or not.

The bounding boxes are packed into a two dimensional tensor of shape `:math:'(R, 4)'`, where `:math:'R'` is the number of bounding boxes in the image. The second axis represents attributes of the bounding box. They are `:math:'(y_{min}, x_{min}, y_{max}, x_{max})'`, where the four attributes are coordinates of the top left and the bottom right vertices.

The labels are packed into a one dimensional tensor of shape `:math:'(R,)'`. `:math:'R'` is the number of bounding boxes in the image. The class name of the label `:obj:'l'` is `:math:'l'`th element of `:obj:'VOC_BBOX_LABEL_NAMES'`.

The array `:obj:'difficult'` is a one dimensional boolean array of shape `:math:'(R,)'`. `:math:'R'` is the number of bounding boxes in the image. If `:obj:'use_difficult'` is `:obj:'False'`, this array is a boolean array with all `:obj:'False'`.

The type of the image, the bounding boxes and the labels are as follows.

```
* :obj:'img.dtype == numpy.float32'
* :obj:'bbox.dtype == numpy.float32'
* :obj:'label.dtype == numpy.int32'
* :obj:'difficult.dtype == numpy.bool'
```

Args:

```
data_dir (string): Path to the root of the training data.
    i.e. "/data/image/voc/VOCdevkit/VOC2007/"
split ({'train', 'val', 'trainval', 'test'}): Select a split of the
    dataset. :obj:'test' split is only available for
    2007 dataset.
year ({'2007', '2012'}): Use a dataset prepared for a challenge
    held in :obj:'year'.
use_difficult (bool): If :obj:'True', use images that are labeled as
    difficult in the original annotation.
return_difficult (bool): If :obj:'True', this dataset returns
    a boolean array
    that indicates whether bounding boxes are labeled as difficult
    or not. The default value is :obj:'False'.
```

data_load.py 파일의 29 번째 라인에서 정의되었습니다.

8.15.2 생성자 & 소멸자 문서화

8.15.2.1 __init__()

```
def utils.data_load.VOCBboxDataset.__init__(
    self,
    data_dir,
    split = 'trainval',
    use_difficult = False,
    return_difficult = False )
```

data_load.py 파일의 85 번째 라인에서 정의되었습니다.

```
85     def __init__(self, data_dir, split='trainval',
86                 use_difficult=False, return_difficult=False,
87                 ):
88
89         id_list_file = os.path.join(
90             data_dir, 'ImageSets/Main/{0}.txt'.format(split))
91
92         self.ids = [id_.strip() for id_ in open(id_list_file)]
93         self.data_dir = data_dir
94         self.use_difficult = use_difficult
95         self.return_difficult = return_difficult
96         self.label_names = VOC_BBOX_LABEL_NAMES
97
```

8.15.3 멤버 함수 문서화

8.15.3.1 __len__()

```
def utils.data_load.VOCBboxDataset.__len__(
    self )
```

data_load.py 파일의 98 번째 라인에서 정의되었습니다.

```
98     def __len__(self):
99         return len(self.ids)
100
```

8.15.3.2 get_example()

```
def utils.data_load.VOCBboxDataset.get_example (
    self,
    i )
```

Returns the i-th example.

Returns a color image and bounding boxes. The image is in CHW format.
The returned image is RGB.

Args:

i (int): The index of the example.

Returns:

tuple of an image and bounding boxes
img: RGB image with shape [H, W, C], type float32

data_load.py 파일의 101 번째 라인에서 정의되었습니다.

```

101 def get_example(self, i):
102     """Returns the i-th example.
103
104     Returns a color image and bounding boxes. The image is in CHW format.
105     The returned image is RGB.
106
107     Args:
108         i (int): The index of the example.
109
110     Returns:
111         tuple of an image and bounding boxes
112         img: RGB image with shape [H, W, C], type float32
113
114     """
115     id_ = self.ids[i]
116     anno = ET.parse(
117         os.path.join(self.data_dir, 'Annotations', id_ + '.xml'))
118     bbox = list()
119     label = list()
120     difficult = list()
121     for obj in anno.findall('object'):
122         # when in not using difficult split, and the object is
123         # difficult, skip it.
124         if not self.use_difficult and int(obj.find('difficult').text) == 1:
125             continue
126
127         difficult.append(int(obj.find('difficult').text))
128         bndbox_anno = obj.find('bndbox')
129         # subtract 1 to make pixel indexes 0-based
130         bbox.append([
131             int(bndbox_anno.find(tag).text) - 1
132             for tag in ('ymin', 'xmin', 'ymax', 'xmax')])
133         name = obj.find('name').text.lower().strip()
134         label.append(VOC_BBOX_LABEL_NAMES.index(name))
135         bbox = np.stack(bbox).astype(np.float32)
136         label = np.stack(label).astype(np.int32)
137         # When 'use_difficult==False', all elements in 'difficult' are False.
138         difficult = np.array(difficult, dtype=np.bool).astype(np.uint8) # PyTorch don't support np.bool
139
140         # Load a image
141         img_file = os.path.join(self.data_dir, 'JPEGImages', id_ + '.jpg')
142         img = plt.imread(img_file)
143         img = img.astype(np.float32)
144
145         # if self.return_difficult:
146         #     return img, bbox, label, difficult
147         return img, bbox, label, difficult
148

```

8.15.4 멤버 데이터 문서화

8.15.4.1 __getitem__

```
def utils.data_load.VOCBboxDataset.__getitem__ = get_example [static], [private]
```

data_load.py 파일의 149 번째 라인에서 정의되었습니다.

8.15.4.2 data_dir

utils.data_load.VOCBboxDataset.data_dir

data_load.py 파일의 91 번째 라인에서 정의되었습니다.

8.15.4.3 ids

utils.data_load.VOCBboxDataset.ids

data_load.py 파일의 90 번째 라인에서 정의되었습니다.

8.15.4.4 label_names

utils.data_load.VOCBboxDataset.label_names

data_load.py 파일의 94 번째 라인에서 정의되었습니다.

8.15.4.5 return_difficult

utils.data_load.VOCBboxDataset.return_difficult

data_load.py 파일의 93 번째 라인에서 정의되었습니다.

8.15.4.6 use_difficult

utils.data_load.VOCBboxDataset.use_difficult

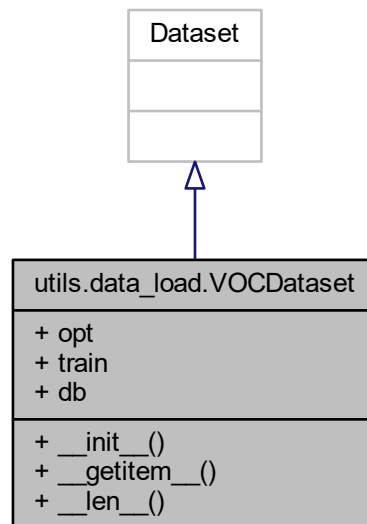
data_load.py 파일의 92 번째 라인에서 정의되었습니다.

이 클래스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

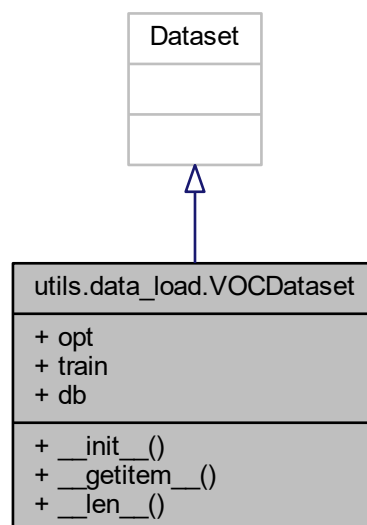
- [utils/data_load.py](#)

8.16 utils.data_load.VOCDataSet 클래스 참조

utils.data_load.VOCDataSet에 대한 상속 다이어그램 :



utils.data_load.VOCDataSet에 대한 협력 다이어그램:



Public 멤버 함수

- def `__init__` (self, `opt`, `train`=True)
- def `__getitem__` (self, `idx`)
- def `__len__` (self)

Public 속성

- `opt`
- `train`
- `db`

8.16.1 상세한 설명

```

returned image:
scaled image (mean, std, /255), float32, HWC, RGB
mean=[0.485, 0.456, 0.406]
std=[0.229, 0.224, 0.225]

```

data_load.py 파일의 178 번째 라인에서 정의되었습니다.

8.16.2 생성자 & 소멸자 문서화

8.16.2.1 `__init__()`

```

def utils.data_load.VOCDataset.__init__(
    self,
    opt,
    train = True )

```

data_load.py 파일의 185 번째 라인에서 정의되었습니다.

```

185 def __init__(self, opt, train=True):
186     self.opt = opt
187     self.train = train
188     if train:
189         self.db = VOCBboxDataset(opt.voc_data_dir)
190     else:
191         self.db = VOCBboxDataset(opt.voc_data_dir, split='test', use_difficult=True)
192

```

8.16.3 멤버 함수 문서화

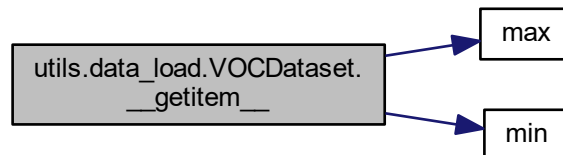
8.16.3.1 `__getitem__()`

```
def utils.data_load.VOCDataSet.__getitem__(
    self,
    idx )
```

`data_load.py` 파일의 193 번째 라인에서 정의되었습니다.

```
193     def __getitem__(self, idx):
194         ori_img, bbox, label, difficult = self.db.get_example(idx)
195
196         # RESCALE -----
197         # image rescale to [opt.min_size, opt.max_size]
198         H0, W0, C = ori_img.shape
199         scale = min(self.opt.min_size/min(H0, W0), self.opt.max_size/max(H0, W0))
200         scaled_img = cv2.resize(ori_img, (0,0), fx=scale, fy=scale)
201         H1, W1, _ = scaled_img.shape
202         # bbox rescale
203         bbox = scale * bbox
204
205         # NORMALIZE -----
206         normalized_img = scaled_img / 255.0
207         normalized_img = (normalized_img - np.array([0.485, 0.456, 0.406])) / np.array([0.229, 0.224,
0.225])
208
209         # HORIZON FLIP -----
210         if self.train and random.random() < 0.5:
211             normalized_img = normalized_img[:,::-1,:]
212             bbox[:,1], bbox[:,3] = W1 - bbox[:, 3], W1 - bbox[:, 1]
213         return normalized_img.astype(np.float32), bbox, label, scale
214
```

이 함수 내부에서 호출하는 함수들에 대한 그래프입니다.:



8.16.3.2 `__len__()`

```
def utils.data_load.VOCDataSet.__len__(
    self )
```

`data_load.py` 파일의 215 번째 라인에서 정의되었습니다.

```
215     def __len__(self):
216         return len(self.db)
```

8.16.4 멤버 데이터 문서화

8.16.4.1 `db`

`utils.data_load.VOCDataset.db`

`data_load.py` 파일의 189 번째 라인에서 정의되었습니다.

8.16.4.2 `opt`

`utils.data_load.VOCDataset.opt`

`data_load.py` 파일의 186 번째 라인에서 정의되었습니다.

8.16.4.3 `train`

`utils.data_load.VOCDataset.train`

`data_load.py` 파일의 187 번째 라인에서 정의되었습니다.

이 클래스에 대한 문서화 페이지는 다음의 파일로부터 생성되었습니다.:

- `utils/`[data_load.py](#)

Chapter 9

파일 문서화

9.1 data/__init__.py 파일 참조

네임스페이스

- [data](#)

9.2 model/__init__.py 파일 참조

네임스페이스

- [model](#)

9.3 model/utils/__init__.py 파일 참조

네임스페이스

- [model.utils](#)

9.4 model/utils/nms/__init__.py 파일 참조

네임스페이스

- [model.utils.nms](#)

9.5 trainer/__init__.py 파일 참조

네임스페이스

- [trainer](#)

9.6 utils/__init__.py 파일 참조

네임스페이스

- [utils](#)

9.7 data/dataset.py 파일 참조

클래스

- class [data.dataset.Transform](#)
이미지 크기 변경 및 B.B 크기 변경
- class [data.dataset.Dataset](#)
- class [data.dataset.TestDataset](#)

네임스페이스

- [data.dataset](#)

함수

- def [data.dataset.AddGaussianNoise](#) (img, noise_val)
- def [data.dataset.ChangeLuminance](#) (img, contrast_range, brightness_range)
- def [data.dataset.blur](#) (img)
- def [data.dataset.inverse_normalize](#) (img)
- def [data.dataset.pytorch_normalize](#) (img)
- def [data.dataset.caffe_normalize](#) (img)
- def [data.dataset.preprocess](#) (img, min_size=600, max_size=1000)
size가 다른 이미지에 대해서 resize 및 normalization

9.8 data/util.py 파일 참조

네임스페이스

- [data.util](#)

함수

- def [data.util.read_image](#) (path, dtype=np.float32, color=True)
- def [data.util.resize_bbox](#) (bbox, in_size, out_size)
- def [data.util.flip_bbox](#) (bbox, size, y_flip=False, x_flip=False)
- def [data.util.crop_bbox](#) (bbox, y_slice=None, x_slice=None, allow_outside_center=True, return_param=False)
- def [data.util._slice_to_bounds](#) (slice_)
- def [data.util.translate_bbox](#) (bbox, y_offset=0, x_offset=0)
- def [data.util.random_flip](#) (img, y_random=False, x_random=False, return_param=False, copy=False)

9.9 data/voc_dataset.py 파일 참조

클래스

- class `data.voc_dataset.VOCBboxDataset`

네임스페이스

- `data.voc_dataset`

변수

- tuple `data.voc_dataset.VOC_BBOX_LABEL_NAMES`

9.10 data/VOCdevkit/create_segmentations_from_detections.m 파일 참조

함수

- Creates segmentation `results` from detection `results` `CREATE_SEGMENTATIONS_FROM_DETECTIONS` (ID) creates `segmentations` from % the detection `results` with identifier ID e.g. 'comp3'. All detections % will be used
- Creates segmentation `results` from detection `results` no matter what their `confidence` level `CREATE_SEGMENTATIONS_FROM_DETECTIONS` (ID, CONFIDENCE) as above
- Creates segmentation `results` from detection `results` no matter what their `confidence` level but only detections above the specified `confidence` will be used `function create_segmentations_from_detections` (id, confidence) if nargin< 2 confidence
- `end` change this path if you install the VOC code elsewhere `addpath` ([cd '/VOCcode'])
- `if ~exist (resultsfile, 'file') error('Could not find detection results file to use to create segmentations(%s not found)'`
- `if ~strcmp (imgid, previd) ind`
- `if isempty (ind) imgids`
- `end im (ind).det(detnum)`
- `if ~exist (resultsdir, 'dir') mkdir(resultsdir)`
- `end if ~exist (resultsdirinst, 'dir') mkdir(resultsdirinst)`
- `imwrite (instim, cmap, instlabelfile)`
- `imwrite (classim, cmap, classlabelfile)`
- `class png ()`
- `copyfile (gtlabelfile, gtclasslabelfile)`
- `zeros([H W] uint8 ())`

변수

- initialize VOC options `VOCinit`
- load detection `results tic`
- `imgids = {}`
- for `clsnum`
- if `resultsfile`
- end `[ids, confs, b1, b2, b3, b4] = textread(resultsfile, '%s %f %f %f %f %f')`
- `BBOXS = [b1 b2 b3 b4]`
- `previd = ''`
- for `j`
- drawnow
- end `imgid = ids{j}`
- `conf = confs(j)`
- detinfo `bbox = BBOXS(j,:)`
- `ind = numel(imgids)`
- `detnum = 1`
- end end Write out the `segmentations resultsdir = sprintf(VOCopts.seg.clsresdir, id, VOCopts.testset)`
- `resultsdirinst = sprintf(VOCopts.seg.instresdir, id, VOCopts.testset)`
- end `cmap = VOClabelcolormap(255)`
- end `imname = imgids{j}`
- `classlabelfile = sprintf(VOCopts.seg.clsrespath, id, VOCopts.testset, imname)`
- `instlabelfile = sprintf(VOCopts.seg.instrespath, id, VOCopts.testset, imname)`
- `imgfile = sprintf(VOCopts.imgpath, imname)`
- `imginfo = imfinfo(imgfile)`
- `gtclasslabelfile = sprintf('%s/%d_gt.png', resultsdir, imnums(j))`
- `classim = uint8(zeros([H W]))`

9.10.1 함수 문서화

9.10.1.1 addpath()

`end` change this path if you install the VOC code elsewhere `addpath ()`

9.10.1.2 copyfile()

```
copyfile (
    gtlabelfile ,
    gtclasslabelfile )
```

9.10.1.3 CREATE_SEGMENTATIONS_FROM_DETECTIONS() [1/2]

Creates segmentation [results](#) from detection [results](#) CREATE_SEGMENTATIONS_FROM_DETECTIONS (
 ID)

9.10.1.4 CREATE_SEGMENTATIONS_FROM_DETECTIONS() [2/2]

Creates segmentation [results](#) from detection [results](#) no matter what their [confidence](#) level CREATE_SEGMENTATIONS_FROM_DETECTIONS (
 ID ,
 CONFIDENCE)

9.10.1.5 create_segmentations_from_detections()

Creates segmentation [results](#) from detection [results](#) no matter what their [confidence](#) level but only detections above the specified [confidence](#) will be used [function](#) create_segmentations_from_detections (
 id ,
 confidence)

9.10.1.6 im()

[end](#) im (
 ind)

9.10.1.7 imwrite() [1/2]

imwrite (
 classim ,
 cmap ,
 classlabelfile)

9.10.1.8 imwrite() [2/2]

imwrite (
 instim ,
 cmap ,
 instlabelfile)

9.10.1.9 isempty()

```
if isempty (
    ind )
```

create_segmentations_from_detections.m 파일의 50 번째 라인에서 정의되었습니다.
 51 {end+1}=imgid;

9.10.1.10 png()

```
class png ( ) [virtual]
```

9.10.1.11 uint8()

```
zeros([H W] uint8 ( ) [virtual]
```

9.10.1.12 ~exist() [1/3]

```
if ~exist (
    resultsdir ,
    'dir' )
```

9.10.1.13 ~exist() [2/3]

```
end if ~exist (
    resultsdirinst ,
    'dir' )
```

9.10.1.14 ~exist() [3/3]

```
if ~exist (
    resultsfile ,
    'file' )
```


9.10.1.15 ~strcmp()

```
if ~strcmp (
    imgid ,
    previd )
```

9.10.2 변수 문서화

9.10.2.1 bbox

```
record objects(obj bbox = BBOXS(j,:))
```

create_segmentations_from_detections.m 파일의 49 번째 라인에서 정의되었습니다.

9.10.2.2 BBOXS

```
BBOXS =[b1 b2 b3 b4]
```

create_segmentations_from_detections.m 파일의 30 번째 라인에서 정의되었습니다.

9.10.2.3 classim

```
classim = uint8(zeros([H W]))
```

create_segmentations_from_detections.m 파일의 106 번째 라인에서 정의되었습니다.

9.10.2.4 classlabelfile

```
classlabelfile = sprintf(VOCopts.seg.clsrespath,id,VOCopts.testset,imname)
```

create_segmentations_from_detections.m 파일의 84 번째 라인에서 정의되었습니다.

9.10.2.5 clsnum

```
end detinfo clsnum
```

초기값:

```
= 1:VOCopts.nclasses  
resultsfile = sprintf(VOCopts.detrespath,id,VOCopts.classes{clsnum})
```

create_segmentations_from_detections.m 파일의 24 번째 라인에서 정의되었습니다.

9.10.2.6 cmap

```
end cmap = VOClabelcolormap(255)
```

create_segmentations_from_detections.m 파일의 73 번째 라인에서 정의되었습니다.

9.10.2.7 conf

```
detinfo conf = confs(j)
```

create_segmentations_from_detections.m 파일의 41 번째 라인에서 정의되었습니다.

9.10.2.8 detnum

```
else detnum =1
```

create_segmentations_from_detections.m 파일의 53 번째 라인에서 정의되었습니다.

9.10.2.9 drawnow

```
drawnow
```

create_segmentations_from_detections.m 파일의 36 번째 라인에서 정의되었습니다.

9.10.2.10 end

```
end[ids, confs, b1, b2, b3, b4] =textread(resultsfile,'%s %f %f %f %f %f')
```

create_segmentations_from_detections.m 파일의 29 번째 라인에서 정의되었습니다.

9.10.2.11 gtclasslabelfile

```
gtclasslabelfile = sprintf('%s/%d_gt.png',resultsdir,imnums(j))
```

create_segmentations_from_detections.m 파일의 97 번째 라인에서 정의되었습니다.

9.10.2.12 imgfile

```
imgfile = sprintf(VOCopts.imgpath,imname)
```

create_segmentations_from_detections.m 파일의 87 번째 라인에서 정의되었습니다.

9.10.2.13 imgid

```
end imgid = ids{j}
```

create_segmentations_from_detections.m 파일의 40 번째 라인에서 정의되었습니다.

9.10.2.14 imgids

```
imgids = {}
```

create_segmentations_from_detections.m 파일의 23 번째 라인에서 정의되었습니다.

9.10.2.15 imginfo

```
imginfo = imfinfo(imgfile)
```

create_segmentations_from_detections.m 파일의 88 번째 라인에서 정의되었습니다.

9.10.2.16 imname

```
end imname = imgids{j}
```

create_segmentations_from_detections.m 파일의 82 번째 라인에서 정의되었습니다.

9.10.2.17 ind

```
ind = numel(imgids)
```

create_segmentations_from_detections.m 파일의 52 번째 라인에서 정의되었습니다.

9.10.2.18 instlabelfile

```
instlabelfile = sprintf(VOCopts.seg.instrespath,id,VOCopts.testset,imname)
```

create_segmentations_from_detections.m 파일의 85 번째 라인에서 정의되었습니다.

9.10.2.19 j

```
for j
```

초기값:

```
=1:numel(ids)
    % display progress
    if toc>1
        fprintf('class %d/%d: load detections: %d/%d\n',clsnum,VOCopts.nclasses,j,numel(ids))
```

create_segmentations_from_detections.m 파일의 32 번째 라인에서 정의되었습니다.

9.10.2.20 previd

```
previd = ''
```

create_segmentations_from_detections.m 파일의 31 번째 라인에서 정의되었습니다.

9.10.2.21 resultsdir

```
end end Write out the segmentations resultsdir = sprintf(VOCopts.seg.clsresdir,id,VOCopts.testset)
```

create_segmentations_from_detections.m 파일의 62 번째 라인에서 정의되었습니다.

9.10.2.22 resultsdirinst

resultsdirinst = sprintf(VOCopts.seg.instresdir, id, VOCopts.testset)

create_segmentations_from_detections.m 파일의 63 번째 라인에서 정의되었습니다.

9.10.2.23 resultsfile

if resultsfile

create_segmentations_from_detections.m 파일의 27 번째 라인에서 정의되었습니다.

9.10.2.24 tic

classify each image tic

create_segmentations_from_detections.m 파일의 22 번째 라인에서 정의되었습니다.

9.10.2.25 VOCinit

initialize VOC options VOCinit

create_segmentations_from_detections.m 파일의 18 번째 라인에서 정의되었습니다.

9.11 data/VOCdevkit/example_classifier.m 파일 참조

함수

- `function` example_classifier change this path if you install the VOC code elsewhere `addpath` ([cd '/VOCcode'])
- train classifier test (VOCopts, cls, classifier)
- compute and display PR if i < VOCopts.nclasses fprintf('press any key to continue with next class...\n');drawnow;pause;endend% train classifierfunction classifier=train(VOCopts, cls)% load 'train' image set for class[ids, classifier.gt]=textread(sprintf(VOCopts.clsimgsetpath, cls, 'train'), '%s %d');% extract features for each imageclassifier.FD=zeros(0, length(ids));tic;for i=1↔:length(ids) % display progress if toc > fprintf('%s:train:%d/%d\n', cls, i, length(ids))
- end try try to load features load (sprintf(VOCopts.exfdpath, ids{i}), 'fd')
- save (sprintf(VOCopts.exfdpath, ids{i}), 'fd')
- end classifier FD (1:length(fd), i)
- write to results file fprintf (fid, '%s %f\n', ids{i}, c)
- end close results file fclose (fid)

변수

- initialize VOC options `VOCinit`
- train and test classifier for each class for `i`
- `classifier = train(VOCopts,cls)`
- `drawnow`
- `tic`
- catch compute and save features `I = imread(sprintf(VOCopts.imgpath,ids{i}))`
- `fd = extractfd(VOCopts,I)`
- end run classifier on test images `function gt = textread(sprintf(VOCopts.imgsetpath,VOCopts.testset),'%s %d')`
- create results file `fid = fopen(sprintf(VOCopts.clsrespath,'comp1',cls),'w')`
- end compute confidence of positive classification `c = classify(VOCopts,classifier,fd)`
- trivial feature extractor
- trivial `fd d = sum(fd.*fd)+sum(classifier.FD.*classifier.FD)-2*fd'*classifier.FD`
- `dp = min(d(classifier.gt>0))`
- `dn = min(d(classifier.gt<0))`

9.11.1 함수 문서화

9.11.1.1 addpath()

`function` example_classifier change this path if you install the VOC code elsewhere `addpath ()`

9.11.1.2 fclose()

end close results file `fclose (`
`fd)`

9.11.1.3 FD()

end classifier `FD (`
`1:length(fd) ,`
`i)`

9.11.1.4 fprintf() [1/2]

```

compute and display PR if i<VOCOpts.nclasses fprintf('press any key to continue with next class...\n'); drawnow; pause; en-
dend% train classifierfunction classifier = train(VOCOpts,cls)% load 'train' image set for class[ids,classifier.gt]=textread(sprintf(V←
OCOpts.clsimgsetpath,cls,'train'),'s %d');% extract features for each imageclassifier.FD=zeros(0,length(ids));tic;for i=1←
:length(ids) % display progress if toc> fprintf (
    '%s:train:%d/%d\n' ,
    cls ,
    i ,
    length(ids) )

```

9.11.1.5 fprintf() [2/2]

```

write to results file fprintf (
    fid ,
    '%s %f\n' ,
    ids{i} ,
    c )

```

9.11.1.6 load()

```

end try try to load features load (
    sprintf(VOCOpts.exfdpath, ids{i}) ,
    'fd' )

```

9.11.1.7 save()

```

save (
    sprintf(VOCOpts.exfdpath, ids{i}) ,
    'fd' )

```

9.11.1.8 test()

```

end run classifier on test images function test (
    VOCOpts ,
    cls ,
    classifier )

```

9.11.2 변수 문서화

9.11.2.1 c

```
c=classify(VOCopts,classifier,fd)
```

example_classifier.m 파일의 83 번째 라인에서 정의되었습니다.

9.11.2.2 classifier

```
trivial classifier=train(VOCopts,cls)
```

example_classifier.m 파일의 12 번째 라인에서 정의되었습니다.

9.11.2.3 d

```
trivial fd d=sum(fd.*fd)+sum(classifier.FD.*classifier.FD)-2*fd'*classifier.FD
```

example_classifier.m 파일의 102 번째 라인에서 정의되었습니다.

9.11.2.4 dn

```
dn=min(d(classifier.gt<0))
```

example_classifier.m 파일의 104 번째 라인에서 정의되었습니다.

9.11.2.5 dp

```
dp=min(d(classifier.gt>0))
```

example_classifier.m 파일의 103 번째 라인에서 정의되었습니다.

9.11.2.6 drawnow

```
drawnow
```

example_classifier.m 파일의 36 번째 라인에서 정의되었습니다.

9.11.2.7 extractor

trivial feature extractor

example_classifier.m 파일의 93 번째 라인에서 정의되었습니다.

9.11.2.8 fd

trivial feature `I` `fd = extractfd(VOCopts,I)`

example_classifier.m 파일의 46 번째 라인에서 정의되었습니다.

9.11.2.9 fid

create `results` file `fid = fopen(sprintf(VOCopts.clsrespath,'comp1',cls),'w')`

example_classifier.m 파일의 60 번째 라인에서 정의되었습니다.

9.11.2.10 gt

`end` run classifier on test images `function gt = textread(sprintf(VOCopts.imgsetpath,VOCopts.testset),'%s %d')`

example_classifier.m 파일의 57 번째 라인에서 정의되었습니다.

9.11.2.11 i

for `i`

초기값:

```
=1:VOCopts.nclasses
    cls=VOCopts.classes{i}
```

example_classifier.m 파일의 10 번째 라인에서 정의되었습니다.

9.11.2.12 I

catch compute and `save` features `I = imread(sprintf(VOCopts.imgpath,ids{i}))`

example_classifier.m 파일의 45 번째 라인에서 정의되었습니다.

9.11.2.13 tic

tic

example_classifier.m 파일의 37 번째 라인에서 정의되었습니다.

9.11.2.14 VOCinit

initialize VOC options VOCinit

example_classifier.m 파일의 7 번째 라인에서 정의되었습니다.

9.12 data/VOCdevkit/example_detector.m 파일 참조

함수

- `function` example_detector change this path `if` you install the VOC code elsewhere `addpath` ([cd '/VOCcode'])
- train `detector` test (VOCopts, `cls`, `detector`)
- compute and display PR `if` `i < VOCopts.nclasses` `fprintf`('press any key to continue with next class...\n');`drawnow`;`pause`;endend% train detectorfunction `detector`=train(VOCopts, `cls`)% load 'train' image setids=textread(sprintf(VOCopts.imgsetpath, 'train','%s'),'%s');% extract features and bounding `boxesdetector.FD=[]`;`detector.bbox={}`;`detector.gt=[]`;`tic`;for `i=1:length(ids)` % display progress `if` `toc >` `fprintf` ('%s:train:%d/%d\n', `cls`, `i`, `length(ids)`)

변수

- initialize VOC options `VOCinit`
- train and test `detector` for each `class` for `i`
- `detector` =train(VOCopts,`cls`)
- `drawnow`
- `tic`
- `end` read annotation `rec` =PASreadrecord(sprintf(VOCopts.annopath,`ids`{`i`}))

9.12.1 함수 문서화

9.12.1.1 addpath()

`function` example_detector change this path `if` you install the VOC code elsewhere `addpath` ()

9.12.1.2 fprintf()

```
compute and display PR if i<VOCopts.nclasses fprintf('press any key to continue with next class...\n'); drawnow; pause;
endend% train detectorfunction detector = train(VOCopts,cls)% load 'train' image setids=textread(sprintf(VOCopts.%s\
imgsetpath,'train'),'%s');% extract features and bounding boxesdetector.FD=[];detector.bbox={};detector.gt=[];tic;for
i=1:length(ids) % display progress if toc> fprintf (
    '%s:train:%d/%d\n' ,
    cls ,
    i ,
    length(ids) )
```

9.12.1.3 test()

```
train detector test (
    VOCopts ,
    cls ,
    detector )
```

9.12.2 변수 문서화

9.12.2.1 detector

```
test detector =train(VOCopts,cls)
```

example_detector.m 파일의 12 번째 라인에서 정의되었습니다.

9.12.2.2 drawnow

```
drawnow
```

example_detector.m 파일의 38 번째 라인에서 정의되었습니다.

9.12.2.3 i

train and test detector for each class for i

초기값:

```
=1:VOCopts.nclasses
cls=VOCopts.classes{i}
```

example_detector.m 파일의 10 번째 라인에서 정의되었습니다.

9.12.2.4 rec

```
else rec =PASreadrecord(sprintf(VOCopts.annopath,ids{i}))
```

example_detector.m 파일의 43 번째 라인에서 정의되었습니다.

9.12.2.5 tic

```
tic
```

example_detector.m 파일의 39 번째 라인에서 정의되었습니다.

9.12.2.6 VOCinit

```
initialize VOC options VOCinit
```

example_detector.m 파일의 7 번째 라인에서 정의되었습니다.

9.13 data/VOCdevkit/example_layout.m 파일 참조

함수

- `function` example_layout change this path `if` you install the VOC code elsewhere `addpath` ([cd '/VOCcode'])
- train `detector` test (VOCopts, `cls`, `detector`)

변수

- initialize VOC options `VOCinit`
- train and test `detector` `cls` ='person'
- `detector` =train(VOCopts,`cls`)

9.13.1 함수 문서화

9.13.1.1 addpath()

```
function example_layout change this path if you install the VOC code elsewhere addpath ( )
```

9.13.1.2 test()

```
train detector test (
    VCOpts ,
    cls ,
    detector )
```

9.13.2 변수 문서화

9.13.2.1 cls

train and test detector cls ='person'

example_layout.m 파일의 11 번째 라인에서 정의되었습니다.

9.13.2.2 detector

test detector[recall, prec, ap] =train(VCOpts,cls)

example_layout.m 파일의 12 번째 라인에서 정의되었습니다.

9.13.2.3 VOCinit

initialize VOC options VOCinit

example_layout.m 파일의 7 번째 라인에서 정의되었습니다.

9.14 data/VOCdevkit/example_segmenter.m 파일 참조

함수

- example_segmenter Segmentation algorithm based on detection results This segmenter requires that some detection results are present in Results e g by running example_detector Segmentations are generated from detection bounding boxes function example_segmenter VOCinit create_segmentations_from_detections ('comp3', 1) VOCevalseg(VCOpts

변수

- example_segmenter Segmentation algorithm based on detection results This segmenter requires that some detection results are present in Results e g by running example_detector Segmentations are generated from detection bounding boxes function example_segmenter VOCinit comp3

9.14.1 함수 문서화

9.14.1.1 create_segmentations_from_detections()

example__segmenter Segmentation algorithm based on detection results This segmenter requires that some detection results are present in Results e g by running example__detector Segmentations are generated from detection bounding boxes function example__segmenter VOCinit create_segmentations_from_detections (

```
'comp3',
1 )
```

9.14.2 변수 문서화

9.14.2.1 comp3

example__segmenter Segmentation algorithm based on detection results This segmenter requires that some detection results are present in Results e g by running example__detector Segmentations are generated from detection bounding boxes function example__segmenter VOCinit comp3

example__segmenter.m 파일의 11 번째 라인에서 정의되었습니다.

9.15 data/VOCdevkit/viewanno.m 파일 참조

함수

- **function** viewanno(imgset) if nargin< 1 error(['usage end change this path if you install the VOC code elsewhere addpath ([cd '/VOCcode'])
- **end** display annotation if rec segmented subplot (131)
- **end** imshow (I)
- if rec objects (j).difficult ls
- not bb ([2 2 4 4 2])
- not text (bb(1), bb(2), rec.objects(j).class, 'color', 'k', 'backgroundcolor', ls(1),... 'verticalalignment', 'top', 'horizontalalignment', 'left', 'fontsize', 8)
- plot (bb([1 3 3 1 1]), bb([2 2 4 4 2]), ls, 'linewidth', 2)
- text (bb(1), bb(2), rec.objects(j).part(k).class, 'color', 'k', 'backgroundcolor', ls(1),... 'verticalalignment', 'top', 'horizontalalignment', 'left', 'fontsize', 8)
- title (sprintf('image:%d/%d:"%s"(dotted=truncated, yellow=difficult)',... i, length(ids), ids{i}))
- if rec segmented subplot (132)
- imshow (Sclass, CMclass)
- title ('segmentation by class')
- subplot (133)
- imshow (Sobj, CMobj)
- title ('segmentation by object')
- **end** fprintf ('press any key to continue with next image\n')

변수

- initialize VOC options `VOCinit`
- `load image set [ids, gt] =textread(sprintf(VOCopts.imgsetpath,['../' imgset]),'%s %d')`
- for `i`
- read `image I` =`imread(sprintf(VOCopts.imgpath,ids{i}))`
- if `rec segmented` read `segmentations` [`Sclass, CMclass`] =`imread(sprintf(VOCopts.seg.↵clsimgpath,ids{i}))`
- else `clf`
- hold on
- for `j`
- `difficult __pad0__`
- not `difficult`
- `truncated __pad1__`
- not `truncated`
- not `ls`
- not `linewidth`
- for `k`
- `end end` hold off
- axis `image`
- `pause`

9.15.1 함수 문서화

9.15.1.1 addpath()

```
function viewanno (imgset) if nargin<1 error(['usage end change this path if you install the VOC code elsewhere addpath
( )
```

9.15.1.2 bb()

```
not bb ( )
```

9.15.1.3 fprintf()

```
end fprintf (
    'press any key to continue with next image\n' )
```

9.15.1.4 imshow() [1/3]

```
end imshow (
    I )
```

9.15.1.5 imshow() [2/3]

```
imshow (
    Sclass ,
    CMclass )
```

9.15.1.6 imshow() [3/3]

```
imshow (
    Sobj ,
    CMobj )
```

9.15.1.7 objects()

```
if rec objects (
    j )
```

9.15.1.8 plot()

```
plot (
    bb([1 3 3 1 1]) ,
    bb([2 2 4 4 2]) ,
    ls ,
    'linewidth' ,
    2 )
```

9.15.1.9 subplot() [1/3]

```
end display annotation if rec segmented subplot (
    131 )
```


9.15.1.10 subplot() [2/3]

```
if rec.segmented subplot (
    132 )
```

9.15.1.11 subplot() [3/3]

```
subplot (
    133 )
```

9.15.1.12 text() [1/2]

```
not text (
    bb(1) ,
    bb(2) ,
    rec.objects(j).class,
    'color' ,
    'k' ,
    'backgroundcolor' ,
    ls(1) ,
    ... 'verticalalignment' ,
    'top' ,
    'horizontalalignment' ,
    'left' ,
    'fontsize' ,
    8 )
```

9.15.1.13 text() [2/2]

```
text (
    bb(1) ,
    bb(2) ,
    rec.objects(j).part(k).class,
    'color' ,
    'k' ,
    'backgroundcolor' ,
    ls(1) ,
    ... 'verticalalignment' ,
    'top' ,
    'horizontalalignment' ,
    'left' ,
    'fontsize' ,
    8 )
```

9.15.1.14 title() [1/3]

```
title (
    'segmentation by class' )
```

9.15.1.15 title() [2/3]

```
title (
    'segmentation by object' )
```

9.15.1.16 title() [3/3]

```
title (
    sprintf( 'image:%d/%d:"%s"(dotted=truncated, yellow=difficult)',... i, length(ids), ids{i} ) )
```

9.15.2 변수 문서화**9.15.2.1 __pad0__**

```
difficult __pad0__
```

viewanno.m 파일의 48 번째 라인에서 정의되었습니다.

9.15.2.2 __pad1__

```
truncated __pad1__
```

viewanno.m 파일의 53 번째 라인에서 정의되었습니다.

9.15.2.3 clf

```
else clf
```

viewanno.m 파일의 38 번째 라인에서 정의되었습니다.

9.15.2.4 difficult

not difficult

viewanno.m 파일의 51 번째 라인에서 정의되었습니다.

9.15.2.5 i

for i

초기값:

```
=1:length(ids)

% read annotation
rec=PASreadrecord(sprintf(VOCopts.annopath,ids{i}))
```

viewanno.m 파일의 17 번째 라인에서 정의되었습니다.

9.15.2.6 I

```
read image I = imread(sprintf(VOCopts.imgpath,ids{i}))
```

viewanno.m 파일의 23 번째 라인에서 정의되었습니다.

9.15.2.7 image

axis image

viewanno.m 파일의 67 번째 라인에서 정의되었습니다.

9.15.2.8 j

for j

초기값:

```
=1:length(rec.objects)
bb=rec.objects(j).bbox
```

viewanno.m 파일의 43 번째 라인에서 정의되었습니다.

9.15.2.9 k

for k

초기값:

```
=1:length(rec.objects(j).part)
bb=rec.objects(j).part(k).bbox
```

viewanno.m 파일의 59 번째 라인에서 정의되었습니다.

9.15.2.10 linewidth

not linewidth

viewanno.m 파일의 55 번째 라인에서 정의되었습니다.

9.15.2.11 ls

not ls

viewanno.m 파일의 55 번째 라인에서 정의되었습니다.

9.15.2.12 off

axis off

viewanno.m 파일의 66 번째 라인에서 정의되었습니다.

9.15.2.13 on

hold on

viewanno.m 파일의 42 번째 라인에서 정의되었습니다.

9.15.2.14 pause

pause

viewanno.m 파일의 87 번째 라인에서 정의되었습니다.

9.15.2.15 segmentations

```
if rec segmented read segmentations[Sclass, CMclass] =imread(sprintf(VOCopts.seg.clsimgpath,ids{i}))
```

viewanno.m 파일의 29 번째 라인에서 정의되었습니다.

9.15.2.16 set

```
load image set[ids, gt] =textread(sprintf(VOCopts.imgsetpath,['../' imgset]),'%s %d')
```

viewanno.m 파일의 15 번째 라인에서 정의되었습니다.

9.15.2.17 truncated

```
not truncated
```

viewanno.m 파일의 55 번째 라인에서 정의되었습니다.

9.15.2.18 VOCinit

```
initialize VOC options VOCinit
```

viewanno.m 파일의 12 번째 라인에서 정의되었습니다.

9.16 data/VOCdevkit/viewdet.m 파일 참조

함수

- `function viewdet(cls, onlytp) if nargin< 1 error(['usage end if nargin< 2 onlytp=false;end% change this path if you install the VOC code elsewhereaddpath([cd '/VOCcode']);% initialize VOC optionsVOCinit;% load test set[gtids, t]=textread(sprintf(VOCopts.imgsetpath, VOCopts.testset),'%s %d');% load ground truth objectstic;npos=0;for i=1:length(gtids) % display progress if toc > fprintf ('%s:viewdet:load:%d/%d\n', cls, i, length(gtids))`
- `gt(i).BB`
- `if isempty(i) error('unrecognized image "%s"')`
- `elseif length(i)>1 error('multiple image "%s"')`
- `max(bb(2), bbgt(2))`
- `min(bb(3), bbgt(3))`
- `min(bb(4), bbgt(4))`
- `plot(bbgt([1 3 3 1 1]), bbgt([2 2 4 4 2]), 'y-', 'linewidth', 2)`
- `plot(bb([1 3 3 1 1]), bb([2 2 4 4 2]), 'g:', 'linewidth', 2)`
- `else plot(bb([1 3 3 1 1]), bb([2 2 4 4 2]), 'r-', 'linewidth', 2)`

변수

- `drawnow`
- `tic`
- `end` read annotation `rec =PASreadrecord(sprintf(VOCopts.annopath,gtids{i}))`
- extract `objects` of `class clsinds =strmatch(cls,{rec.objects(:).class},'exact')`
- `npos =npos+sum(~gt(i).diff)`
- `end load results [ids, confidence, b1, b2, b3, b4]=textread(sprintf(VOCopts.detrespath,'comp3',cls),'%s %f %f %f %f %f')`
- `BB =[b1 b2 b3 b4]'`
- sort detections by decreasing `confidence [sc, si] =sort(-confidence)`
- `ids =ids(si)`
- `view` detections `nd =length(confidence)`
- for `d`
- `end` find ground truth `image i =strmatch(ids{d},gtids,'exact')`
- `end` assign detection to ground truth `object if any bb =BB(:,d)`
- `ovmax =-inf`
- for `j`
- `bi =[max(bb(1),bbgt(1))`
- `iw =bi(3)-bi(1)+1`
- `ih =bi(4)-bi(2)+1`
- `if iw &ih` compute overlap as area of intersection area of union `ua`
- `ov =iw*ih/ua`
- `jmax =j`
- `end end end` skip false positives `if onlytp & ovmax< VOCopts.minoverlap continue end % read image I=imread(spr`
- `end` hold off
- axis `image`

9.16.1 함수 문서화

9.16.1.1 fprintf()

```
function viewdet (cls,onlytp) if nargin<1 error(['usage end if nargin<2 onlytp=false;end% change this path if
you install the VOC code elsewhereaddpath([cd '/VOCcode']);% initialize VOC optionsVOCinit;% load test
set[gtids,t]=textread(sprintf(VOCopts.imgsetpath,VOCopts.testset),'%s %d');% load ground truth objectstic;npos=0;for
i=1:length(gtids) % display progress if toc> fprintf (
    '%s:viewdet:load:%d/%d\n' ,
    cls ,
    i ,
    length(gtids) )
```

9.16.1.2 gt()

```
gt (
    i )
```

9.16.1.3 isempty()

```
if isempty (
    i )
```

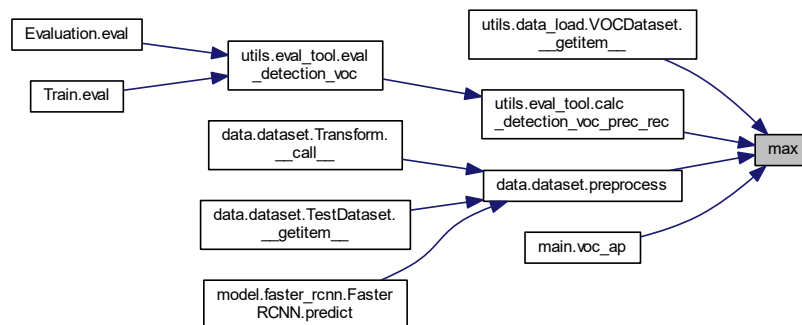
9.16.1.4 length()

```
elseif length (
    i )
```

9.16.1.5 max()

```
max (
    bb(2) ,
    bbgt(2) )
```

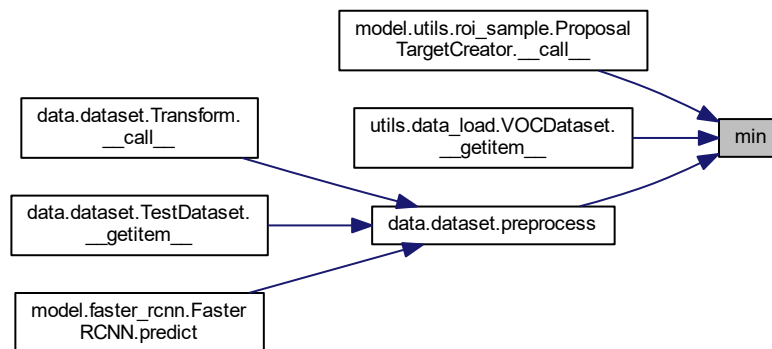
이 함수를 호출하는 함수들에 대한 그래프입니다.:



9.16.1.6 min() [1/2]

```
min (
    bb(3) ,
    bbgt(3) )
```

이 함수를 호출하는 함수들에 대한 그래프입니다.:



9.16.1.7 min() [2/2]

```
min (
    bb(4) ,
    bbgt(4) )
```

9.16.1.8 plot() [1/3]

```
plot (
    bb([1 3 3 1 1]) ,
    bb([2 2 4 4 2]) ,
    'g:' ,
    'linewidth' ,
    2 )
```

9.16.1.9 plot() [2/3]

```
else plot (
    bb([1 3 3 1 1]) ,
    bb([2 2 4 4 2]) ,
    'r-' ,
    'linewidth' ,
    2 )
```


9.16.1.10 plot() [3/3]

```
plot (
    bbgt([1 3 3 1 1]) ,
    bbgt([2 2 4 4 2]) ,
    'y-' ,
    'linewidth' ,
    2 )
```

9.16.2 변수 문서화**9.16.2.1 BB**

```
BB =[b1 b2 b3 b4]'
```

viewdet.m 파일의 45 번째 라인에서 정의되었습니다.

9.16.2.2 bb

```
end assign detection to ground truth object if any bb =BB(:,d)
```

viewdet.m 파일의 72 번째 라인에서 정의되었습니다.

9.16.2.3 bi

```
bi =[max(bb(1),bbgt(1))
```

viewdet.m 파일의 76 번째 라인에서 정의되었습니다.

9.16.2.4 clsinds

```
extract objects of class clsinds =strmatch(cls,{rec.objects(:).class},'exact')
```

viewdet.m 파일의 36 번째 라인에서 정의되었습니다.

9.16.2.5 confidence

sort detections by decreasing confidence[sc, si] =sort(-confidence)

viewdet.m 파일의 48 번째 라인에서 정의되었습니다.

9.16.2.6 d

for d

초기값:

```
=1:nd
    % display progress
    if onlytp&toc>1
        fprintf('%s: viewdet: find true pos: %d/%d\n',cls,i,length(gtids))
```

viewdet.m 파일의 55 번째 라인에서 정의되었습니다.

9.16.2.7 drawnow

drawnow

viewdet.m 파일의 28 번째 라인에서 정의되었습니다.

9.16.2.8 i

```
end find ground truth image i =strmatch(ids{d},gtids,'exact')
```

viewdet.m 파일의 64 번째 라인에서 정의되었습니다.

9.16.2.9 ids

```
elseif ids =ids(si)
```

viewdet.m 파일의 49 번째 라인에서 정의되었습니다.

9.16.2.10 ih

```
ih =bi(4)-bi(2)+1
```

viewdet.m 파일의 78 번째 라인에서 정의되었습니다.

9.16.2.11 image

```
axis image
```

viewdet.m 파일의 111 번째 라인에서 정의되었습니다.

9.16.2.12 iw

```
iw =bi(3)-bi(1)+1
```

viewdet.m 파일의 77 번째 라인에서 정의되었습니다.

9.16.2.13 j

```
for j
```

초기값:

```
=1:size(gt(i).BB,2)  
bbgt=gt(i).BB(:,j)
```

viewdet.m 파일의 74 번째 라인에서 정의되었습니다.

9.16.2.14 jmax

```
jmax =j
```

viewdet.m 파일의 87 번째 라인에서 정의되었습니다.

9.16.2.15 nd

```
view detections nd=length(confidence)
```

viewdet.m 파일의 53 번째 라인에서 정의되었습니다.

9.16.2.16 npos

```
npos =npos+sum(~gt(i).diff)
```

viewdet.m 파일의 40 번째 라인에서 정의되었습니다.

9.16.2.17 off

```
axis off
```

viewdet.m 파일의 110 번째 라인에서 정의되었습니다.

9.16.2.18 ov

```
ov =iw*ih/ua
```

viewdet.m 파일의 84 번째 라인에서 정의되었습니다.

9.16.2.19 ovmax

```
end end end assign detection as true false positive if ovmax ==-inf
```

viewdet.m 파일의 73 번째 라인에서 정의되었습니다.

```
9.16.2.20 ovmax< VOCopts.minoverlap continue end % read image I=imread(sprintf(VOCopts.imgpath, gtids{i}));% draw detection bounding box and ground truth bounding box(if any) imagesc(I);hold on;if ovmax >
```

```
end end end skip false positives if onlytp& ovmax< VOCopts.minoverlap continue end % read image I=imread(sprintf(VOCopts.imgpath, gtids{i})); % draw detection bounding box and ground truth bounding box(if any) imagesc(I); hold on; if ovmax >
```

초기값:

```
=VOCopts.minoverlap  
bbgt=gt(i).BB(:,jmax)
```

viewdet.m 파일의 103 번째 라인에서 정의되었습니다.

9.16.2.21 rec

```
end read annotation rec =PASreadrecord(sprintf(VOCopts.annopath,gtids{i}))
```

viewdet.m 파일의 33 번째 라인에서 정의되었습니다.

9.16.2.22 results

```
end load results[ids, confidence, b1, b2, b3, b4] =textread(sprintf(VOCopts.detrespath,'comp3',cls), '%s %f %f %f %f %f')
```

viewdet.m 파일의 44 번째 라인에서 정의되었습니다.

9.16.2.23 tic

```
tic
```

viewdet.m 파일의 29 번째 라인에서 정의되었습니다.

9.16.2.24 ua

```
if iw & ih compute overlap as area of intersection area of union ua
```

초기값:

```
= (bb(3)-bb(1)+1) * (bb(4)-bb(2)+1) + ...  
    (bbgt(3)-bbgt(1)+1) * (bbgt(4)-bbgt(2)+1) - ...  
    iw*ih
```

viewdet.m 파일의 81 번째 라인에서 정의되었습니다.

9.17 data/VOCdevkit/VOCcode/PASemptyobject.m 파일 참조

변수

- function object
- object orglabel =''
- object bbox =[]
- object polygon =[]
- object mask =''
- object class =''
- object view =''
- object truncated =false
- object difficult =false

9.17.1 변수 문서화

9.17.1.1 bbox

```
object bbox = []
```

PASemptyobject.m 파일의 4 번째 라인에서 정의되었습니다.

9.17.1.2 class

```
object class = ''
```

PASemptyobject.m 파일의 7 번째 라인에서 정의되었습니다.

9.17.1.3 difficult

```
object difficult = false
```

PASemptyobject.m 파일의 10 번째 라인에서 정의되었습니다.

9.17.1.4 mask

```
else p mask = ''
```

PASemptyobject.m 파일의 6 번째 라인에서 정의되었습니다.

9.17.1.5 object

```
function object
```

초기값:

```
=PASemptyobject  
    object.label = ''
```

PASemptyobject.m 파일의 1 번째 라인에서 정의되었습니다.

9.17.1.6 orglabel

```
object orglabel =''
```

PASemptyobject.m 파일의 3 번째 라인에서 정의되었습니다.

9.17.1.7 polygon

```
else p polygon =[]
```

PASemptyobject.m 파일의 5 번째 라인에서 정의되었습니다.

9.17.1.8 truncated

```
object truncated =false
```

PASemptyobject.m 파일의 9 번째 라인에서 정의되었습니다.

9.17.1.9 view

```
end else p view =''
```

PASemptyobject.m 파일의 8 번째 라인에서 정의되었습니다.

9.18 data/VOCdevkit/VOCcode/PASemptyrecord.m 파일 참조

변수

- `function record`
- `record imgsize =[]`
- `record database =''`
- `record objects =PASemptyobject`

9.18.1 변수 문서화

9.18.1.1 database

```
record database =''
```

PASemptyrecord.m 파일의 4 번째 라인에서 정의되었습니다.

9.18.1.2 imgsize

```
record imgsize =[]
```

PASemptyrecord.m 파일의 3 번째 라인에서 정의되었습니다.

9.18.1.3 objects

```
record objects =PASemptyobject
```

PASemptyrecord.m 파일의 5 번째 라인에서 정의되었습니다.

9.18.1.4 record

```
function record
```

초기값:

```
=PASemptyrecord
record.imgname=""
```

PASemptyrecord.m 파일의 1 번째 라인에서 정의되었습니다.

9.19 data/VOCdevkit/VOCcode/PASerrmsg.m 파일 참조

함수

- `function PASerrmsg(PASerr, SYSerr) fprintf('Pascal Error Message fprintf ('System Error Message:%s\n', SYSerr)`
- `if (~isempty(k))`
- `if (lower(k)=='k')`
- `fprintf ('\n')`

변수

- `k=input('Enter K for keyboard, any other key to continue or ^C to quit ...','s')`
- `keyboard`
- `end`

9.19.1 함수 문서화

9.19.1.1 fprintf() [1/2]

```
fprintf (
    '\n' )
```

9.19.1.2 fprintf() [2/2]

```
function PASerrmsg (PASerr,SYserr) fprintf('Pascal Error Message fprintf (
    'System Error Message:%s\n' ,
    SYserr )
```

9.19.1.3 if() [1/2]

```
if (
    lower(k) == 'k' )
```

9.19.1.4 if() [2/2]

```
if (
    ~ isempty(k) )
```

9.19.2 변수 문서화

9.19.2.1 end

```
end
```

PASerrmsg.m 파일의 5 번째 라인에서 정의되었습니다.

9.19.2.2 k

```
k=input('Enter K for keyboard, any other key to continue or ^C to quit ...','s')
```

PASerrmsg.m 파일의 4 번째 라인에서 정의되었습니다.

9.19.2.3 keyboard

```
keyboard
```

PASerrmsg.m 파일의 5 번째 라인에서 정의되었습니다.

9.20 data/VOCdevkit/VOCcode/PASreadrecord.m 파일 참조

함수

- [end if strcmp](#) (path([end-3:end](#)),'.txt') [rec](#)

변수

- [function rec](#)

9.20.1 함수 문서화

9.20.1.1 strcmp()

```
end if strcmp (
    path(end-3:end) ,
    '.txt' )
```

9.20.2 변수 문서화

9.20.2.1 rec

```
else rec
```

초기값:

```
= PASreadrecord(path)
if length(path)<4
    error('unable to determine format: %s',path)
```

PASreadrecord.m 파일의 1 번째 라인에서 정의되었습니다.

9.21 data/VOCdevkit/VOCcode/PASreadrectxt.m 파일 참조

함수

- `if (fd== -1)`
- `PASerrmsg (PASmsg, syserrmsg)`
- `while (notEOF)`
- `if (notEOF)`
- `record objects (obj).label`

변수

- `function record`
- `PASmsg =sprintf('Could not open %s for reading',filename)`
- `end`
- `matchstrs =initstrings`
- `notEOF =1`
- `line =fgetl(fd)`
- `matchnum =match(line,matchstrs)`
- `switch case`
- `switch record imgname =char(imgname)`
- `record imgsize =[x y c]`
- `record database =char(database)`
- `tmp =findstr(line,' : ')`

9.21.1 함수 문서화

9.21.1.1 if() [1/2]

```
if (
    fd == -1 )
```

9.21.1.2 if() [2/2]

```
if (
    notEOF )
```

9.21.1.3 objects()

```
record objects (
    obj )
```

9.21.1.4 PASerrmsg()

```
PASerrmsg (
    PASmsg ,
    syserrmsg )
```

9.21.1.5 while()

```
while (
    notEOF )
```

9.21.2 변수 문서화

9.21.2.1 case

```
case
```

PASreadrectxt.m 파일의 16 번째 라인에서 정의되었습니다.

9.21.2.2 database

```
record database =char(database)
```

PASreadrectxt.m 파일의 22 번째 라인에서 정의되었습니다.

9.21.2.3 end

```
end
```

PASreadrectxt.m 파일의 6 번째 라인에서 정의되었습니다.

9.21.2.4 imgname

```
switch record imgname =char(imgname)
```

PASreadrectxt.m 파일의 18 번째 라인에서 정의되었습니다.

9.21.2.5 imgsize

```
record imgsize =[x y c]
```

PASreadrectxt.m 파일의 20 번째 라인에서 정의되었습니다.

9.21.2.6 line

```
line =fgetl(fd)
```

PASreadrectxt.m 파일의 12 번째 라인에서 정의되었습니다.

9.21.2.7 matchnum

```
switch matchnum =match(line,matchstrs)
```

PASreadrectxt.m 파일의 15 번째 라인에서 정의되었습니다.

9.21.2.8 matchstrs

```
matchstrs =initstrings
```

PASreadrectxt.m 파일의 8 번째 라인에서 정의되었습니다.

9.21.2.9 notEOF

```
notEOF =1
```

PASreadrectxt.m 파일의 10 번째 라인에서 정의되었습니다.

9.21.2.10 PASmsg

```
PASmsg =sprintf('Could not open %s for reading',filename)
```

PASreadrectxt.m 파일의 4 번째 라인에서 정의되었습니다.

9.21.2.11 record

record

초기값:

```
=PASreadrectxt(filename)
[fd,syserrmsg]=fopen(filename,'rt')
```

PASreadrectxt.m 파일의 1 번째 라인에서 정의되었습니다.

9.21.2.12 tmp

tmp=findstr(line,' : ')

PASreadrectxt.m 파일의 26 번째 라인에서 정의되었습니다.

9.22 data/VOCdevkit/VOCcode/VOCevalcls.m 파일 참조

함수

- if isempty(j) error('unrecognized image "%s"')
- elseif length(j)>1 error('multiple image "%s"')
- else out(j)
- if isempty(p) p=0
- end if draw plot precision recall plot(rec, prec, '-')
- xlabel recall ylabel precision title(sprintf('class:%s, subset:%s, AP=%.3f', cls, VOCopts.testset, ap))

변수

- function [rec, prec, ap]
- load results [ids, confidence] =textread(sprintf(VOCopts.clsrespath,id,cls), '%s %f')
- map results to ground truth images out =ones(size(gt))*-inf
- tic
- for i
- drawnow
- end find ground truth image j =strmatch(ids{i},gtids,'exact')
- if ids {i}
- end end compute precision recall [so, si] =sort(-out)
- tp =gt(si)>0
- fp =gt(si)<0
- rec =tp/sum(gt>0)
- prec =tp./(fp+tp)
- compute average precision ap =0
- for t
- grid

9.22.1 함수 문서화

9.22.1.1 isempty() [1/2]

```
if isempty (
    j )
```

9.22.1.2 isempty() [2/2]

```
if isempty (
    p ) [pure virtual]
```

9.22.1.3 length()

```
elseif length (
    j )
```

9.22.1.4 out()

```
else out (
    j )
```

9.22.1.5 plot()

```
end if draw plot precision recall plot (
    rec ,
    prec ,
    '-' )
```

9.22.1.6 title()

```
xlabel recall ylabel precision title (
    sprintf( 'class:%s, subset:%s, AP=%0.3f', cls, VOCopts.testset, ap) )
```

9.22.2 변수 문서화

9.22.2.1 ap

```
end ap =0
```

VOCevalcls.m 파일의 44 번째 라인에서 정의되었습니다.

9.22.2.2 drawnow

```
drawnow
```

VOCevalcls.m 파일의 16 번째 라인에서 정의되었습니다.

9.22.2.3 fp

```
no overlapping object end end compute precision recall fp =gt(si)<0
```

VOCevalcls.m 파일의 35 번째 라인에서 정의되었습니다.

9.22.2.4 function

```
function[rec, prec, ap]
```

초기값:

```
= VOCevalcls(VOCopts,id,cls,draw)
% load test set
[gtids,gt]=textread(sprintf(VOCopts.clsimgsetpath,cls,VOCopts.testset),'%s %d')
```

VOCevalcls.m 파일의 1 번째 라인에서 정의되었습니다.

9.22.2.5 grid

```
grid
```

VOCevalcls.m 파일의 56 번째 라인에서 정의되었습니다.

9.22.2.6 i

for i

초기값:

```
=1:length(ids)
% display progress
if toc>1
    fprintf('%s: pr: %d/%d\n',cls,i,length(ids))
```

VOCevalcls.m 파일의 12 번째 라인에서 정의되었습니다.

9.22.2.7 ids

elseif ids {i})

VOCevalcls.m 파일의 23 번째 라인에서 정의되었습니다.

9.22.2.8 j

```
end find ground truth image j =strmatch(ids{i},gtids,'exact')
```

VOCevalcls.m 파일의 21 번째 라인에서 정의되었습니다.

9.22.2.9 out

```
map results to ground truth images out =ones(size(gt))*-inf
```

VOCevalcls.m 파일의 10 번째 라인에서 정의되었습니다.

9.22.2.10 prec

```
prec =tp./ (fp+tp)
```

VOCevalcls.m 파일의 40 번째 라인에서 정의되었습니다.

9.22.2.11 rec

```
rec = tp/sum(gt>0)
```

VOCevalcls.m 파일의 39 번째 라인에서 정의되었습니다.

9.22.2.12 recall

```
end end compute precision recall[so, si] =sort(-out)
```

VOCevalcls.m 파일의 33 번째 라인에서 정의되었습니다.

9.22.2.13 results

```
load results[ids, confidence] =textread(sprintf(VOCopts.clsrespath,id,cls), '%s %f')
```

VOCevalcls.m 파일의 7 번째 라인에서 정의되었습니다.

9.22.2.14 t

```
for t
```

초기값:

```
=0:0.1:1  
p=max(prec(rec>=t))
```

VOCevalcls.m 파일의 45 번째 라인에서 정의되었습니다.

9.22.2.15 tic

```
tic
```

VOCevalcls.m 파일의 11 번째 라인에서 정의되었습니다.

9.22.2.16 tp

```
tp =gt(si)>0
```

VOCevalcls.m 파일의 34 번째 라인에서 정의되었습니다.

9.23 data/VOCdevkit/VOCcode/VOCevaldet.m 파일 참조

함수

- `gt (length(gtids))`
- `gt (i).BB`
- `if isempty (i) error('unrecognized image "%s"')`
- `elseif length (i)>1 error('multiple image "%s"')`
- `max (bb(2), bbgt(2))`
- `min (bb(3), bbgt(3))`
- `min (bb(4), bbgt(4))]`
- `else fp (d)`
- `false positive (multiple detection) end end else fp(d)`
- `if isempty (p) p=0`
- `end if draw plot precision recall plot (rec, prec, '-')`
- `xlabel recall ylabel precision title (sprintf('class:%s, subset:%s, AP=%.3f', cls, VOCopts.testset, ap))`

변수

- `function [rec, prec, ap]`
- `load ground truth objects tic`
- `npos =0`
- `for i`
- `drawnow`
- `end read annotation rec =PASreadrecord(sprintf(VOCopts.annopath,gtids{i}))`
- `extract objects of class clsinds =strmatch(cls,{rec.objects(:).class},'exact')`
- `end load results [ids, confidence, b1, b2, b3, b4] =textread(sprintf(VOCopts.detrespath,id,cls),'%s %f %f %f %f %f')`
- `BB =[b1 b2 b3 b4]'`
- `sort detections by decreasing confidence [sc, si] =sort(-confidence)`
- `ids =ids(si)`
- `assign detections to ground truth objects nd =length(confidence)`
- `tp =zeros(nd,1)`
- `fp =zeros(nd,1)`
- `for d`
- `end assign detection to ground truth object if any bb =BB(:,d)`
- `ovmax =-inf`
- `for j`
- `bi =[max(bb(1),bbgt(1))`
- `iw =bi(3)-bi(1)+1`
- `ih =bi(4)-bi(2)+1`
- `if iw &ih compute overlap as area of intersection area of union ua`
- `ov =iw*ih/ua`
- `jmax =j`
- `prec =tp./(fp+tp)`
- `compute average precision ap =0`
- `for t`
- `grid`

9.23.1 함수 문서화

9.23.1.1 fp()

```
else fp (
    d )
```

9.23.1.2 gt() [1/2]

```
gt (
    i )
```

9.23.1.3 gt() [2/2]

```
gt (
    length(gtids) )
```

9.23.1.4 isempty() [1/2]

```
if isempty (
    i )
```

9.23.1.5 isempty() [2/2]

```
if isempty (
    p ) [pure virtual]
```

9.23.1.6 length()

```
elseif length (
    i )
```

9.23.1.7 max()

```
max (
    bb(2) ,
    bbgt(2) )
```

9.23.1.8 min() [1/2]

```
min (
    bb(3) ,
    bbgt(3) )
```

9.23.1.9 min() [2/2]

```
min (
    bb(4) ,
    bbgt(4) )
```

9.23.1.10 plot()

```
end if draw plot precision recall plot (
    rec ,
    prec ,
    '-' )
```

9.23.1.11 positive()

```
false positive (
    multiple detection )
```

9.23.1.12 title()

```
xlabel recall ylabel precision title (
    sprintf( 'class:%s, subset:%s, AP=%0.3f', cls, VOCopts.testset, ap) )
```

9.23.2 변수 문서화

9.23.2.1 ap

```
end ap =0
```

VOCevaldet.m 파일의 102 번째 라인에서 정의되었습니다.

9.23.2.2 BB

```
BB =[b1 b2 b3 b4]'
```

VOCevaldet.m 파일의 31 번째 라인에서 정의되었습니다.

9.23.2.3 bb

```
end assign detection to ground truth object if any bb =BB(:,d)
```

VOCevaldet.m 파일의 60 번째 라인에서 정의되었습니다.

9.23.2.4 bi

```
bi =[max(bb(1),bbgt(1))
```

VOCevaldet.m 파일의 64 번째 라인에서 정의되었습니다.

9.23.2.5 clsinds

```
extract objects of class clsinds =strmatch(cls,{rec.objects(:).class},'exact')
```

VOCevaldet.m 파일의 22 번째 라인에서 정의되었습니다.

9.23.2.6 confidence

sort detections by decreasing confidence[sc, si] = sort(-confidence)

VOCevaldet.m 파일의 34 번째 라인에서 정의되었습니다.

9.23.2.7 d

for d

초기값:

```
=1:nd
% display progress
if toc>1
    fprintf('%s: pr: compute: %d/%d\n', cls, d, nd)
```

VOCevaldet.m 파일의 43 번째 라인에서 정의되었습니다.

9.23.2.8 drawnow

drawnow

VOCevaldet.m 파일의 14 번째 라인에서 정의되었습니다.

9.23.2.9 fp

false positive end end compute precision recall fp=zeros(nd,1)

VOCevaldet.m 파일의 41 번째 라인에서 정의되었습니다.

9.23.2.10 function

function[rec, prec, ap]

초기값:

```
= VOCevaldet(VOCopts, id, cls, draw)
% load test set
[gtids, t]=textread(sprintf(VOCopts.imgsetpath, VOCopts.testset), '%s %d')
```

VOCevaldet.m 파일의 1 번째 라인에서 정의되었습니다.

9.23.2.11 grid

grid

VOCevaldet.m 파일의 114 번째 라인에서 정의되었습니다.

9.23.2.12 i

end find ground truth image i

초기값:

```
=1:length(gtids)
    % display progress
    if toc>1
        fprintf('%s: pr: load: %d/%d\n',cls,i,length(gtids))
```

VOCevaldet.m 파일의 10 번째 라인에서 정의되었습니다.

9.23.2.13 ids

elseif ids ==ids(si)

VOCevaldet.m 파일의 35 번째 라인에서 정의되었습니다.

9.23.2.14 ih

ih =bi(4)-bi(2)+1

VOCevaldet.m 파일의 66 번째 라인에서 정의되었습니다.

9.23.2.15 iw

iw =bi(3)-bi(1)+1

VOCevaldet.m 파일의 65 번째 라인에서 정의되었습니다.

9.23.2.16 j

for j

초기값:

```
=1:size(gt(i).BB,2)
    bbgd=gt(i).BB(:,j)
```

VOCevaldet.m 파일의 62 번째 라인에서 정의되었습니다.

9.23.2.17 jmax

jmax =j

VOCevaldet.m 파일의 75 번째 라인에서 정의되었습니다.

9.23.2.18 nd

assign detections to ground truth objects nd=length(confidence)

VOCevaldet.m 파일의 39 번째 라인에서 정의되었습니다.

9.23.2.19 npos

npos =0

VOCevaldet.m 파일의 8 번째 라인에서 정의되었습니다.

9.23.2.20 ov

ov =iw*ih/ua

VOCevaldet.m 파일의 72 번째 라인에서 정의되었습니다.

9.23.2.21 ovmax

end end end assign detection as true positive don t care false positive if ovmax ==-inf

VOCevaldet.m 파일의 61 번째 라인에서 정의되었습니다.

9.23.2.22 prec

```
prec = tp./((fp+tp))
```

VOCevaldet.m 파일의 98 번째 라인에서 정의되었습니다.

9.23.2.23 rec

```
rec =PASreadrecord(sprintf(VOCopts.annopath,gtids{i}))
```

VOCevaldet.m 파일의 19 번째 라인에서 정의되었습니다.

9.23.2.24 results

```
end load results[ids, confidence, b1, b2, b3, b4] =textread(sprintf(VOCopts.detrespath,id,cls),'%s %f %f %f %f %f')
```

VOCevaldet.m 파일의 30 번째 라인에서 정의되었습니다.

9.23.2.25 t

```
for t
```

초기값:

```
=0:0.1:1  
p=max(prec(rec>=t))
```

VOCevaldet.m 파일의 103 번째 라인에서 정의되었습니다.

9.23.2.26 tic

```
tic
```

VOCevaldet.m 파일의 7 번째 라인에서 정의되었습니다.

9.23.2.27 tp

```
tp =zeros(nd,1)
```

VOCevaldet.m 파일의 40 번째 라인에서 정의되었습니다.

9.23.2.28 ua

if iw & ih compute overlap as area of intersection area of union ua

초기값:

```
= (bb(3) - bb(1) + 1) * (bb(4) - bb(2) + 1) + ...
    (bbgt(3) - bbgt(1) + 1) * (bbgt(4) - bbgt(2) + 1) - ...
iw * ih
```

VOCevaldet.m 파일의 69 번째 라인에서 정의되었습니다.

9.24 data/VOCdevkit/VOCcode/VOCevallayout.m 파일 참조

함수

- if ~isempty (rec.objects)[rec.objects.detected]
- end gt (i)
- end load results fprintf ('layout pr:loading results...\n')
- if isempty (i) error('unrecognized image "%s"')
- elseif length (i) > 1 error('multiple image "%s"')
- max (bb(2), bbgt(2))
- min (bb(3), bbgt(3))
- min (bb(4), bbgt(4))]
- dp (strmatch(VOCopts.parts{k},{d.part.class}, 'exact'))
- M (k, l)
- end end end valid assignments for all part types tp (di)
- each part matchable and sufficient matches if (all(any(M(:, v))) && sum(any(M(:, v), 2)) >= sum(v))
- tp(di)=0
- fp (di)
- if isempty (p) p=0
- end if draw plot precision recall plot (rec, prec, '-')
- xlabel recall ylabel precision title (sprintf('class:%s, subset:%s, AP=%.3f', cls, VOCopts.testset, ap))

변수

- function [rec, prec, ap]
- npos = 0
- tic
- for i
- drawnow
- end rec = PASreadrecord(sprintf(VOCopts.annopath, gtids{i}))
- rec.objects = rec.objects(strmatch(cls, {rec.objects(:).class}, 'exact'))
- res = VOCreadxml(sprintf(VOCopts.layout.respath, id, cls))
- sort detections by decreasing confidence [t, si] = sort(-str2double({res.results.layout.confidence}))
- assign detections to ground truth objects nd = length(si)
- tp = zeros(nd, 1)
- fp = zeros(nd, 1)
- for di
- end d = res.results.layout(si(di))
- id = d.image
- if ids {d})

- `end` assign detection to ground truth `object` if any `bb =str2double({d.bndbox.xmin d.bndbox.ymin d.bndbox.xmax d.bndbox.ymax})`
- `ovmax =-inf`
- `for j`
- `bi =[max(bb(1),bbgt(1))`
- `iw =bi(3)-bi(1)+1`
- `ih =bi(4)-bi(2)+1`
- `if iw &ih` compute overlap as area of intersection area of union `ua`
- `ov =iw*ih/ua`
- `jmax =j`
- num detected `parts`
- `dp =zeros(size(d.part))`
- `for k`
- `for l`
- `v =tp+fp>0`
- `prec =tp(v)./(fp(v)+tp(v))`
- compute average precision `ap =0`
- `for t`
- `grid`

9.24.1 함수 문서화

9.24.1.1 dp()

```
dp (
    strmatch(VOCopts.parts{k},{d.part.class}, 'exact') )
```

9.24.1.2 fp()

```
fp (
    di )
```

9.24.1.3 fprintf()

```
end load results fprintf (
    'layout pr:loading results...\n' )
```

9.24.1.4 gt()

```
end gt (
    i )
```

9.24.1.5 if()

each **part** matchable and sufficient matches if (

$$\text{all}(\text{any}(\mathbf{M}(:, \mathbf{v}))) \&\& \text{sum}(\text{any}(\mathbf{M}(:, \mathbf{v}), 2)) \geq \text{sum}(\mathbf{v}) \quad \text{[pure virtual]}$$
9.24.1.6 isempty() [1/2]

```
if isempty (
    i )
```

9.24.1.7 isempty() [2/2]

```
if isempty (
    p ) [pure virtual]
```

9.24.1.8 length()

```
elseif length (
    i )
```

9.24.1.9 M()

```
M (
    k ,
    l )
```

9.24.1.10 max()

```
max (
    bb(2) ,
    bbgt(2) )
```

9.24.1.11 min() [1/2]

```
min (
    bb(3) ,
    bbgt(3) )
```

9.24.1.12 min() [2/2]

```
min (
    bb(4) ,
    bbgt(4) )
```

9.24.1.13 plot()

```
end if draw plot precision recall plot (
    rec ,
    prec ,
    '-' )
```

9.24.1.14 title()

```
xlabel recall ylabel precision title (
    sprintf( 'class:%s, subset:%s, AP=%.3f', cls, VOCopts.testset, ap) )
```

9.24.1.15 tp()

```
end end end valid assignments for all part types tp (
    di )
```

9.24.1.16 ~isempty()

```
if ~isempty (
    rec. objects )
```

9.24.2 변수 문서화

9.24.2.1 ap

```
end ap =0
```

VOCevallayout.m 파일의 148 번째 라인에서 정의되었습니다.

9.24.2.2 bb

```
end assign detection to ground truth object if any bb =str2double({d.bndbox.xmin d.bndbox.ymin d.bndbox.xmax
d.bndbox.ymax})
```

VOCevallayout.m 파일의 57 번째 라인에서 정의되었습니다.

9.24.2.3 bi

```
bi =[max(bb(1),bbgt(1))
```

VOCevallayout.m 파일의 61 번째 라인에서 정의되었습니다.

9.24.2.4 confidence

```
sort detections by decreasing confidence[t, si] =sort(-str2double({res.results.layout.confidence}))
```

VOCevallayout.m 파일의 30 번째 라인에서 정의되었습니다.

9.24.2.5 d

```
end d =res.results.layout(si(di))
```

VOCevallayout.m 파일의 45 번째 라인에서 정의되었습니다.

9.24.2.6 di

for di

초기값:

```
=1:nd
% display progress
if toc>1
    fprintf('%s: layout pr: compute: %d/%d\n',cls,di,nd)
```

VOCevallayout.m 파일의 37 번째 라인에서 정의되었습니다.

9.24.2.7 dp

dp=zeros(size(d.part))

VOCevallayout.m 파일의 86 번째 라인에서 정의되었습니다.

9.24.2.8 drawnow

drawnow

VOCevallayout.m 파일의 12 번째 라인에서 정의되었습니다.

9.24.2.9 fp

no overlapping object end end compute precision recall fp=zeros(nd,1)

VOCevallayout.m 파일의 35 번째 라인에서 정의되었습니다.

9.24.2.10 function

function[rec, prec, ap]

초기값:

```
= VOCevallayout(VOCopts,id,cls,draw)
% load test set
[gtids,t]=textread(sprintf(VOCopts.layout.imgsetpath,VOCopts.testset),'%s %d')
```

VOCevallayout.m 파일의 1 번째 라인에서 정의되었습니다.

9.24.2.11 grid

grid

VOCevallayout.m 파일의 160 번째 라인에서 정의되었습니다.

9.24.2.12 i

find ground truth `image` i

초기값:

```
=1:length(gtids)
    % display progress
    if toc>1
        fprintf('%s: layout pr: load: %d/%d\n',cls,i,length(gtids))
```

VOCevallayout.m 파일의 8 번째 라인에서 정의되었습니다.

9.24.2.13 id

id =`d.image`

VOCevallayout.m 파일의 46 번째 라인에서 정의되었습니다.

9.24.2.14 ids

elseif ids {`d`})

VOCevallayout.m 파일의 51 번째 라인에서 정의되었습니다.

9.24.2.15 ih

ih =`bi`(4)-`bi`(2)+1

VOCevallayout.m 파일의 63 번째 라인에서 정의되었습니다.

9.24.2.16 iw

```
iw = bi(3)-bi(1)+1
```

VOCevallayout.m 파일의 62 번째 라인에서 정의되었습니다.

9.24.2.17 j

```
for j
```

초기값:

```
=1:length(gt(i).objects)
    bbgf=gt(i).objects(j).bbbox
```

VOCevallayout.m 파일의 59 번째 라인에서 정의되었습니다.

9.24.2.18 jmax

```
jmax =j
```

VOCevallayout.m 파일의 72 번째 라인에서 정의되었습니다.

9.24.2.19 k

```
for k
```

초기값:

```
=1:VOCopts.nparts
    op(strmatch(VOCopts.parts{k},{o.part.class},'exact'))=k
```

VOCevallayout.m 파일의 87 번째 라인에서 정의되었습니다.

9.24.2.20 l

```
for l
```

초기값:

```
=find(op==dp(k))
    bbgf=o.part(l).bbbox
```

VOCevallayout.m 파일의 98 번째 라인에서 정의되었습니다.

9.24.2.21 nd

assign detections to ground truth `objects` `nd=length(si)`

VOCevallayout.m 파일의 33 번째 라인에서 정의되었습니다.

9.24.2.22 npos

`npos=0`

VOCevallayout.m 파일의 6 번째 라인에서 정의되었습니다.

9.24.2.23 objects

`rec` `objects=rec.objects(strmatch(cls,{rec.objects(:).class},'exact'))`

VOCevallayout.m 파일의 16 번째 라인에서 정의되었습니다.

9.24.2.24 ov

`ov=iw*ih/ua`

VOCevallayout.m 파일의 69 번째 라인에서 정의되었습니다.

9.24.2.25 ovmax

`end end end` assign detection as true false `positive` `if` `ovmax=-inf`

VOCevallayout.m 파일의 58 번째 라인에서 정의되었습니다.

9.24.2.26 parts

`end` bag of detected parts

초기값:

```
= num gt parts?
    if length(o.part)==length(d.part)
        op=zeros(size(o.part))
```

VOCevallayout.m 파일의 83 번째 라인에서 정의되었습니다.

9.24.2.27 prec

```
prec = tp(v)./(fp(v)+tp(v))
```

VOCevallayout.m 파일의 144 번째 라인에서 정의되었습니다.

9.24.2.28 rec

```
rec = PASreadrecord(sprintf(VOCopts.annopath,gtids{i}))
```

VOCevallayout.m 파일의 15 번째 라인에서 정의되었습니다.

9.24.2.29 res

```
res = VOCreadxml(sprintf(VOCopts.layout.respath,id,cls))
```

VOCevallayout.m 파일의 27 번째 라인에서 정의되었습니다.

9.24.2.30 t

```
for t
```

초기값:

```
=0:0.1:1  
p=max(prec(rec>=t))
```

VOCevallayout.m 파일의 149 번째 라인에서 정의되었습니다.

9.24.2.31 tic

```
tic
```

VOCevallayout.m 파일의 7 번째 라인에서 정의되었습니다.

9.24.2.32 tp

```
tp=zeros(nd,1)
```

VOCevallayout.m 파일의 34 번째 라인에서 정의되었습니다.

9.24.2.33 ua

if iw & ih compute overlap as area of intersection area of union ua

초기값:

```
= (bb(3)-bb(1)+1) * (bb(4)-bb(2)+1) + ...
    (bbgt(3)-bbgt(1)+1) * (bbgt(4)-bbgt(2)+1) - ...
    iw*ih
```

VOCevallayout.m 파일의 66 번째 라인에서 정의되었습니다.

9.24.2.34 v

v = tp+fp > 0

VOCevallayout.m 파일의 142 번째 라인에서 정의되었습니다.

9.25 data/VOCdevkit/VOCcode/VOCevalseg.m 파일 참조

함수

- VOCEVALSEG Creates a confusion matrix for a [set](#) of segmentation [results](#) VOCEVALSEG (V ← OCopts, ID)

9.25.1 함수 문서화

9.25.1.1 VOCEVALSEG()

VOCEVALSEG Creates a confusion matrix for a [set](#) of segmentation [results](#) VOCEVALSEG (
 VOCopts ,
 ID)

9.26 data/VOCdevkit/VOCcode/VOCinit.m 파일 참조

변수

- clear VOCopts use VOC2006 or VOC2007 data [VOC2006](#) = false
- [set](#) true to use [VOC2006](#) data dataset if [VOC2006](#) VOCopts dataset = 'VOC2006'
- [end](#) get current directory with forward slashes [cwd](#) = cd

9.26.1 변수 문서화

9.26.1.1 cwd

`end` get current directory with forward slashes `cwd = cd`

VOCinit.m 파일의 17 번째 라인에서 정의되었습니다.

9.26.1.2 dataset

`else` VOCopts dataset = 'VOC2006'

VOCinit.m 파일의 10 번째 라인에서 정의되었습니다.

9.26.1.3 VOC2006

`clear` VOCopts use VOC2006 or VOC2007 data VOC2006 = false

VOCinit.m 파일의 5 번째 라인에서 정의되었습니다.

9.27 data/VOCdevkit/VOCcode/VOClabelcolormap.m 파일 참조

함수

- `end cmap` (i, 1)
- `cmap` (i, 2)
- `cmap` (i, 3)

변수

- VOCLABELCOLORMAP Creates a `label` color map such that adjacent indices have different colors Useful for reading and writing index images which contain large `indices`
- VOCLABELCOLORMAP Creates a `label` color map such that adjacent `indices` have different colors Useful for reading and writing index images which contain large by encoding them as RGB images `CMAP`
- `for` i
- `r` = 0
- `g` = 0
- `b` = 0
- `for` j
- `id` = bitshift(id,-3)
- `end cmap` = cmap / 255

9.27.1 함수 문서화

9.27.1.1 cmap() [1/3]

```
end cmap (
    i,
    1 )
```

9.27.1.2 cmap() [2/3]

```
cmap (
    i,
    2 )
```

9.27.1.3 cmap() [3/3]

```
cmap (
    i,
    3 )
```

9.27.2 변수 문서화

9.27.2.1 b

```
b = 0
```

VOClabelcolormap.m 파일의 13 번째 라인에서 정의되었습니다.

9.27.2.2 CMAP

VOCLABELCOLORMAP Creates a **label** color map such that adjacent **indices** have different colors Useful for reading and writing index images which contain large by encoding them as RGB images CMAP

초기값:

```
= VOCLABELCOLORMAP(N) creates a label color map with N entries.
function cmap = labelcolormap(N)
if nargin==0
    N=256
end
cmap = zeros(N,3)
```

VOClabelcolormap.m 파일의 5 번째 라인에서 정의되었습니다.

9.27.2.3 cmap

```
end cmap = cmap / 255
```

VOClabeledcolormap.m 파일의 22 번째 라인에서 정의되었습니다.

9.27.2.4 g

```
g = 0
```

VOClabeledcolormap.m 파일의 13 번째 라인에서 정의되었습니다.

9.27.2.5 i

```
for i
```

초기값:

```
=1:N  
    id = i-1
```

VOClabeledcolormap.m 파일의 12 번째 라인에서 정의되었습니다.

9.27.2.6 id

```
id = bitshift(id,-3)
```

VOClabeledcolormap.m 파일의 18 번째 라인에서 정의되었습니다.

9.27.2.7 indices

VOCLABELCOLORMAP Creates a [label](#) color map such that adjacent indices have different colors Useful for reading and writing index images which contain large indices

VOClabeledcolormap.m 파일의 2 번째 라인에서 정의되었습니다.

9.27.2.8 j

for j

초기값:

```
=0:7
r = bitor(r, bitshift(bitget(id,1),7 - j))
```

VOClabelcolormap.m 파일의 14 번째 라인에서 정의되었습니다.

9.27.2.9 r

r =0

VOClabelcolormap.m 파일의 13 번째 라인에서 정의되었습니다.

9.28 data/VOCdevkit/VOCcode/VOCreadrecxml.m 파일 참조

함수

- if isfield (o, 'pose') if strcmp(o.pose
- end if isfield (o, 'truncated') p.truncated
- end if isfield (o, 'difficult') p.difficult
- if isfield (o, 'polygon') warning('polygon unimplemented')
- end if isfield (o, 'mask') warning('mask unimplemented')
- end if isfield (o, 'part') &&~isempty(o.part) p.hasparts

변수

- function rec
- x =x.annotation
- rec size width =str2double(rec.size.width)
- rec size height =str2double(rec.size.height)
- rec size depth =str2double(rec.size.depth)
- rec segmented =strcmp(rec.segmented,'1')
- rec imgname =[x.folder '/JPEGImages/' x.filename]
- rec imgsize =str2double({x.size.width x.size.height x.size.depth})
- rec database =rec.source.database
- for i
- end function p
- if Unspecified p view =''
- else p truncated =false
- else p difficult =false
- end p label =['PAS' p.class p.view]
- end p orglabel =p.label
- p bbox =str2double({o.bndbox.xmin o.bndbox.ymin o.bndbox.xmax o.bndbox.ymax})
- p bndbox xmin =str2double(o.bndbox.xmin)
- p bndbox ymin =str2double(o.bndbox.ymin)
- p bndbox xmax =str2double(o.bndbox.xmax)
- p bndbox ymax =str2double(o.bndbox.ymax)
- p polygon =[]
- p mask =[]
- end else p hasparts =false
- p part =[]

9.28.1 함수 문서화

9.28.1.1 isfield() [1/6]

```
end if isfield (
    o ,
    'difficult' )
```

9.28.1.2 isfield() [2/6]

```
end if isfield (
    o ,
    'mask' )
```

9.28.1.3 isfield() [3/6]

```
end if isfield (
    o ,
    'part' ) &&
```

9.28.1.4 isfield() [4/6]

```
if isfield (
    o ,
    'polygon' )
```

9.28.1.5 isfield() [5/6]

```
if isfield (
    o ,
    'pose' )
```

9.28.1.6 isfield() [6/6]

```
end if isfield (
    o ,
    'truncated' )
```

9.28.2 변수 문서화

9.28.2.1 bbox

```
p bbox =str2double({o.bndbox.xmin o.bndbox.ymin o.bndbox.xmax o.bndbox.ymax})
```

VOCreadrecxml.m 파일의 58 번째 라인에서 정의되었습니다.

9.28.2.2 database

```
rec database =rec.source.database
```

VOCreadrecxml.m 파일의 16 번째 라인에서 정의되었습니다.

9.28.2.3 depth

```
rec size depth =str2double(rec.size.depth)
```

VOCreadrecxml.m 파일의 10 번째 라인에서 정의되었습니다.

9.28.2.4 difficult

```
else p difficult =false
```

VOCreadrecxml.m 파일의 45 번째 라인에서 정의되었습니다.

9.28.2.5 hasparts

```
end else p hasparts =false
```

VOCreadrecxml.m 파일의 85 번째 라인에서 정의되었습니다.

9.28.2.6 height

```
rec.size.height =str2double(rec.size.height)
```

VOCreadrecxml.m 파일의 9 번째 라인에서 정의되었습니다.

9.28.2.7 i

```
for i
```

초기값:

```
=1:length(x.object)
    rec.objects(i)=xmlobjtopas(x.object(i))
```

VOCreadrecxml.m 파일의 18 번째 라인에서 정의되었습니다.

9.28.2.8 imgname

```
rec.imgname =[x.folder '/JPEGImages/' x.filename]
```

VOCreadrecxml.m 파일의 14 번째 라인에서 정의되었습니다.

9.28.2.9 imgsize

```
rec.imgsize =str2double({x.size.width x.size.height x.size.depth})
```

VOCreadrecxml.m 파일의 15 번째 라인에서 정의되었습니다.

9.28.2.10 label

```
end if p.difficult p.label =['PAS' p.class p.view]
```

VOCreadrecxml.m 파일의 48 번째 라인에서 정의되었습니다.

9.28.2.11 mask

```
else p.mask =[]
```

VOCreadrecxml.m 파일의 74 번째 라인에서 정의되었습니다.

9.28.2.12 orglabel

```
end p orglabel =p.label
```

VOCreadrecxml.m 파일의 56 번째 라인에서 정의되었습니다.

9.28.2.13 p

```
end function p
```

초기값:

```
= xmlobjtopas(o)  
p.class=o.name
```

VOCreadrecxml.m 파일의 22 번째 라인에서 정의되었습니다.

9.28.2.14 part

```
p part=[]
```

VOCreadrecxml.m 파일의 86 번째 라인에서 정의되었습니다.

9.28.2.15 polygon

```
else p polygon=[]
```

VOCreadrecxml.m 파일의 67 번째 라인에서 정의되었습니다.

9.28.2.16 rec

```
rec
```

초기값:

```
= VOCreadrecxml(path)  
x=VOCreadxml(path)
```

VOCreadrecxml.m 파일의 1 번째 라인에서 정의되었습니다.

9.28.2.17 segmented

```
rec segmented =strcmp(rec.segmented,'1')
```

VOCreadrecxml.m 파일의 12 번째 라인에서 정의되었습니다.

9.28.2.18 truncated

```
else p truncated =false
```

VOCreadrecxml.m 파일의 39 번째 라인에서 정의되었습니다.

9.28.2.19 view

```
end else p view =''
```

VOCreadrecxml.m 파일의 28 번째 라인에서 정의되었습니다.

9.28.2.20 width

```
rec size width =str2double(rec.size.width)
```

VOCreadrecxml.m 파일의 8 번째 라인에서 정의되었습니다.

9.28.2.21 x

```
x =x.annotation
```

VOCreadrecxml.m 파일의 4 번째 라인에서 정의되었습니다.

9.28.2.22 xmax

```
p bndbox xmax =str2double(o.bndbox.xmax)
```

VOCreadrecxml.m 파일의 62 번째 라인에서 정의되었습니다.

9.28.2.23 xmin

```
p bndbox xmin =str2double(o.bndbox.xmin)
```

VOCreadrecxml.m 파일의 60 번째 라인에서 정의되었습니다.

9.28.2.24 ymax

```
p bndbox ymax =str2double(o.bndbox.ymax)
```

VOCreadrecxml.m 파일의 63 번째 라인에서 정의되었습니다.

9.28.2.25 ymin

```
p bndbox ymin =str2double(o.bndbox.ymin)
```

VOCreadrecxml.m 파일의 61 번째 라인에서 정의되었습니다.

9.29 data/VOCdevkit/VOCcode/VOCreadxml.m 파일 참조

변수

- `function rec`
- `else f =fopen(path,'r')`

9.29.1 변수 문서화

9.29.1.1 f

```
else f =fopen(path,'r')
```

VOCreadxml.m 파일의 6 번째 라인에서 정의되었습니다.

9.29.1.2 rec

```
function rec
```

초기값:

```
= VOCreadxml(path)
if length(path)>5&&strcmp(path(1:5),'http:')
    xml=urlread(path)'
```

VOCreadxml.m 파일의 1 번째 라인에서 정의되었습니다.

9.30 data/VOCdevkit/VOCcode/VOCwritexml.m 파일 참조

함수

- `function VOCwritexml (rec, path) fid`
- `writexml (fid, rec, 0)`
- `fclose (fid)`
- `if ~isempty (f) if isstruct(f) for j`
- `fprintf (fid,'<%s >\n', fn{i})`
- `writexml (fid, rec.(fn{i})(j), depth+1)`
- `fprintf (fid,'%s', repmat(char(9), 1, depth))`
- `fprintf (fid,'</%s >\n', fn{i})`
- `end else if ~iscell (f) f`
- `fprintf (fid,'<%s >', fn{i})`
- `if ischar (f{j}) fprintf(fid`
- `elseif isnumeric (f{j}) && numel(f`
- `else error ('unsupported type')`

변수

- `function xml`
- `for i`
- `end for j`
- `if s`
- `if f {j})`

9.30.1 함수 문서화

9.30.1.1 error()

```
else error (
    'unsupported type' )
```


9.30.1.2 fclose()

```
fclose (
    fid )
```

9.30.1.3 fprintf() [1/4]

```
fprintf (
    fid ,
    '%s' ,
    repmat(char(9), 1, depth) )
```

9.30.1.4 fprintf() [2/4]

```
fprintf (
    fid ,
    '<%s >' ,
    fn{i} )
```

9.30.1.5 fprintf() [3/4]

```
fprintf (
    fid ,
    '<%s >\n' ,
    fn{i} )
```

9.30.1.6 fprintf() [4/4]

```
end fprintf (
    fid ,
    '</%s >\n' ,
    fn{i} )
```

9.30.1.7 ischar()

```
if ischar (
    f{j} )
```

9.30.1.8 isnumeric()

```
elseif isnumeric (
    f{j} ) &&
```

VOCwritexml.m 파일의 30 번째 라인에서 정의되었습니다.

```
30 ~{j}) && numel (f{j}) ==1
```

9.30.1.9 VOCwritexml()

```
function VOCwritexml (
    rec ,
    path )
```

9.30.1.10 writexml() [1/2]

```
writexml (
    fid ,
    rec ,
    0 )
```

9.30.1.11 writexml() [2/2]

```
writexml (
    fid ,
    rec. fn{i})(j,
    depth+ 1 )
```

9.30.1.12 ~iscell()

```
end else if ~iscell (
    f )
```

9.30.1.13 ~isempty()

```
if ~isempty (
    f )
```

9.30.2 변수 문서화

9.30.2.1 f

```
if f{j})
```

VOCwritexml.m 파일의 29 번째 라인에서 정의되었습니다.

9.30.2.2 i

```
for i
```

초기값:

```
=1:length(fn)  
f=rec.(fn{i})
```

VOCwritexml.m 파일의 10 번째 라인에서 정의되었습니다.

9.30.2.3 j

```
end for j
```

초기값:

```
=1:length(f)  
fprintf(fid,'%s',repmat(char(9),1,depth))
```

VOCwritexml.m 파일의 25 번째 라인에서 정의되었습니다.

9.30.2.4 s

```
if s
```

VOCwritexml.m 파일의 29 번째 라인에서 정의되었습니다.

9.30.2.5 xml

```
function xml
```

초기값:

```
= writexml(fid,rec,depth)  
fn=fieldnames(rec)
```

VOCwritexml.m 파일의 7 번째 라인에서 정의되었습니다.

9.31 data/VOCdevkit/VOCcode/VOCxml2struct.m 파일 참조

함수

- `if ~isempty (parent) &&xml(ind)~`
- `if ~strcmp (tag,['/' parent]) error('<%s > closed with<%s >')`

변수

- `function res`
- `function [res, ind]`
- `ind =i`
- `if parent`
- `if tag`

9.31.1 함수 문서화

9.31.1.1 ~isempty()

```
if ~isempty (
    parent ) &&
```

9.31.1.2 ~strcmp()

```
if ~strcmp (
    tag )
```

9.31.2 변수 문서화

9.31.2.1 function

```
function[res, ind]
```

초기값:

```
=parse (xml, ind, parent)
res=[]
```

VOCxml2struct.m 파일의 7 번째 라인에서 정의되었습니다.

9.31.2.2 ind

```
ind =i
```

VOCxml2struct.m 파일의 13 번째 라인에서 정의되었습니다.

9.31.2.3 parent

```
if parent
```

VOCxml2struct.m 파일의 16 번째 라인에서 정의되었습니다.

9.31.2.4 res

```
res
```

초기값:

```
= VOCxml2struct(xml)  
xml(xml==9 | xml==10 | xml==13) = []
```

VOCxml2struct.m 파일의 1 번째 라인에서 정의되었습니다.

9.31.2.5 tag

```
if tag
```

VOCxml2struct.m 파일의 16 번째 라인에서 정의되었습니다.

9.32 demo.py 파일 참조

네임스페이스

- [demo](#)

변수

- [tuple demo.VOC_BBOX_LABEL_NAMES](#)
- [demo.img = read_image\('C:/Users/user/Desktop/1.jpg'\)](#)
- [demo.anno](#)
- [demo.bbox](#)
- [demo.label](#)
- [demo.difficult](#)
- [demo.bndbox_anno](#)
- [demo.name](#)
- [demo.faster_rcnn](#)
- [demo.trainer](#)
- [demo.caffe_pretrain](#)
- [demo.ori_img_](#)
- [demo.__bboxes](#)
- [demo.__labels](#)
- [demo.__scores](#)
- [demo.visualize](#)
- [demo.output_img](#)

9.33 Evaluation.py 파일 참조

네임스페이스

- [Evaluation](#)

함수

- [def Evaluation.eval](#) (dataloader, faster_rcnn, test_num=10000)

변수

- [Evaluation.faster_rcnn](#)
- [Evaluation.testset](#)
- [Evaluation.test_data_loader](#)
- [int Evaluation.batch_size](#) = 1,
- [int Evaluation.num_workers](#) = 8,
- [bool Evaluation.shuffle](#) = False,
- [bool Evaluation.pin_memory](#) = True)
- [def Evaluation.eval_result](#) = eval(test_data_loader, faster_rcnn, test_num=1000)

9.34 img_crawling.py 파일 참조

네임스페이스

- [img_crawling](#)

함수

- def `img_crawling.imageCrawling` (keyword, dir)
구글 크롤링
- def `img_crawling.img_delete` ()
크롤링한 파일 open 여부 판단
- def `img_crawling.img_check` ()
파일 확장자 check jpg & png 만 남겨놓음

변수

- `img_crawling._create_default_https_context`

9.35 main.py 파일 참조

네임스페이스

- `main`

함수

- def `main.log_average_miss_rate` (precision, fp_cumsum, num_images)
- def `main.error` (msg)
- def `main.is_float_between_0_and_1` (value)
- def `main.voc_ap` (rec, prec)
- def `main.file_lines_to_list` (path)
- def `main.draw_text_in_image` (img, text, pos, color, line_width)
- def `main.adjust_axes` (r, t, fig, axes)
- def `main.draw_plot_func` (dictionary, n_classes, window_title, plot_title, x_label, output_path, to_show, plot_color, true_p_bar)

변수

- `main.testset` = TestDataset(opt)
- `main.test_dataloader`
- `main.faster_rcnn` = FasterRCNN()
- `main.trainer` = FasterRCNNTrainer(faster_rcnn).cuda()
- `main.pred_bboxes`
- `main.pred_labels`
- `main.pred_scores`
- `main.gt_bboxes`
- `main.gt_labels`
- `main.gt_difficults`
- list `main.sizes` = [sizes[0][0], sizes[1][0]]
- `main.pred_bboxes_`
- `main.pred_labels_`
- `main.pred_scores_`
- `main.f` = open('C:/Users/user/PycharmProjects/Faster-RCNN-Pytorch_6/input/detection-results/'+test_dataloader.dataset.db.ids[ii]+' .txt', mode='wt', encoding='utf-8')

- float `main.MINOVERLAP` = 0.5
- `main.parser` = `argparse.ArgumentParser()`
- `main.help`
- `main.action`
- `main.nargs`
- `main.type`
- `main.str`
- `main.args` = `parser.parse_args()`
- `main.ignore`
- bool `main.specific_iou_flagged` = False
- `main.GT_PATH` = `os.path.join(os.getcwd(), 'input', 'ground-truth')`
- `main.DR_PATH` = `os.path.join(os.getcwd(), 'input', 'detection-results')`
- `main.IMG_PATH` = `os.path.join(os.getcwd(), 'input', 'images-optional')`
- `main.no_animation`
- bool `main.show_animation` = False
- bool `main.draw_plot` = False
- `main.no_plot`
- string `main.TEMP_FILES_PATH` = "temp_files"
- string `main.results_files_path` = "results"
- `main.ground_truth_files_list` = `glob.glob(GT_PATH + '/*.txt')`
- dictionary `main.gt_counter_per_class` = {}
- dictionary `main.counter_images_per_class` = {}
- `main.file_id` = `txt_file.split(".txt", 1)[0]`
- `main.temp_path` = `os.path.join(DR_PATH, (file_id + ".txt"))`
- string `main.error_msg` = "Error. File not found: {} \n".format(temp_path)
- def `main.lines_list` = `file_lines_to_list(txt_file)`
- list `main.bounding_boxes` = []
- bool `main.is_difficult` = False
- list `main.already_seen_classes` = []
- `main.class_name` = `line.split()[0]`
- `main.left`
- `main.top`
- `main.right`
- `main.bottom`
- `main._difficult`
- string `main.bbox` = `left + " " + top + " " + right + " " + bottom`
- `main.gt_classes` = `list(gt_counter_per_class.keys())`
- `main.n_classes` = `len(gt_classes)`
- `main.n_args` = `len(args.set_class_iou)`
- `main.specific_iou_classes` = `args.set_class_iou[:2]`
- `main.iou_list` = `args.set_class_iou[1:2]`
- `main.dr_files_list` = `glob.glob(DR_PATH + '/*.txt')`
- def `main.lines` = `file_lines_to_list(txt_file)`
- `main.tmp_class_name`
- `main.confidence`
- `main.key`
- `main.reverse`
- float `main.sum_AP` = 0.0
- dictionary `main.ap_dictionary` = {}
- dictionary `main.lamr_dictionary` = {}
- dictionary `main.count_true_positives` = {}
- string `main.dr_file` = `TEMP_FILES_PATH + "/" + class_name + "_dr.json"`
- `main.dr_data` = `json.load(open(dr_file))`
- `main.nd` = `len(dr_data)`
- list `main.tp` = `[0] * nd`

- list `main.fp` = `[0] * nd`
- `main.ground_truth_img` = `glob.glob1(IMG_PATH, file_id + ".*")`
- `main.img` = `cv2.imread(IMG_PATH + "/" + ground_truth_img[0], cv2.IMREAD_COLOR)`
- string `main.img_cumulative_path` = `results_files_path + "/images/" + ground_truth_img[0]`
- `main.img_cumulative` = `cv2.imread(img_cumulative_path)`
- int `main.bottom_border` = 60
- list `main.BLACK` = `[0, 0, 0]`
- string `main.gt_file` = `TEMP_FILES_PATH + "/" + file_id + "_ground_truth.json"`
- `main.ground_truth_data` = `json.load(open(gt_file))`
- int `main.ovmax` = -1
- int `main.gt_match` = -1
- list `main.bb` = `[float(x) for x in detection["bbox"].split()]`
- list `main.bbgt` = `[float(x) for x in obj["bbox"].split()]`
- list `main.bi` = `[max(bb[0],bbgt[0]), max(bb[1],bbgt[1]), min(bb[2],bbgt[2]), min(bb[3],bbgt[3])]`
- list `main.iw` = `bi[2] - bi[0] + 1`
- list `main.ih` = `bi[3] - bi[1] + 1`
- tuple `main.ua`
- list `main.ov` = `iw * ih / ua`
- string `main.status` = "NO MATCH FOUND!"
- float `main.min_overlap` = MINOVERLAP
- `main.index` = `specific_iou_classes.index(class_name)`
- `main.height`
- `main.widht`
- tuple `main.white` = (255,255,255)
- tuple `main.light_blue` = (255,200,100)
- tuple `main.green` = (0,255,0)
- tuple `main.light_red` = (30,30,255)
- int `main.margin` = 10
- `main.v_pos` = `int(height - margin - (bottom_border / 2.0))`
- string `main.text` = "Image: " + `ground_truth_img[0]` + " "
- `main.line_width`
- tuple `main.color` = `light_red`
- `main._`
- `main.rank_pos` = `str(idx+1)`
- `main.font` = `cv2.FONT_HERSHEY_SIMPLEX`
- string `main.output_img_path` = `results_files_path + "/images/detections_one_by_one/" + class_name + "_detection" + str(idx) + ".jpg"`
- int `main.cumsum` = 0
- list `main.rec` = `tp[:]`
- list `main.prec` = `tp[:]`
- `main.ap`
- `main.mrec`
- `main.mprec`
- list `main.rounded_prec` = `['%.2f' % elem for elem in prec]`
- list `main.rounded_rec` = `['%.2f' % elem for elem in rec]`
- dictionary `main.n_images` = `counter_images_per_class[class_name]`
- `main.lamr`
- `main.mr`
- `main.fppi`
- `main.area_under_curve_x` = `mrec[:-1] + [mrec[-2]] + [mrec[-1]]`
- `main.area_under_curve_y` = `mprec[:-1] + [0.0] + [mprec[-1]]`
- `main.alpha`
- `main.edgecolor`
- `main.fig` = `plt.gcf()`
- `main.axes` = `plt.gca()`

- float `main.mAP` = `sum_AP / n_classes`
- dictionary `main.det_counter_per_class` = {}
- `main.dr_classes` = `list(det_counter_per_class.keys())`
- string `main.window_title` = "ground-truth-info"
- string `main.plot_title` = "ground-truth\n"
- string `main.x_label` = "Number of objects per class"
- string `main.output_path` = `results_files_path + "/ground-truth-info.png"`
- bool `main.to_show` = False
- string `main.plot_color` = 'forestgreen'
- `main.count_non_zero_values_in_dictionary` = `sum(int(x) > 0 for x in list(det_counter_per_class.values()))`
- dictionary `main.true_p_bar` = `count_true_positives`
- dictionary `main.n_det` = `det_counter_per_class[class_name]`

9.36 model/faster_rcnn.py 파일 참조

클래스

- class `model.faster_rcnn.FasterRCNN`

네임스페이스

- `model.faster_rcnn`

9.37 model/roi_module.py 파일 참조

클래스

- class `model.roi_module.VGG16RoIHead`
- class `model.roi_module.RoIPooling2D`
- class `model.roi_module.RoI`

네임스페이스

- `model.roi_module`

함수

- def `model.roi_module.normal_init` (m, mean, stddev, `truncated`=False)
- def `model.roi_module.load_kernel` (kernel_name, code, **kwargs)
- def `model.roi_module.GET_BLOCKS` (N, K=CUDA_NUM_THREADS)

변수

- `model.roi_module.Stream` = `namedtuple('Stream', ['ptr'])`
- `model.roi_module.for_each_device`
- `int model.roi_module.CUDA_NUM_THREADS` = 1024
- `string model.roi_module.kernel_forward`
- `string model.roi_module.kernel_backward`

9.38 model/rpn.py 파일 참조

클래스

- `class model.rpn.RegionProposalNetwork`

네임스페이스

- `model.rpn`

함수

- `def model.rpn.normal_init` (m, mean, stddev, `truncated`=False)

9.39 model/utils/bbox_tools.py 파일 참조

네임스페이스

- `model.utils.bbox_tools`

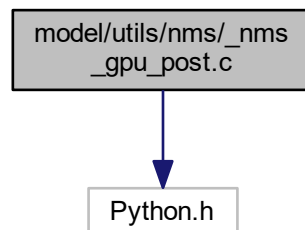
함수

- `def model.utils.bbox_tools.bbox2loc` (src_bbox, dst_bbox)
- `def model.utils.bbox_tools.bbox_iou` (bbox_a, bbox_b)
- `def model.utils.bbox_tools.loc2bbox` (src_bbox, loc)

9.40 model/utils/nms/_nms_gpu_post.c 파일 참조

`#include "Python.h"`

`_nms_gpu_post.c`에 대한 include 의존 그래프



매크로

- `#define PY_SSIZE_T_CLEAN`

9.40.1 매크로 문서화

9.40.1.1 PY_SSIZE_T_CLEAN

`#define PY_SSIZE_T_CLEAN`

`_nms_gpu_post.c` 파일의 3 번째 라인에서 정의되었습니다.

9.41 model/utils/nms/_nms_gpu_post_py.py 파일 참조

네임스페이스

- `model.utils.nms._nms_gpu_post_py`

함수

- `def model.utils.nms._nms_gpu_post_py._nms_gpu_post (mask, n_bbox, threads_per_block, col_blocks)`

9.42 model/utils/nms/build.py 파일 참조

네임스페이스

- `model.utils.nms.build`

변수

- `list model.utils.nms.build.ext_modules = [Extension("_nms_gpu_post", ["_nms_gpu_post.pyx"])]`
- `model.utils.nms.build.name`
- `model.utils.nms.build.cmdclass`

9.43 model/utils/nms/non_maximum_suppression.py 파일 참조

네임스페이스

- `model.utils.nms.non_maximum_suppression`

함수

- def [model.utils.nms.non_maximum_suppression._load_kernel](#) (kernel_name, code, options=())
- def [model.utils.nms.non_maximum_suppression.non_maximum_suppression](#) (bbox, thresh, score=None, limit=None)
- def [model.utils.nms.non_maximum_suppression._non_maximum_suppression_gpu](#) (bbox, thresh, score=None, limit=None)
- def [model.utils.nms.non_maximum_suppression._call_nms_kernel](#) (bbox, thresh)

변수

- [model.utils.nms.non_maximum_suppression.for_each_device](#)
- string [model.utils.nms.non_maximum_suppression._nms_gpu_code](#)

9.44 model/utils/roi_sample.py 파일 참조

클래스

- class [model.utils.roi_sample.ProposalTargetCreator](#)

네임스페이스

- [model.utils.roi_sample](#)

9.45 model/utils/rpn_gt_loc_label.py 파일 참조

클래스

- class [model.utils.rpn_gt_loc_label.AnchorTargetCreator](#)

네임스페이스

- [model.utils.rpn_gt_loc_label](#)

함수

- def [model.utils.rpn_gt_loc_label._unmap](#) (data, count, index, fill=0)
- def [model.utils.rpn_gt_loc_label._get_inside_index](#) (anchor, H, W)
이미지 안에있는 anchor 박스의 index 반환

9.46 model/vgg16.py 파일 참조

네임스페이스

- [model.vgg16](#)

함수

- def `model.vgg16.decom_vgg16()`
imagenet pretrain vgg16모델 정의 및 상위 4개의 layer 파라미터 고정.

9.47 scripts/extra/convert_dr_darkflow_json.py 파일 참조

네임스페이스

- `convert_dr_darkflow_json`

변수

- `convert_dr_darkflow_json.parent_path` = `os.path.abspath(os.path.join(os.getcwd(), os.pardir))`
- `convert_dr_darkflow_json.DR_PATH` = `os.path.join(parent_path, 'input', 'detection-results')`
- `convert_dr_darkflow_json.json_list` = `glob.glob('*json')`
create the backup dir if it doesn't exist already
- `convert_dr_darkflow_json.data` = `json.load(open(tmp_file))`
- `convert_dr_darkflow_json.obj_name` = `obj['label']`
- `convert_dr_darkflow_json.conf` = `obj['confidence']`
- `convert_dr_darkflow_json.left` = `obj['topleft']['x']`
- `convert_dr_darkflow_json.top` = `obj['topleft']['y']`
- `convert_dr_darkflow_json.right` = `obj['bottomright']['x']`
- `convert_dr_darkflow_json.bottom` = `obj['bottomright']['y']`

9.48 scripts/extra/convert_dr_yolo.py 파일 참조

네임스페이스

- `convert_dr_yolo`

변수

- string `convert_dr_yolo.IN_FILE` = 'result.txt'
- `convert_dr_yolo.parent_path` = `os.path.abspath(os.path.join(os.getcwd(), os.pardir))`
- `convert_dr_yolo.DR_PATH` = `os.path.join(parent_path, 'input', 'detection-results')`
- string `convert_dr_yolo.SEPARATOR_KEY` = 'Enter Image Path:'
- string `convert_dr_yolo.IMG_FORMAT` = '.jpg'
- `convert_dr_yolo.outfile` = None
- `convert_dr_yolo.image_path` = `re.search(SEPARATOR_KEY + '(.)' + IMG_FORMAT, line)`
- `convert_dr_yolo.image_name` = `os.path.basename(image_path.group(1))`
- `convert_dr_yolo.class_name`
- `convert_dr_yolo.info`
- `convert_dr_yolo.confidence`
- `convert_dr_yolo.bbox` = `bbox.replace(' ','')`
- `convert_dr_yolo.left`
- `convert_dr_yolo.top`
- `convert_dr_yolo.width`
- `convert_dr_yolo.height`
- `convert_dr_yolo.right` = `left + width`
- `convert_dr_yolo.bottom` = `top + height`

9.49 scripts/extra/convert_gt_xml.py 파일 참조

네임스페이스

- `convert_gt_xml`

변수

- `convert_gt_xml.parent_path` = `os.path.abspath(os.path.join(os.getcwd(), os.pardir))`
- `convert_gt_xml.GT_PATH` = `os.path.join(parent_path, 'input', 'ground-truth')`
- `convert_gt_xml.xml_list` = `glob.glob('*.xml')`
create the backup dir if it doesn't exist already
- `convert_gt_xml.root` = `ET.parse(tmp_file).getroot()`
- `convert_gt_xml.obj_name` = `obj.find('name').text`
- `convert_gt_xml.bndbox` = `obj.find('bndbox')`
- `convert_gt_xml.left` = `bndbox.find('xmin').text`
- `convert_gt_xml.top` = `bndbox.find('ymin').text`
- `convert_gt_xml.right` = `bndbox.find('xmax').text`
- `convert_gt_xml.bottom` = `bndbox.find('ymax').text`

9.50 scripts/extra/convert_gt_yolo.py 파일 참조

네임스페이스

- `convert_gt_yolo`

함수

- `def convert_gt_yolo.convert_yolo_coordinates_to_voc (x_c_n, y_c_n, width_n, height_n, img_width, img_height)`

변수

- `convert_gt_yolo.obj_list` = `f.readlines()`
remove whitespace characters like `\n` at the end of each line
- `convert_gt_yolo.parent_path` = `os.path.abspath(os.path.join(os.getcwd(), os.pardir))`
e.g.
- `convert_gt_yolo.GT_PATH` = `os.path.join(parent_path, 'input', 'ground-truth')`
- `convert_gt_yolo.txt_list` = `glob.glob('*.txt')`
create the backup dir if it doesn't exist already
- `convert_gt_yolo.image_name` = `tmp_file.split(".txt",1)[0]`
get name before ".txt"
- `convert_gt_yolo.img` = `cv2.imread('../images/' + fname)`
check if image exists
- `convert_gt_yolo.img_height`
get image width and height
- `convert_gt_yolo.img_width`

- `convert_gt_yolo.content = f.readlines()`
image not found
- `convert_gt_yolo.obj_id`
split a line by spaces.
- `convert_gt_yolo.x_c_n`
- `convert_gt_yolo.y_c_n`
- `convert_gt_yolo.width_n`
- `convert_gt_yolo.height_n`
- `convert_gt_yolo.obj_name = obj_list[int(obj_id)]`
- `convert_gt_yolo.left`
- `convert_gt_yolo.top`
- `convert_gt_yolo.right`
- `convert_gt_yolo.bottom`

9.51 scripts/extra/convert_keras-yolo3.py 파일 참조

네임스페이스

- `convert_keras-yolo3`
- `convert_keras`

변수

- `convert_keras-yolo3.annotation_version = datetime.datetime.now().strftime('%Y%m%d%H%M%S')`
- `convert_keras-yolo3.ap = argparse.ArgumentParser()`
- `convert_keras-yolo3.required`
- `convert_keras-yolo3.default`
- `convert_keras-yolo3.type`
- `convert_keras-yolo3.help`
- `convert_keras-yolo3.action`
- `convert_keras-yolo3.group = ap.add_mutually_exclusive_group(required=True)`
- `convert_keras-yolo3.ARGS = ap.parse_args()`
- `convert_keras-yolo3.class_map = class_file.readlines()`
- `convert_keras-yolo3.annotation_file = ARGS.gt if ARGS.gt else ARGS.dr`
- `convert_keras-yolo3.output_path`
- `convert_keras-yolo3.exist_ok`
- `convert_keras-yolo3.annot = annot.split(' ')`
- `convert_keras-yolo3.img_path = annot[0].strip()`
- `convert_keras-yolo3.annotation_dir_name = os.path.dirname(img_path)`
- `convert_keras-yolo3.destination_dir = os.path.join(ARGS.output_path, annotation_dir_name)`
- `convert_keras-yolo3.file_name = os.path.basename(img_path).replace('.jpg', '.txt')`
- `convert_keras-yolo3.output_file_path = os.path.join(destination_dir, file_name)`
- `convert_keras-yolo3.x_min`
- `convert_keras-yolo3.y_min`
- `convert_keras-yolo3.x_max`
- `convert_keras-yolo3.y_max`
- `convert_keras-yolo3.class_id`
- `string convert_keras-yolo3.out_box`
- `convert_keras-yolo3.score`

9.52 scripts/extra/find_class.py 파일 참조

네임스페이스

- [find_class](#)

함수

- def [find_class.find_class](#) (class_name)

변수

- [find_class.searching_class_name](#) = sys.argv[1]
- [find_class.parent_path](#) = os.path.abspath(os.path.join(os.getcwd(), os.pardir))
- [find_class.GT_PATH](#) = os.path.join(parent_path, 'input', 'ground-truth')
- [find_class.DR_PATH](#) = os.path.join(parent_path, 'input', 'detection-results')

9.53 scripts/extra/intersect-gt-and-dr.py 파일 참조

네임스페이스

- [intersect-gt-and-dr](#)

함수

- def [intersect-gt-and-dr.backup](#) (src_folder, backup_files, backup_folder)

변수

- [intersect-gt-and-dr.parent_path](#) = os.path.abspath(os.path.join(os.getcwd(), os.pardir))
This script ensures same number of files in ground-truth and detection-results folder.
- [intersect-gt-and-dr.GT_PATH](#) = os.path.join(parent_path, 'input', 'ground-truth')
- [intersect-gt-and-dr.DR_PATH](#) = os.path.join(parent_path, 'input', 'detection-results')
- string [intersect-gt-and-dr.backup_folder](#) = 'backup_no_matches_found'
- [intersect-gt-and-dr.gt_files](#) = glob.glob('*.txt')
- [intersect-gt-and-dr.dr_files](#) = glob.glob('*.txt')
- [intersect-gt-and-dr.gt_backup](#) = gt_files - dr_files
- [intersect-gt-and-dr.dr_backup](#) = dr_files - gt_files
- [intersect-gt-and-dr.intersection](#) = gt_files & dr_files

9.54 scripts/extra/README.md 파일 참조

9.55 Train.py 파일 참조

네임스페이스

- [Train](#)

함수

- def `Train.eval` (dataloader, faster_rcnn, test_num=1000)
"MAP"를 구하기 위한 함수

변수

- `Train.dataset` = Dataset(opt)
Main `Train` 함수
- `Train.dataloader`
- `Train.testset` = TestDataset(opt)
- `Train.test_dataloader`
- `Train.faster_rcnn` = FasterRCNN()
- `Train.trainer` = FasterRCNNTrainer(faster_rcnn).cuda()
- `Train.lr` = opt.lr
- int `Train.best_map` = 0
- list `Train.loss_list_roi_cls` = []
- list `Train.loss_list_roi_loc` = []
- list `Train.loss_list_rpn_cls` = []
- list `Train.loss_list_rpn_loc` = []
- `Train.scale` = at.scalar(scale)
- `Train.img`
- `Train.bbox`
- `Train.label`
- `Train.loss_list` = trainer.train_step(img, bbox, label, scale)
- `Train.save_optimizer`
- `Train.True`
- def `Train.eval_result` = eval(test_dataloader, faster_rcnn, test_num=opt.test_num)

9.56 trainer/trainer.py 파일 참조

클래스

- class `trainer.trainer.FasterRCNNTrainer`

네임스페이스

- `trainer.trainer`

함수

- def `trainer.trainer._smooth_l1_loss` (x, t, in_weight, sigma)
- def `trainer.trainer._fast_rcnn_loc_loss` (pred_loc, gt_loc, gt_label, sigma)

변수

- `trainer.trainer.LossTuple`

9.57 utils/anchors.py 파일 참조

네임스페이스

- [utils.anchors](#)

함수

- def [utils.anchors.generate_anchor_base](#) (side_length=16, ratios=[0.5, 1, 2], scales=[0.5, 1, 2], strides=16)
(input image size / 16) * (input image size / 16) * 9 개의 기본 anchor 생성
- def [utils.anchors.get_anchors](#) (anchor_base, feat_stride, [height](#), [width](#))
F.M 한 픽셀에 대응하는 3개의 크기 * 3개의 비율을 가지는 9개의 기본 anchor 생성
- def [utils.anchors.get_rois_from_loc_anchors](#) (anchors, rpn_locs)
RPN 네트워크를 통해 나온 anchor 좌표(상대좌표) 값을 input image에 매칭

9.58 utils/array_tool.py 파일 참조

네임스페이스

- [utils.array_tool](#)

함수

- def [utils.array_tool.tonumpy](#) (data)
- def [utils.array_tool.totensor](#) (data, cuda=True)
- def [utils.array_tool.tovariable](#) (data)
- def [utils.array_tool.scalar](#) (data)

9.59 utils/config.py 파일 참조

클래스

- class [utils.config.Config](#)

네임스페이스

- [utils.config](#)

변수

- [utils.config.opt](#) = Config()

9.60 utils/data_load.py 파일 참조

클래스

- class `utils.data_load.VOCBboxDataset`
- class `utils.data_load.VOCDataset`

네임스페이스

- `utils.data_load`

함수

- def `utils.data_load.save_pkl` (filename, f)
- def `utils.data_load.load_pkl` (filename)

변수

- tuple `utils.data_load.VOC_BBOX_LABEL_NAMES`

9.61 utils/eval_tool.py 파일 참조

네임스페이스

- `utils.eval_tool`

함수

- def `utils.eval_tool.eval_detection_voc` (pred_bboxes, pred_labels, pred_scores, gt_bboxes, gt_labels, gt_difficults=None, iou_thresh=0.5, use_07_metric=False)
- def `utils.eval_tool.calc_detection_voc_prec_rec` (pred_bboxes, pred_labels, pred_scores, gt_bboxes, gt_labels, gt_difficults=None, iou_thresh=0.5)
- def `utils.eval_tool.calc_detection_voc_ap` (prec, rec, use_07_metric=False)

9.62 utils/py_nms.py 파일 참조

네임스페이스

- `utils.py_nms`

함수

- def `utils.py_nms.py_cpu_nms` (rois, thresh)
Non-Maximum Suppression(NMS): ground truth box와 IoU가 0.7이상 겹치는 anchor 값이 많이 나오기 때문에 하나만 남기고 나머지는 지운다.

9.63 utils/vis_tool.py 파일 참조

클래스

- class [utils.vis_tool.Visualizer](#)

네임스페이스

- [utils.vis_tool](#)

함수

- def [utils.vis_tool.vis_image](#) (img, ax=None)
- def [utils.vis_tool.vis_bbox](#) (img, [bbox](#), [label](#)=None, score=None, ax=None)
- def [utils.vis_tool.fig2data](#) (fig)
- def [utils.vis_tool.fig4vis](#) (fig)
- def [utils.vis_tool.visdom_bbox](#) (*args, **kwargs)

변수

- tuple [utils.vis_tool.VOC_BBOX_LABEL_NAMES](#)

Index

- - main, [65](#)
- __call__
 - data.dataset.Transform, [201](#)
 - model.utils.roi_sample.ProposalTargetCreator, [178](#)
 - model.utils.rpn_gt_loc_label.AnchorTargetCreator, [141](#)
- __getattr__
 - utils.vis_tool.Visualizer, [209](#)
- __getitem__
 - data.dataset.Dataset, [152](#)
 - data.dataset.TestDataset, [198](#)
 - data.voc_dataset.VOCBboxDataset, [218](#)
 - utils.data_load.VOCBboxDataset, [222](#)
 - utils.data_load.VOCDataset, [225](#)
- __init__
 - data.dataset.Dataset, [152](#)
 - data.dataset.TestDataset, [197](#)
 - data.dataset.Transform, [200](#)
 - data.voc_dataset.VOCBboxDataset, [216](#)
 - model.faster_rcnn.FasterRCNN, [156](#)
 - model.roi_module.RoI, [191](#)
 - model.roi_module.RoIPooling2D, [195](#)
 - model.roi_module.VGG16RoIHead, [204](#)
 - model.rpn.RegionProposalNetwork, [184](#)
 - model.utils.roi_sample.ProposalTargetCreator, [178](#)
 - model.utils.rpn_gt_loc_label.AnchorTargetCreator, [141](#)
 - trainer.trainer.FasterRCNNTrainer, [168](#)
 - utils.data_load.VOCBboxDataset, [221](#)
 - utils.data_load.VOCDataset, [225](#)
 - utils.vis_tool.Visualizer, [209](#)
- __len__
 - data.dataset.Dataset, [153](#)
 - data.dataset.TestDataset, [198](#)
 - data.voc_dataset.VOCBboxDataset, [216](#)
 - utils.data_load.VOCBboxDataset, [221](#)
 - utils.data_load.VOCDataset, [226](#)
- __pad0__
 - viewanno.m, [252](#)
- __pad1__
 - viewanno.m, [252](#)
- __bboxes__
 - demo, [45](#)
- __calc_iious__
 - model.utils.rpn_gt_loc_label.AnchorTargetCreator, [143](#)
- __call_nms_kernel__
 - model.utils.nms.non_maximum_suppression, [99](#)
- __create_default_https_context__
 - img_crawling, [54](#)
- __create_label__
 - model.utils.rpn_gt_loc_label.AnchorTargetCreator, [144](#)
- __difficult__
 - main, [65](#)
- __fast_rcnn_loc_loss__
 - trainer.trainer, [114](#)
- __get_inside_index__
 - model.utils.rpn_gt_loc_label, [104](#)
- __labels__
 - demo, [45](#)
- __load_kernel__
 - model.utils.nms.non_maximum_suppression, [100](#)
- __nms_gpu_code__
 - model.utils.nms.non_maximum_suppression, [103](#)
- __nms_gpu_post__
 - model.utils.nms.__nms_gpu_post_py, [97](#)
- __nms_gpu_post.c__
 - PY_SSIZE_T_CLEAN, [318](#)
- __non_maximum_suppression_gpu__
 - model.utils.nms.non_maximum_suppression, [101](#)
- __parse__
 - utils.config.Config, [147](#)
- __scores__
 - demo, [45](#)
- __slice_to_bounds__
 - data.util, [36](#)
- __smooth_l1_loss__
 - trainer.trainer, [114](#)
- __state_dict__
 - utils.config.Config, [148](#)
- __suppress__
 - model.faster_rcnn.FasterRCNN, [157](#)
- __unmap__
 - model.utils.rpn_gt_loc_label, [105](#)
- __vis_kw__
 - utils.vis_tool.Visualizer, [213](#)
- ~exist
 - create_segmentations_from_detections.m, [234](#)
- ~iscell

- VOCwritexml.m, 308
- ~isempty
 - VOCevallayout.m, 288
 - VOCwritexml.m, 308
 - VOCxml2struct.m, 310
- ~strcmp
 - create_segmentations_from_detections.m, 234
 - VOCxml2struct.m, 310
- action
 - convert_keras-yolo3, 26
 - main, 66
- AddGaussianNoise
 - data.dataset, 31
- addpath
 - create_segmentations_from_detections.m, 232
 - example_classifier.m, 240
 - example_detector.m, 244
 - example_layout.m, 246
 - viewanno.m, 249
- adjust_axes
 - main, 59
- alpha
 - main, 66
- already_seen_classes
 - main, 66
- anchor_base
 - model.rpn.RegionProposalNetwork, 187
- anchor_target_creator
 - trainer.trainer.FasterRCNNTrainer, 174
- anno
 - demo, 45
- annot
 - convert_keras-yolo3, 26
- annotation_dir_name
 - convert_keras-yolo3, 26
- annotation_file
 - convert_keras-yolo3, 27
- annotation_version
 - convert_keras-yolo3, 27
- ap
 - convert_keras-yolo3, 27
 - main, 66
 - VOCevalcls.m, 274
 - VOCevaldet.m, 280
 - VOCevallayout.m, 289
- ap_dictionary
 - main, 66
- area_under_curve_x
 - main, 66
- area_under_curve_y
 - main, 67
- argmax_data
 - model.roi_module.RoI, 192
- ARGS
 - convert_keras-yolo3, 27
- args
 - main, 67
- axes
 - main, 67
- b
 - VOClabelcolormap.m, 297
- backup
 - intersect-gt-and-dr, 55
- backup_folder
 - intersect-gt-and-dr, 55
- backward
 - model.roi_module.RoI, 191
- backward_fn
 - model.roi_module.RoI, 193
- batch_size
 - Evaluation, 49
- BB
 - viewdet.m, 259
 - VOCevaldet.m, 280
- bb
 - main, 67
 - viewanno.m, 249
 - viewdet.m, 259
 - VOCevaldet.m, 280
 - VOCevallayout.m, 289
- bbgt
 - main, 67
- bbox
 - convert_dr_yolo, 15
 - create_segmentations_from_detections.m, 235
 - demo, 45
 - main, 67
 - PASemptyobject.m, 264
 - Train, 109
 - VOCreadrecxml.m, 301
- bbox2loc
 - model.utils.bbox_tools, 92
- bbox_iou
 - model.utils.bbox_tools, 94
- BBOXS
 - create_segmentations_from_detections.m, 235
- best_map
 - Train, 109
- bi
 - main, 68
 - viewdet.m, 259
 - VOCevaldet.m, 280
 - VOCevallayout.m, 289
- BLACK
 - main, 68
- blur
 - data.dataset, 32
- bndbox
 - convert_gt_xml, 19
- bndbox_anno
 - demo, 45
- bottom

- convert_dr_darkflow_json, 13
 - convert_dr_yolo, 15
 - convert_gt_xml, 19
 - convert_gt_yolo, 22
 - main, 68
- bottom_border
 - main, 68
- bounding_boxes
 - main, 68
- c
 - example_classifier.m, 241
- caffe_normalize
 - data.dataset, 32
- caffe_pretrain
 - demo, 46
 - utils.config.Config, 148
- caffe_pretrain_path
 - utils.config.Config, 148
- calc_detection_voc_ap
 - utils.eval_tool, 123
- calc_detection_voc_prec_rec
 - utils.eval_tool, 125
- case
 - PASreadrectxt.m, 270
- ChangeLuminance
 - data.dataset, 32
- class
 - PASemptyobject.m, 264
- class_id
 - convert_keras-yolo3, 27
- class_map
 - convert_keras-yolo3, 27
- class_name
 - convert_dr_yolo, 16
 - main, 68
- classifier
 - example_classifier.m, 242
 - model.roi_module.VGG16RoIHead, 206
- classim
 - create_segmentations_from_detections.m, 235
- classlabelfile
 - create_segmentations_from_detections.m, 235
- clf
 - viewanno.m, 252
- cls
 - example_layout.m, 247
- cls_loc
 - model.roi_module.VGG16RoIHead, 206
- clsinds
 - viewdet.m, 259
 - VOCevaldet.m, 280
- clsnum
 - create_segmentations_from_detections.m, 235
- CMAP
 - VOClabelcolormap.m, 297
- cmap
 - create_segmentations_from_detections.m, 236
 - VOClabelcolormap.m, 297
- cmdclass
 - model.utils.nms.build, 98
- color
 - main, 69
- comp3
 - example_segmenter.m, 248
- conf
 - convert_dr_darkflow_json, 13
 - create_segmentations_from_detections.m, 236
- confidence
 - convert_dr_yolo, 16
 - main, 69
 - viewdet.m, 259
 - VOCevaldet.m, 280
 - VOCevallayout.m, 289
- content
 - convert_gt_yolo, 22
- conv1
 - model.rpn.RegionProposalNetwork, 187
- convert_dr_darkflow_json, 13
 - bottom, 13
 - conf, 13
 - data, 13
 - DR_PATH, 14
 - json_list, 14
 - left, 14
 - obj_name, 14
 - parent_path, 14
 - right, 14
 - top, 15
- convert_dr_yolo, 15
 - bbox, 15
 - bottom, 15
 - class_name, 16
 - confidence, 16
 - DR_PATH, 16
 - height, 16
 - image_name, 16
 - image_path, 16
 - IMG_FORMAT, 17
 - IN_FILE, 17
 - info, 17
 - left, 17
 - outfile, 17
 - parent_path, 17
 - right, 18
 - SEPARATOR_KEY, 18
 - top, 18
 - width, 18
- convert_gt_xml, 18
 - bndbox, 19
 - bottom, 19
 - GT_PATH, 19

- left, 19
- obj_name, 19
- parent_path, 19
- right, 20
- root, 20
- top, 20
- xml_list, 20
- convert_gt_yolo, 20
 - bottom, 22
 - content, 22
 - convert_yolo_coordinates_to_voc, 21
 - GT_PATH, 22
 - height_n, 22
 - image_name, 22
 - img, 23
 - img_height, 23
 - img_width, 23
 - left, 23
 - obj_id, 23
 - obj_list, 24
 - obj_name, 24
 - parent_path, 24
 - right, 24
 - top, 24
 - txt_list, 25
 - width_n, 25
 - x_c_n, 25
 - y_c_n, 25
- convert_keras, 25
- convert_keras-yolo3, 26
 - action, 26
 - annot, 26
 - annotation_dir_name, 26
 - annotation_file, 27
 - annotation_version, 27
 - ap, 27
 - ARGS, 27
 - class_id, 27
 - class_map, 27
 - default, 28
 - destination_dir, 28
 - exist_ok, 28
 - file_name, 28
 - group, 28
 - help, 28
 - img_path, 29
 - out_box, 29
 - output_file_path, 29
 - output_path, 29
 - required, 29
 - score, 29
 - type, 30
 - x_max, 30
 - x_min, 30
 - y_max, 30
 - y_min, 30
- convert_yolo_coordinates_to_voc
 - convert_gt_yolo, 21
- copyfile
 - create_segmentations_from_detections.m, 232
- count_non_zero_values_in_dictionary
 - main, 69
- count_true_positives
 - main, 69
- counter_images_per_class
 - main, 69
- CREATE_SEGMENTATIONS_FROM_DETECTIONS
 - create_segmentations_from_detections.m, 232, 233
- create_segmentations_from_detections
 - create_segmentations_from_detections.m, 233
 - example_segementer.m, 248
- create_segmentations_from_detections.m
 - ~exist, 234
 - ~strcmp, 234
 - addpath, 232
 - bbox, 235
 - BBOXS, 235
 - classim, 235
 - classlabelfile, 235
 - clsnum, 235
 - cmap, 236
 - conf, 236
 - copyfile, 232
 - CREATE_SEGMENTATIONS_FROM_DETECTIONS, 232, 233
 - create_segmentations_from_detections, 233
 - detnum, 236
 - drawnow, 236
 - end, 236
 - gtclasslabelfile, 236
 - im, 233
 - imgfile, 237
 - imgid, 237
 - imgids, 237
 - imginfo, 237
 - imname, 237
 - imwrite, 233
 - ind, 237
 - instlabelfile, 238
 - isempty, 233
 - j, 238
 - png, 234
 - previd, 238
 - resultsdir, 238
 - resultsdirinst, 238
 - resultsfile, 239
 - tic, 239
 - uint8, 234
 - VOCinit, 239
- crop_bbox
 - data.util, 36
- CUDA_NUM_THREADS
 - model.roi_module, 90

- cumsum
 - main, 69
- cwd
 - VOCinit.m, 296
- d
 - example_classifier.m, 242
 - viewdet.m, 260
 - VOCevaldet.m, 281
 - VOCevallayout.m, 289
- data, 31
 - convert_dr_darkflow_json, 13
- data.dataset, 31
 - AddGaussianNoise, 31
 - blur, 32
 - caffe_normalize, 32
 - ChangeLuminance, 32
 - inverse_normalize, 33
 - preprocess, 33
 - pytorch_normalize, 35
- data.dataset.Dataset, 151
 - __getitem__, 152
 - __init__, 152
 - __len__, 153
 - db, 153
 - opt, 153
 - tsf, 154
- data.dataset.TestDataset, 197
 - __getitem__, 198
 - __init__, 197
 - __len__, 198
 - db, 198
 - opt, 198
- data.dataset.Transform, 199
 - __call__, 201
 - __init__, 200
 - max_size, 201
 - min_size, 202
- data.util, 36
 - _slice_to_bounds, 36
 - crop_bbox, 36
 - flip_bbox, 38
 - random_flip, 39
 - read_image, 40
 - resize_bbox, 42
 - translate_bbox, 43
- data.voc_dataset, 44
 - VOC_BBOX_LABEL_NAMES, 44
- data.voc_dataset.VOCBboxDataset, 214
 - __getitem__, 218
 - __init__, 216
 - __len__, 216
 - data_dir, 218
 - get_example, 216
 - ids, 218
 - label_names, 218
 - return_difficult, 218
 - use_difficult, 219
- data/___init___py, 229
 - data/dataset.py, 230
 - data/util.py, 230
 - data/voc_dataset.py, 231
 - data/VOCdevkit/create_segmentations_from_detections.m, 231
 - data/VOCdevkit/example_classifier.m, 239
 - data/VOCdevkit/example_detector.m, 244
 - data/VOCdevkit/example_layout.m, 246
 - data/VOCdevkit/example_segmenter.m, 247
 - data/VOCdevkit/viewanno.m, 248
 - data/VOCdevkit/viewdet.m, 255
 - data/VOCdevkit/VOCcode/PASemptyobject.m, 263
 - data/VOCdevkit/VOCcode/PASemptyrecord.m, 265
 - data/VOCdevkit/VOCcode/PASerrmsg.m, 266
 - data/VOCdevkit/VOCcode/PASreadrecord.m, 268
 - data/VOCdevkit/VOCcode/PASreadrectxt.m, 269
 - data/VOCdevkit/VOCcode/VOCevalcls.m, 272
 - data/VOCdevkit/VOCcode/VOCevaldet.m, 277
 - data/VOCdevkit/VOCcode/VOCevallayout.m, 285
 - data/VOCdevkit/VOCcode/VOCevalseg.m, 295
 - data/VOCdevkit/VOCcode/VOCinit.m, 295
 - data/VOCdevkit/VOCcode/VOClabelcolormap.m, 296
 - data/VOCdevkit/VOCcode/VOCreadrecxml.m, 299
 - data/VOCdevkit/VOCcode/VOCreadxml.m, 305
 - data/VOCdevkit/VOCcode/VOCwritexml.m, 306
 - data/VOCdevkit/VOCcode/VOCxml2struct.m, 310
- data_dir
 - data.voc_dataset.VOCBboxDataset, 218
 - utils.data_load.VOCBboxDataset, 222
- database
 - PASemptyrecord.m, 265
 - PASreadrectxt.m, 270
 - VOCreadrecxml.m, 301
- dataloader
 - Train, 110
- dataset
 - Train, 110
 - VOCinit.m, 296
- db
 - data.dataset.Dataset, 153
 - data.dataset.TestDataset, 198
 - utils.data_load.VOCDataset, 226
- decom_vgg16
 - model.vgg16, 106
- default
 - convert_keras-yolo3, 28
- demo, 44
 - _bboxes, 45
 - _labels, 45
 - _scores, 45
 - anno, 45

- bbox, 45
- bndbox_anno, 45
- caffe_pretrain, 46
- difficult, 46
- faster_rcnn, 46
- img, 46
- label, 46
- name, 46
- ori_img_, 47
- output_img, 47
- trainer, 47
- visualize, 47
- VOC_BBOX_LABEL_NAMES, 47
- demo.py, 311
- depth
 - VOCreadrecxml.m, 301
- destination_dir
 - convert_keras-yolo3, 28
- det_counter_per_class
 - main, 70
- detector
 - example_detector.m, 245
 - example_layout.m, 247
- detnum
 - create_segmentations_from_detections.m, 236
- di
 - VOCevallayout.m, 289
- difficult
 - demo, 46
 - PASemptyobject.m, 264
 - viewanno.m, 252
 - VOCreadrecxml.m, 301
- dn
 - example_classifier.m, 242
- dp
 - example_classifier.m, 242
 - VOCevallayout.m, 286, 290
- dr_backup
 - intersect-gt-and-dr, 55
- dr_classes
 - main, 70
- dr_data
 - main, 70
- dr_file
 - main, 70
- dr_files
 - intersect-gt-and-dr, 55
- dr_files_list
 - main, 70
- DR_PATH
 - convert_dr_darkflow_json, 14
 - convert_dr_yolo, 16
 - find_class, 51
 - intersect-gt-and-dr, 55
 - main, 70
- draw_plot
 - main, 71
- draw_plot_func
 - main, 60
- draw_text_in_image
 - main, 62
- drawnow
 - create_segmentations_from_detections.m, 236
 - example_classifier.m, 242
 - example_detector.m, 245
 - viewdet.m, 260
 - VOCevalcls.m, 274
 - VOCevaldet.m, 281
 - VOCevallayout.m, 290
- edgecolor
 - main, 71
- end
 - create_segmentations_from_detections.m, 236
 - PASerrmsg.m, 267
 - PASreadrectxt.m, 270
- epoch
 - utils.config.Config, 149
- error
 - main, 62
 - VOCwritexml.m, 306
- error_msg
 - main, 71
- eval
 - Evaluation, 48
 - Train, 107
- eval_detection_voc
 - utils.eval_tool, 128
- eval_result
 - Evaluation, 49
 - Train, 110
- Evaluation, 48
 - batch_size, 49
 - eval, 48
 - eval_result, 49
 - faster_rcnn, 49
 - num_workers, 49
 - pin_memory, 49
 - shuffle, 49
 - test_dataloader, 50
 - testset, 50
- Evaluation.py, 312
- example_classifier.m
 - addpath, 240
 - c, 241
 - classifier, 242
 - d, 242
 - dn, 242
 - dp, 242
 - drawnow, 242
 - extractor, 242
 - fclose, 240
 - FD, 240
 - fd, 243

- fid, [243](#)
- fprintf, [240](#), [241](#)
- gt, [243](#)
- I, [243](#)
- i, [243](#)
- load, [241](#)
- save, [241](#)
- test, [241](#)
- tic, [243](#)
- VOCinit, [244](#)
- example_detector.m
 - addpath, [244](#)
 - detector, [245](#)
 - drawnow, [245](#)
 - fprintf, [244](#)
 - i, [245](#)
 - rec, [245](#)
 - test, [245](#)
 - tic, [246](#)
 - VOCinit, [246](#)
- example_layout.m
 - addpath, [246](#)
 - cls, [247](#)
 - detector, [247](#)
 - test, [246](#)
 - VOCinit, [247](#)
- example_segenter.m
 - comp3, [248](#)
 - create_segmentations_from_detections, [248](#)
- exist_ok
 - convert_keras-yolo3, [28](#)
- ext_modules
 - model.utils.nms.build, [98](#)
- extractor
 - example_classifier.m, [242](#)
 - model.faster_rcnn.FasterRCNN, [164](#)
- f
 - main, [71](#)
 - VOCreadxml.m, [305](#)
 - VOCwritexml.m, [309](#)
- faster_rcnn
 - demo, [46](#)
 - Evaluation, [49](#)
 - main, [71](#)
 - Train, [110](#)
 - trainer.trainer.FasterRCNNTrainer, [174](#)
- fclose
 - example_classifier.m, [240](#)
 - VOCwritexml.m, [306](#)
- FD
 - example_classifier.m, [240](#)
- fd
 - example_classifier.m, [243](#)
- feat_stride
 - model.rpn.RegionProposalNetwork, [188](#)
- fid
 - example_classifier.m, [243](#)
- fig
 - main, [71](#)
- fig2data
 - utils.vis_tool, [132](#)
- fig4vis
 - utils.vis_tool, [133](#)
- file_id
 - main, [72](#)
- file_lines_to_list
 - main, [62](#)
- file_name
 - convert_keras-yolo3, [28](#)
- find_class, [50](#)
 - DR_PATH, [51](#)
 - find_class, [50](#)
 - GT_PATH, [51](#)
 - parent_path, [51](#)
 - searching_class_name, [51](#)
- flip_bbox
 - data.util, [38](#)
- font
 - main, [72](#)
- for_each_device
 - model.roi_module, [90](#)
 - model.utils.nms.non_maximum_suppression, [104](#)
- forward
 - model.faster_rcnn.FasterRCNN, [158](#)
 - model.roi_module.RoI, [192](#)
 - model.roi_module.RoIPooling2D, [196](#)
 - model.roi_module.VGG16RoIHead, [204](#)
 - model.rpn.RegionProposalNetwork, [184](#)
 - trainer.trainer.FasterRCNNTrainer, [168](#)
- forward_fn
 - model.roi_module.RoI, [193](#)
- fp
 - main, [72](#)
 - VOCevalcls.m, [274](#)
 - VOCevaldet.m, [278](#), [281](#)
 - VOCevallayout.m, [286](#), [290](#)
- fppi
 - main, [72](#)
- fprintf
 - example_classifier.m, [240](#), [241](#)
 - example_detector.m, [244](#)
 - PASerrmsg.m, [267](#)
 - viewanno.m, [249](#)
 - viewdet.m, [256](#)
 - VOCevallayout.m, [286](#)
 - VOCwritexml.m, [307](#)
- function
 - VOCevalcls.m, [274](#)
 - VOCevaldet.m, [281](#)
 - VOCevallayout.m, [290](#)
 - VOCxml2struct.m, [310](#)
- g
 - VOClabelcolormap.m, [298](#)
- generate_anchor_base
 - utils.anchors, [116](#)

- get_anchors
 - utils.anchors, [116](#)
- GET_BLOCKS
 - model.roi_module, [89](#)
- get_example
 - data.voc_dataset.VOCBboxDataset, [216](#)
 - utils.data_load.VOCBboxDataset, [221](#)
- get_meter_data
 - trainer.trainer.FasterRCNNTrainer, [171](#)
- get_optimizer
 - model.faster_rcnn.FasterRCNN, [158](#), [159](#)
- get_rois_from_loc_anchors
 - utils.anchors, [117](#)
- green
 - main, [72](#)
- grid
 - VOCevalcls.m, [274](#)
 - VOCevaldet.m, [281](#)
 - VOCevallayout.m, [290](#)
- ground_truth_data
 - main, [72](#)
- ground_truth_files_list
 - main, [73](#)
- ground_truth_img
 - main, [73](#)
- group
 - convert_keras-yolo3, [28](#)
- gt
 - example_classifier.m, [243](#)
 - viewdet.m, [256](#)
 - VOCevaldet.m, [278](#)
 - VOCevallayout.m, [286](#)
- gt_backup
 - intersect-gt-and-dr, [56](#)
- gt_bboxes
 - main, [73](#)
- gt_classes
 - main, [73](#)
- gt_counter_per_class
 - main, [73](#)
- gt_difficults
 - main, [73](#)
- gt_file
 - main, [74](#)
- gt_files
 - intersect-gt-and-dr, [56](#)
- gt_labels
 - main, [74](#)
- gt_match
 - main, [74](#)
- GT_PATH
 - convert_gt_xml, [19](#)
 - convert_gt_yolo, [22](#)
 - find_class, [51](#)
 - intersect-gt-and-dr, [56](#)
 - main, [74](#)
- gtclasslabelfile
 - create_segmentations_from_detections.m, [236](#)
- hasparts
 - VOCreadrecxml.m, [301](#)
- head
 - model.faster_rcnn.FasterRCNN, [164](#)
- height
 - convert_dr_yolo, [16](#)
 - main, [74](#)
 - VOCreadrecxml.m, [301](#)
- height_n
 - convert_gt_yolo, [22](#)
- help
 - convert_keras-yolo3, [28](#)
 - main, [74](#)
- I
 - example_classifier.m, [243](#)
 - viewanno.m, [253](#)
- i
 - example_classifier.m, [243](#)
 - example_detector.m, [245](#)
 - viewanno.m, [253](#)
 - viewdet.m, [260](#)
 - VOCevalcls.m, [274](#)
 - VOCevaldet.m, [282](#)
 - VOCevallayout.m, [291](#)
 - VOClabelcolormap.m, [298](#)
 - VOCreadrecxml.m, [302](#)
 - VOCwritexml.m, [309](#)
- id
 - VOCevallayout.m, [291](#)
 - VOClabelcolormap.m, [298](#)
- ids
 - data.voc_dataset.VOCBboxDataset, [218](#)
 - utils.data_load.VOCBboxDataset, [223](#)
 - viewdet.m, [260](#)
 - VOCevalcls.m, [275](#)
 - VOCevaldet.m, [282](#)
 - VOCevallayout.m, [291](#)
- if
 - PASerrmsg.m, [267](#)
 - PASreadrectxt.m, [269](#)
 - VOCevallayout.m, [287](#)
- ignore
 - main, [75](#)
- ih
 - main, [75](#)
 - viewdet.m, [260](#)
 - VOCevaldet.m, [282](#)
 - VOCevallayout.m, [291](#)
- im
 - create_segmentations_from_detections.m, [233](#)
- image
 - viewanno.m, [253](#)
 - viewdet.m, [261](#)
- image_name

- convert_dr_yolo, 16
 - convert_gt_yolo, 22
- image_path
 - convert_dr_yolo, 16
- imageCrawling
 - img_crawling, 52
- img
 - convert_gt_yolo, 23
 - demo, 46
 - main, 75
 - Train, 110
 - utils.vis_tool.Visualizer, 209
- img_check
 - img_crawling, 53
- img_crawling, 52
 - _create_default_https_context, 54
 - imageCrawling, 52
 - img_check, 53
 - img_delete, 53
- img_crawling.py, 312
- img_cumulative
 - main, 75
- img_cumulative_path
 - main, 75
- img_delete
 - img_crawling, 53
- IMG_FORMAT
 - convert_dr_yolo, 17
- img_height
 - convert_gt_yolo, 23
- img_many
 - utils.vis_tool.Visualizer, 210
- IMG_PATH
 - main, 75
- img_path
 - convert_keras-yolo3, 29
- img_width
 - convert_gt_yolo, 23
- imgfile
 - create_segmentations_from_detections.m, 237
- imgid
 - create_segmentations_from_detections.m, 237
- imgids
 - create_segmentations_from_detections.m, 237
- imginfo
 - create_segmentations_from_detections.m, 237
- imgname
 - PASreadrectxt.m, 270
 - VOCreadrecxml.m, 302
- imgsize
 - PASemptyrecord.m, 266
 - PASreadrectxt.m, 270
 - VOCreadrecxml.m, 302
- imname
 - create_segmentations_from_detections.m, 237
- imshow
 - viewanno.m, 249, 250
- imwrite
 - create_segmentations_from_detections.m, 233
- IN_FILE
 - convert_dr_yolo, 17
- in_size
 - model.roi_module.RoI, 193
- ind
 - create_segmentations_from_detections.m, 237
 - VOCxml2struct.m, 310
- index
 - main, 76
 - utils.vis_tool.Visualizer, 213
- indices
 - VOClabelcolormap.m, 298
- info
 - convert_dr_yolo, 17
- instlabelfile
 - create_segmentations_from_detections.m, 238
- intersect-gt-and-dr, 54
 - backup, 55
 - backup_folder, 55
 - dr_backup, 55
 - dr_files, 55
 - DR_PATH, 55
 - gt_backup, 56
 - gt_files, 56
 - GT_PATH, 56
 - intersection, 56
 - parent_path, 56
- intersection
 - intersect-gt-and-dr, 56
- inverse_normalize
 - data.dataset, 33
- iou_list
 - main, 76
- is_difficult
 - main, 76
- is_float_between_0_and_1
 - main, 63
- ischar
 - VOCwritexml.m, 307
- isempty
 - create_segmentations_from_detections.m, 233
 - viewdet.m, 256
 - VOCevalcls.m, 273
 - VOCevaldet.m, 278
 - VOCevallayout.m, 287
- isfield
 - VOCreadrecxml.m, 300
- isnumeric

- VOCwritexml.m, 307
- iw
 - main, 76
 - viewdet.m, 261
 - VOCevaldet.m, 282
 - VOCevallayout.m, 291
- j
 - create_segmentations_from_detections.m, 238
 - viewanno.m, 253
 - viewdet.m, 261
 - VOCevalcls.m, 275
 - VOCevaldet.m, 282
 - VOCevallayout.m, 292
 - VOClabelcolormap.m, 298
 - VOCwritexml.m, 309
- jmax
 - viewdet.m, 261
 - VOCevaldet.m, 283
 - VOCevallayout.m, 292
- json_list
 - convert_dr_darkflow_json, 14
- k
 - PASerrmsg.m, 267
 - viewanno.m, 253
 - VOCevallayout.m, 292
- kernel_backward
 - model.roi_module, 90
- kernel_forward
 - model.roi_module, 90
- key
 - main, 76
- keyboard
 - PASerrmsg.m, 268
- l
 - VOCevallayout.m, 292
- label
 - demo, 46
 - Train, 111
 - VOCreadrecxml.m, 302
- label_names
 - data.voc_dataset.VOCBboxDataset, 218
 - utils.data_load.VOCBboxDataset, 223
- lamr
 - main, 76
- lamr_dictionary
 - main, 77
- left
 - convert_dr_darkflow_json, 14
 - convert_dr_yolo, 17
 - convert_gt_xml, 19
 - convert_gt_yolo, 23
 - main, 77
- length
 - viewdet.m, 257
 - VOCevalcls.m, 273
 - VOCevaldet.m, 278
 - VOCevallayout.m, 287
- light_blue
 - main, 77
- light_red
 - main, 77
- line
 - PASreadrectxt.m, 271
- line_width
 - main, 77
- lines
 - main, 77
- lines_list
 - main, 78
- linewidth
 - viewanno.m, 254
- load
 - example_classifier.m, 241
 - trainer.trainer.FasterRCNNTrainer, 171
- load_kernel
 - model.roi_module, 89
- load_path
 - utils.config.Config, 149
- load_pkl
 - utils.data_load, 122
- load_state_dict
 - utils.vis_tool.Visualizer, 210
- loc
 - model.rpn.RegionProposalNetwork, 188
- loc2bbox
 - model.utils.bbox_tools, 95
- loc_normalize_mean
 - model.faster_rcnn.FasterRCNN, 164
 - trainer.trainer.FasterRCNNTrainer, 174
- loc_normalize_std
 - model.faster_rcnn.FasterRCNN, 164
 - trainer.trainer.FasterRCNNTrainer, 175
- log
 - utils.vis_tool.Visualizer, 211
- log_average_miss_rate
 - main, 63
- log_text
 - utils.vis_tool.Visualizer, 213
- loss_list
 - Train, 111
- loss_list_roi_cls
 - Train, 111
- loss_list_roi_loc
 - Train, 111
- loss_list_rpn_cls
 - Train, 111
- loss_list_rpn_loc
 - Train, 111
- LossTuple
 - trainer.trainer, 115
- lr
 - utils.config.Config, 149
- lr_

- Train, 112
- lr_decay
 - utils.config.Config, 149
- ls
 - viewanno.m, 254
- M
 - VOCevallayout.m, 287
- main, 57
 - _, 65
 - _difficult, 65
 - action, 66
 - adjust_axes, 59
 - alpha, 66
 - already_seen_classes, 66
 - ap, 66
 - ap_dictionary, 66
 - area_under_curve_x, 66
 - area_under_curve_y, 67
 - args, 67
 - axes, 67
 - bb, 67
 - bbgt, 67
 - bbox, 67
 - bi, 68
 - BLACK, 68
 - bottom, 68
 - bottom_border, 68
 - bounding_boxes, 68
 - class_name, 68
 - color, 69
 - confidence, 69
 - count_non_zero_values_in_dictionary, 69
 - count_true_positives, 69
 - counter_images_per_class, 69
 - cumsum, 69
 - det_counter_per_class, 70
 - dr_classes, 70
 - dr_data, 70
 - dr_file, 70
 - dr_files_list, 70
 - DR_PATH, 70
 - draw_plot, 71
 - draw_plot_func, 60
 - draw_text_in_image, 62
 - edgecolor, 71
 - error, 62
 - error_msg, 71
 - f, 71
 - faster_rcnn, 71
 - fig, 71
 - file_id, 72
 - file_lines_to_list, 62
 - font, 72
 - fp, 72
 - fpfi, 72
 - green, 72
 - ground_truth_data, 72
 - ground_truth_files_list, 73
 - ground_truth_img, 73
 - gt_bboxes, 73
 - gt_classes, 73
 - gt_counter_per_class, 73
 - gt_difficults, 73
 - gt_file, 74
 - gt_labels, 74
 - gt_match, 74
 - GT_PATH, 74
 - height, 74
 - help, 74
 - ignore, 75
 - ih, 75
 - img, 75
 - img_cumulative, 75
 - img_cumulative_path, 75
 - IMG_PATH, 75
 - index, 76
 - iou_list, 76
 - is_difficult, 76
 - is_float_between_0_and_1, 63
 - iw, 76
 - key, 76
 - lamr, 76
 - lamr_dictionary, 77
 - left, 77
 - light_blue, 77
 - light_red, 77
 - line_width, 77
 - lines, 77
 - lines_list, 78
 - log_average_miss_rate, 63
 - mAP, 78
 - margin, 78
 - min_overlap, 78
 - MINOVERLAP, 78
 - mprec, 78
 - mr, 79
 - mrec, 79
 - n_args, 79
 - n_classes, 79
 - n_det, 79
 - n_images, 79
 - nargs, 80
 - nd, 80
 - no_animation, 80
 - no_plot, 80
 - output_img_path, 80
 - output_path, 80
 - ov, 81
 - ovmax, 81
 - parser, 81
 - plot_color, 81
 - plot_title, 81
 - prec, 81
 - pred_bboxes, 82
 - pred_bboxes_, 82
 - pred_labels, 82

- pred_labels_, 82
- pred_scores, 82
- pred_scores_, 82
- rank_pos, 83
- rec, 83
- results_files_path, 83
- reverse, 83
- right, 83
- rounded_prec, 83
- rounded_rec, 84
- show_animation, 84
- sizes, 84
- specific_iou_classes, 84
- specific_iou_flagged, 84
- status, 84
- str, 85
- sum_AP, 85
- TEMP_FILES_PATH, 85
- temp_path, 85
- test_dataloader, 85
- testset, 85
- text, 86
- tmp_class_name, 86
- to_show, 86
- top, 86
- tp, 86
- trainer, 86
- true_p_bar, 87
- type, 87
- ua, 87
- v_pos, 87
- voc_ap, 64
- white, 87
- widht, 87
- window_title, 88
- x_label, 88
- main.py, 313
- mAP
 - main, 78
- margin
 - main, 78
- mask
 - PASemptyobject.m, 264
 - VOCreadexml.m, 302
- matchnum
 - PASreadrectxt.m, 271
- matchstrs
 - PASreadrectxt.m, 271
- max
 - viewdet.m, 257
 - VOCevaldet.m, 278
 - VOCevallayout.m, 287
- max_size
 - data.dataset.Transform, 201
 - utils.config.Config, 149
- min
 - viewdet.m, 257, 258
 - VOCevaldet.m, 279
 - VOCevallayout.m, 288
- min_overlap
 - main, 78
- min_size
 - data.dataset.Transform, 202
 - utils.config.Config, 149
- MINOVERLAP
 - main, 78
- model, 88
- model.faster_rcnn, 88
- model.faster_rcnn.FasterRCNN, 154
 - __init__, 156
 - _suppress, 157
 - extractor, 164
 - forward, 158
 - get_optimizer, 158, 159
 - head, 164
 - loc_normalize_mean, 164
 - loc_normalize_std, 164
 - n_class, 160
 - nms_thresh, 164
 - optimizer, 164
 - predict, 160
 - rpn, 165
 - scale_lr, 162
 - score_thresh, 165
 - use_preset, 163
- model.roi_module, 88
 - CUDA_NUM_THREADS, 90
 - for_each_device, 90
 - GET_BLOCKS, 89
 - kernel_backward, 90
 - kernel_forward, 90
 - load_kernel, 89
 - normal_init, 89
 - Stream, 91
- model.roi_module.RoI, 189
 - __init__, 191
 - argmax_data, 192
 - backward, 191
 - backward_fn, 193
 - forward, 192
 - forward_fn, 193
 - in_size, 193
 - N, 193
 - rois, 193
 - spatial_scale, 193
- model.roi_module.RoIPooling2D, 194
 - __init__, 195
 - forward, 196
 - RoI, 196
- model.roi_module.VGG16RoIHead, 202
 - __init__, 204
 - classifier, 206
 - cls_loc, 206
 - forward, 204
 - n_class, 206
 - roi, 206

- roi_size, 206
 - score, 206
 - spatial_scale, 207
- model.rpn, 91
 - normal_init, 91
- model.rpn.RegionProposalNetwork, 182
 - __init__, 184
 - anchor_base, 187
 - conv1, 187
 - feat_stride, 188
 - forward, 184
 - loc, 188
 - score, 188
- model.utils, 92
- model.utils.bbox_tools, 92
 - bbox2loc, 92
 - bbox_iou, 94
 - loc2bbox, 95
- model.utils.nms, 97
- model.utils.nms._nms_gpu_post_py, 97
 - _nms_gpu_post, 97
- model.utils.nms.build, 98
 - cmdclass, 98
 - ext_modules, 98
 - name, 99
- model.utils.nms.non_maximum_suppression, 99
 - _call_nms_kernel, 99
 - _load_kernel, 100
 - _nms_gpu_code, 103
 - _non_maximum_suppression_gpu, 101
 - for_each_device, 104
 - non_maximum_suppression, 102
- model.utils.roi_sample, 104
- model.utils.roi_sample.ProposalTargetCreator, 176
 - __call__, 178
 - __init__, 178
 - n_sample, 181
 - neg_iou_thresh_hi, 181
 - neg_iou_thresh_lo, 181
 - pos_iou_thresh, 181
 - pos_ratio, 182
- model.utils.rpn_gt_loc_label, 104
 - _get_inside_index, 104
 - _unmap, 105
- model.utils.rpn_gt_loc_label.AnchorTargetCreator, 139
 - __call__, 141
 - __init__, 141
 - _calc_iou, 143
 - _create_label, 144
 - n_sample, 145
 - neg_iou_thresh, 145
 - pos_iou_thresh, 146
 - pos_ratio, 146
- model.vgg16, 106
 - decom_vgg16, 106
- model/___init___py, 229
- model/faster_rcnn.py, 316
- model/roi_module.py, 316
- model/rpn.py, 317
- model/utils/___init___py, 229
- model/utils/bbox_tools.py, 317
- model/utils/nms/___init___py, 229
- model/utils/nms/_nms_gpu_post.c, 317
- model/utils/nms/_nms_gpu_post_py.py, 318
- model/utils/nms/build.py, 318
- model/utils/nms/non_maximum_suppression.py, 318
- model/utils/roi_sample.py, 319
- model/utils/rpn_gt_loc_label.py, 319
- model/vgg16.py, 319
- mprec
 - main, 78
- mr
 - main, 79
- mrec
 - main, 79
- N
 - model.roi_module.RoI, 193
- n_args
 - main, 79
- n_class
 - model.faster_rcnn.FasterRCNN, 160
 - model.roi_module.VGG16RoIHead, 206
- n_classes
 - main, 79
- n_det
 - main, 79
- n_images
 - main, 79
- n_sample
 - model.utils.roi_sample.ProposalTargetCreator, 181
 - model.utils.rpn_gt_loc_label.AnchorTargetCreator, 145
- name
 - demo, 46
 - model.utils.nms.build, 99
- nargs
 - main, 80
- nd
 - main, 80
 - viewdet.m, 261
 - VOCevaldet.m, 283
 - VOCevallayout.m, 292
- neg_iou_thresh
 - model.utils.rpn_gt_loc_label.AnchorTargetCreator, 145
- neg_iou_thresh_hi
 - model.utils.roi_sample.ProposalTargetCreator, 181
- neg_iou_thresh_lo
 - model.utils.roi_sample.ProposalTargetCreator, 181
- nms_thresh

- model.faster_rcnn.FasterRCNN, 164
- no_animation
 - main, 80
- no_plot
 - main, 80
- non_maximum_suppression
 - model.utils.nms.non_maximum_suppression, 102
- normal_init
 - model.roi_module, 89
 - model.rpn, 91
- notEOF
 - PASreadrectxt.m, 271
- npos
 - viewdet.m, 261
 - VOCevaldet.m, 283
 - VOCevallayout.m, 293
- num_workers
 - Evaluation, 49
 - utils.config.Config, 150
- obj_id
 - convert_gt_yolo, 23
- obj_list
 - convert_gt_yolo, 24
- obj_name
 - convert_dr_darkflow_json, 14
 - convert_gt_xml, 19
 - convert_gt_yolo, 24
- object
 - PASemptyobject.m, 264
- objects
 - PASemptyrecord.m, 266
 - PASreadrectxt.m, 269
 - viewanno.m, 250
 - VOCevallayout.m, 293
- off
 - viewanno.m, 254
 - viewdet.m, 262
- on
 - viewanno.m, 254
- opt
 - data.dataset.Dataset, 153
 - data.dataset.TestDataset, 198
 - utils.config, 121
 - utils.data_load.VOCDataset, 227
- optimizer
 - model.faster_rcnn.FasterRCNN, 164
 - trainer.trainer.FasterRCNNTrainer, 175
- orglabel
 - PASemptyobject.m, 264
 - VOCreadrecxml.m, 302
- ori_img
 - demo, 47
- out
 - VOCevalcls.m, 273, 275
- out_box
 - convert_keras-yolo3, 29
- outfile
 - convert_dr_yolo, 17
- output_file_path
 - convert_keras-yolo3, 29
- output_img
 - demo, 47
- output_img_path
 - main, 80
- output_path
 - convert_keras-yolo3, 29
 - main, 80
- ov
 - main, 81
 - viewdet.m, 262
 - VOCevaldet.m, 283
 - VOCevallayout.m, 293
- ovmax
 - main, 81
 - viewdet.m, 262
 - VOCevaldet.m, 283
 - VOCevallayout.m, 293
- ovmax < VOCOpts.minoverlap continue end % read image I=imread(sprintf(VOCOpts.imgpath, gtids{i}));% draw detection bounding box and ground truth bounding box(if any) imagesc(I);hold on;if ovmax >
 - viewdet.m, 262
- p
 - VOCreadrecxml.m, 303
- parent
 - VOCxml2struct.m, 311
- parent_path
 - convert_dr_darkflow_json, 14
 - convert_dr_yolo, 17
 - convert_gt_xml, 19
 - convert_gt_yolo, 24
 - find_class, 51
 - intersect-gt-and-dr, 56
- parser
 - main, 81
- part
 - VOCreadrecxml.m, 303
- parts
 - VOCevallayout.m, 293
- PASemptyobject.m
 - bbox, 264
 - class, 264
 - difficult, 264
 - mask, 264
 - object, 264
 - orglabel, 264
 - polygon, 265
 - truncated, 265
 - view, 265
- PASemptyrecord.m
 - database, 265
 - imgsize, 266
 - objects, 266
 - record, 266

- PASerrmsg
 - PASreadrectxt.m, [269](#)
- PASerrmsg.m
 - end, [267](#)
 - fprintf, [267](#)
 - if, [267](#)
 - k, [267](#)
 - keyboard, [268](#)
- PASmsg
 - PASreadrectxt.m, [271](#)
- PASreadrecord.m
 - rec, [268](#)
 - strcmp, [268](#)
- PASreadrectxt.m
 - case, [270](#)
 - database, [270](#)
 - end, [270](#)
 - if, [269](#)
 - imgname, [270](#)
 - imgsize, [270](#)
 - line, [271](#)
 - matchnum, [271](#)
 - matchstrs, [271](#)
 - notEOF, [271](#)
 - objects, [269](#)
 - PASerrmsg, [269](#)
 - PASmsg, [271](#)
 - record, [271](#)
 - tmp, [272](#)
 - while, [270](#)
- pause
 - viewanno.m, [254](#)
- pin_memory
 - Evaluation, [49](#)
- plot
 - utils.vis_tool.Visualizer, [211](#)
 - viewanno.m, [250](#)
 - viewdet.m, [258](#)
 - VOCevalcls.m, [273](#)
 - VOCevaldet.m, [279](#)
 - VOCevallayout.m, [288](#)
- plot_color
 - main, [81](#)
- plot_many
 - utils.vis_tool.Visualizer, [212](#)
- plot_title
 - main, [81](#)
- png
 - create_segmentations_from_detections.m, [234](#)
- polygon
 - PASemptyobject.m, [265](#)
 - VOCreadrecxml.m, [303](#)
- pos_iou_thresh
 - model.utils.roi_sample.ProposalTargetCreator, [181](#)
 - model.utils.rpn_gt_loc_label.AnchorTargetCreator, [146](#)
- pos_ratio
 - model.utils.roi_sample.ProposalTargetCreator, [182](#)
 - model.utils.rpn_gt_loc_label.AnchorTargetCreator, [146](#)
- positive
 - VOCevaldet.m, [279](#)
- prec
 - main, [81](#)
 - VOCevalcls.m, [275](#)
 - VOCevaldet.m, [283](#)
 - VOCevallayout.m, [293](#)
- pred_bboxes
 - main, [82](#)
- pred_bboxes_
 - main, [82](#)
- pred_labels
 - main, [82](#)
- pred_labels_
 - main, [82](#)
- pred_scores
 - main, [82](#)
- pred_scores_
 - main, [82](#)
- predict
 - model.faster_rcnn.FasterRCNN, [160](#)
- preprocess
 - data.dataset, [33](#)
- previd
 - create_segmentations_from_detections.m, [238](#)
- proposal_target_creator
 - trainer.trainer.FasterRCNNTrainer, [175](#)
- py_cpu_nms
 - utils.py_nms, [131](#)
- PY_SSIZE_T_CLEAN
 - _nms_gpu_post.c, [318](#)
- pytorch_normalize
 - data.dataset, [35](#)
- r
 - VOClabelcolormap.m, [299](#)
- random_flip
 - data.util, [39](#)
- rank_pos
 - main, [83](#)
- read_image
 - data.util, [40](#)
- rec
 - example_detector.m, [245](#)
 - main, [83](#)
 - PASreadrecord.m, [268](#)
 - viewdet.m, [262](#)
 - VOCevalcls.m, [275](#)
 - VOCevaldet.m, [284](#)
 - VOCevallayout.m, [294](#)
 - VOCreadrecxml.m, [303](#)
 - VOCreadxml.m, [305](#)
- recall

- VOCevalcls.m, 276
- record
 - PASemptyrecord.m, 266
 - PASreadrectxt.m, 271
- reinit
 - utils.vis_tool.Visualizer, 212
- required
 - convert_keras-yolo3, 29
- res
 - VOCevallayout.m, 294
 - VOCxml2struct.m, 311
- reset_meters
 - trainer.trainer.FasterRCNNTrainer, 172
- resize_bbox
 - data.util, 42
- results
 - viewdet.m, 263
 - VOCevalcls.m, 276
 - VOCevaldet.m, 284
- results_files_path
 - main, 83
- resultsdir
 - create_segmentations_from_detections.m, 238
- resultsdirinst
 - create_segmentations_from_detections.m, 238
- resultsfile
 - create_segmentations_from_detections.m, 239
- return_difficult
 - data.voc_dataset.VOCBboxDataset, 218
 - utils.data_load.VOCBboxDataset, 223
- reverse
 - main, 83
- right
 - convert_dr_darkflow_json, 14
 - convert_dr_yolo, 18
 - convert_gt_xml, 20
 - convert_gt_yolo, 24
 - main, 83
- RoI
 - model.roi_module.RoIPooling2D, 196
- roi
 - model.roi_module.VGG16RoIHead, 206
- roi_sigma
 - trainer.trainer.FasterRCNNTrainer, 175
 - utils.config.Config, 150
- roi_size
 - model.roi_module.VGG16RoIHead, 206
- rois
 - model.roi_module.RoI, 193
- root
 - convert_gt_xml, 20
- rounded_prec
 - main, 83
- rounded_rec
 - main, 84
- rpn
 - model.faster_rcnn.FasterRCNN, 165
- rpn_sigma
 - trainer.trainer.FasterRCNNTrainer, 175
 - utils.config.Config, 150
- s
 - VOCwritexml.m, 309
- save
 - example_classifier.m, 241
 - trainer.trainer.FasterRCNNTrainer, 172
- save_optimizer
 - Train, 112
- save_pkl
 - utils.data_load, 122
- scalar
 - utils.array_tool, 119
- scale
 - Train, 112
- scale_lr
 - model.faster_rcnn.FasterRCNN, 162
- score
 - convert_keras-yolo3, 29
 - model.roi_module.VGG16RoIHead, 206
 - model.rpn.RegionProposalNetwork, 188
- score_thresh
 - model.faster_rcnn.FasterRCNN, 165
- scripts/extra/convert_dr_darkflow_json.py, 320
- scripts/extra/convert_dr_yolo.py, 320
- scripts/extra/convert_gt_xml.py, 321
- scripts/extra/convert_gt_yolo.py, 321
- scripts/extra/convert_keras-yolo3.py, 322
- scripts/extra/find_class.py, 323
- scripts/extra/intersect-gt-and-dr.py, 323
- scripts/extra/README.md, 323
- searching_class_name
 - find_class, 51
- segmentations
 - viewanno.m, 254
- segmented
 - VOCreadrecxml.m, 303
- SEPARATOR_KEY
 - convert_dr_yolo, 18
- set
 - viewanno.m, 255
- show_animation
 - main, 84
- shuffle
 - Evaluation, 49
- sizes
 - main, 84
- spatial_scale
 - model.roi_module.RoI, 193
 - model.roi_module.VGG16RoIHead, 207
- specific_iou_classes
 - main, 84
- specific_iou_flagged
 - main, 84
- state_dict

- utils.vis_tool.Visualizer, [213](#)
- status
 - main, [84](#)
- str
 - main, [85](#)
- strcmp
 - PASreadrecord.m, [268](#)
- Stream
 - model.roi_module, [91](#)
- subplot
 - viewanno.m, [250](#), [251](#)
- sum_AP
 - main, [85](#)
- t
 - VOCevalcls.m, [276](#)
 - VOCevaldet.m, [284](#)
 - VOCevallayout.m, [294](#)
- tag
 - VOCxml2struct.m, [311](#)
- TEMP_FILES_PATH
 - main, [85](#)
- temp_path
 - main, [85](#)
- test
 - example_classifier.m, [241](#)
 - example_detector.m, [245](#)
 - example_layout.m, [246](#)
- test_dataloader
 - Evaluation, [50](#)
 - main, [85](#)
 - Train, [112](#)
- test_num
 - utils.config.Config, [150](#)
- test_num_workers
 - utils.config.Config, [150](#)
- testset
 - Evaluation, [50](#)
 - main, [85](#)
 - Train, [112](#)
- text
 - main, [86](#)
 - viewanno.m, [251](#)
- tic
 - create_segmentations_from_detections.m, [239](#)
 - example_classifier.m, [243](#)
 - example_detector.m, [246](#)
 - viewdet.m, [263](#)
 - VOCevalcls.m, [276](#)
 - VOCevaldet.m, [284](#)
 - VOCevallayout.m, [294](#)
- title
 - viewanno.m, [251](#), [252](#)
 - VOCevalcls.m, [273](#)
 - VOCevaldet.m, [279](#)
 - VOCevallayout.m, [288](#)
- tmp
 - PASreadrectxt.m, [272](#)
- tmp_class_name
 - main, [86](#)
- to_show
 - main, [86](#)
- tonumpy
 - utils.array_tool, [120](#)
- top
 - convert_dr_darkflow_json, [15](#)
 - convert_dr_yolo, [18](#)
 - convert_gt_xml, [20](#)
 - convert_gt_yolo, [24](#)
 - main, [86](#)
- totensor
 - utils.array_tool, [120](#)
- tovariable
 - utils.array_tool, [120](#)
- tp
 - main, [86](#)
 - VOCevalcls.m, [276](#)
 - VOCevaldet.m, [284](#)
 - VOCevallayout.m, [288](#), [294](#)
- Train, [107](#)
 - bbox, [109](#)
 - best_map, [109](#)
 - dataloader, [110](#)
 - dataset, [110](#)
 - eval, [107](#)
 - eval_result, [110](#)
 - faster_rcnn, [110](#)
 - img, [110](#)
 - label, [111](#)
 - loss_list, [111](#)
 - loss_list_roi_cls, [111](#)
 - loss_list_roi_loc, [111](#)
 - loss_list_rpn_cls, [111](#)
 - loss_list_rpn_loc, [111](#)
 - lr_, [112](#)
 - save_optimizer, [112](#)
 - scale, [112](#)
 - test_dataloader, [112](#)
 - testset, [112](#)
 - trainer, [113](#)
 - True, [113](#)
- train
 - utils.data_load.VOCDataset, [227](#)
- Train.py, [323](#)
- train_step
 - trainer.trainer.FasterRCNNTrainer, [173](#)
- trainer, [113](#)
 - demo, [47](#)
 - main, [86](#)
 - Train, [113](#)
- trainer.trainer, [113](#)
 - _fast_rcnn_loc_loss, [114](#)
 - _smooth_l1_loss, [114](#)
 - LossTuple, [115](#)
- trainer.trainer.FasterRCNNTrainer, [166](#)
 - __init__, [168](#)

- anchor_target_creator, 174
- faster_rcnn, 174
- forward, 168
- get_meter_data, 171
- load, 171
- loc_normalize_mean, 174
- loc_normalize_std, 175
- optimizer, 175
- proposal_target_creator, 175
- reset_meters, 172
- roi_sigma, 175
- rpn_sigma, 175
- save, 172
- train_step, 173
- update_meters, 174
- trainer/___init__.py, 229
- trainer/trainer.py, 324
- translate_bbox
 - data.util, 43
- True
 - Train, 113
- true_p_bar
 - main, 87
- truncated
 - PASemptyobject.m, 265
 - viewanno.m, 255
 - VOCreadrecxml.m, 304
- tsf
 - data.dataset.Dataset, 154
- txt_list
 - convert_gt_yolo, 25
- type
 - convert_keras-yolo3, 30
 - main, 87
- ua
 - main, 87
 - viewdet.m, 263
 - VOCevaldet.m, 284
 - VOCevallayout.m, 294
- uint8
 - create_segmentations_from_detections.m, 234
- update_meters
 - trainer.trainer.FasterRCNNTrainer, 174
- use_adam
 - utils.config.Config, 150
- use_difficult
 - data.voc_dataset.VOCBboxDataset, 219
 - utils.data_load.VOCBboxDataset, 223
- use_preset
 - model.faster_rcnn.FasterRCNN, 163
- utils, 115
- utils.anchors, 116
 - generate_anchor_base, 116
 - get_anchors, 116
 - get_rois_from_loc_anchors, 117
- utils.array_tool, 119
 - scalar, 119
 - tonumpy, 120
 - totensor, 120
 - tovariable, 120
- utils.config, 121
 - opt, 121
- utils.config.Config, 146
 - __parse, 147
 - __state_dict, 148
 - caffe_pretrain, 148
 - caffe_pretrain_path, 148
 - epoch, 149
 - load_path, 149
 - lr, 149
 - lr_decay, 149
 - max_size, 149
 - min_size, 149
 - num_workers, 150
 - roi_sigma, 150
 - rpn_sigma, 150
 - test_num, 150
 - test_num_workers, 150
 - use_adam, 150
 - voc_data_dir, 151
 - weight_decay, 151
- utils.data_load, 122
 - load_pkl, 122
 - save_pkl, 122
 - VOC_BBOX_LABEL_NAMES, 123
- utils.data_load.VOCBboxDataset, 219
 - __getitem__, 222
 - __init__, 221
 - __len__, 221
 - data_dir, 222
 - get_example, 221
 - ids, 223
 - label_names, 223
 - return_difficult, 223
 - use_difficult, 223
- utils.data_load.VOCDataset, 224
 - __getitem__, 225
 - __init__, 225
 - __len__, 226
 - db, 226
 - opt, 227
 - train, 227
- utils.eval_tool, 123
 - calc_detection_voc_ap, 123
 - calc_detection_voc_prec_rec, 125
 - eval_detection_voc, 128
- utils.py_nms, 131
 - py_cpu_nms, 131
- utils.vis_tool, 132
 - fig2data, 132
 - fig4vis, 133
 - vis_bbox, 134
 - vis_image, 136
 - visdom_bbox, 137
 - VOC_BBOX_LABEL_NAMES, 137

- utils.vis_tool.Visualizer, 207
 - __getattr__, 209
 - __init__, 209
 - _vis_kw, 213
 - img, 209
 - img_many, 210
 - index, 213
 - load_state_dict, 210
 - log, 211
 - log_text, 213
 - plot, 211
 - plot_many, 212
 - reinit, 212
 - state_dict, 213
 - vis, 214
- utils/___init__.py, 230
- utils/anchors.py, 325
- utils/array_tool.py, 325
- utils/config.py, 325
- utils/data_load.py, 326
- utils/eval_tool.py, 326
- utils/py_nms.py, 326
- utils/vis_tool.py, 327
- v
 - VOCevallayout.m, 295
- v_pos
 - main, 87
- view
 - PASemptyobject.m, 265
 - VOCreadrecxml.m, 304
- viewanno.m
 - __pad0__, 252
 - __pad1__, 252
 - addpath, 249
 - bb, 249
 - clf, 252
 - difficult, 252
 - fprintf, 249
 - I, 253
 - i, 253
 - image, 253
 - imshow, 249, 250
 - j, 253
 - k, 253
 - linewidth, 254
 - ls, 254
 - objects, 250
 - off, 254
 - on, 254
 - pause, 254
 - plot, 250
 - segmentations, 254
 - set, 255
 - subplot, 250, 251
 - text, 251
 - title, 251, 252
 - truncated, 255
 - VOCinit, 255
- viewdet.m
 - BB, 259
 - bb, 259
 - bi, 259
 - clsinds, 259
 - confidence, 259
 - d, 260
 - drawnow, 260
 - fprintf, 256
 - gt, 256
 - i, 260
 - ids, 260
 - ih, 260
 - image, 261
 - isempty, 256
 - iw, 261
 - j, 261
 - jmax, 261
 - length, 257
 - max, 257
 - min, 257, 258
 - nd, 261
 - npos, 261
 - off, 262
 - ov, 262
 - ovmax, 262
 - ovmax< VOCopts.minoverlap continue end %
 - read image I=imread(sprintf(VOCopts.imgpath,
 - gtids{i}));% draw detection bounding box
 - and ground truth bounding box(if any)
 - imagesc(I);hold on;if ovmax >, 262
 - plot, 258
 - rec, 262
 - results, 263
 - tic, 263
 - ua, 263
- vis
 - utils.vis_tool.Visualizer, 214
- vis_bbox
 - utils.vis_tool, 134
- vis_image
 - utils.vis_tool, 136
- visdom_bbox
 - utils.vis_tool, 137
- visualize
 - demo, 47
- VOC2006
 - VOCinit.m, 296
- voc_ap
 - main, 64
- VOC_BBOX_LABEL_NAMES
 - data.voc_dataset, 44
 - demo, 47
 - utils.data_load, 123
 - utils.vis_tool, 137
- voc_data_dir
 - utils.config.Config, 151
- VOCevalcls.m

- ap, 274
- drawnow, 274
- fp, 274
- function, 274
- grid, 274
- i, 274
- ids, 275
- isempty, 273
- j, 275
- length, 273
- out, 273, 275
- plot, 273
- prec, 275
- rec, 275
- recall, 276
- results, 276
- t, 276
- tic, 276
- title, 273
- tp, 276
- VOCevaldet.m
 - ap, 280
 - BB, 280
 - bb, 280
 - bi, 280
 - clsinds, 280
 - confidence, 280
 - d, 281
 - drawnow, 281
 - fp, 278, 281
 - function, 281
 - grid, 281
 - gt, 278
 - i, 282
 - ids, 282
 - ih, 282
 - isempty, 278
 - iw, 282
 - j, 282
 - jmax, 283
 - length, 278
 - max, 278
 - min, 279
 - nd, 283
 - npos, 283
 - ov, 283
 - ovmax, 283
 - plot, 279
 - positive, 279
 - prec, 283
 - rec, 284
 - results, 284
 - t, 284
 - tic, 284
 - title, 279
 - tp, 284
 - ua, 284
- VOCevallayout.m
 - ~isempty, 288
 - ap, 289
 - bb, 289
 - bi, 289
 - confidence, 289
 - d, 289
 - di, 289
 - dp, 286, 290
 - drawnow, 290
 - fp, 286, 290
 - fprintf, 286
 - function, 290
 - grid, 290
 - gt, 286
 - i, 291
 - id, 291
 - ids, 291
 - if, 287
 - ih, 291
 - isempty, 287
 - iw, 291
 - j, 292
 - jmax, 292
 - k, 292
 - l, 292
 - length, 287
 - M, 287
 - max, 287
 - min, 288
 - nd, 292
 - npos, 293
 - objects, 293
 - ov, 293
 - ovmax, 293
 - parts, 293
 - plot, 288
 - prec, 293
 - rec, 294
 - res, 294
 - t, 294
 - tic, 294
 - title, 288
 - tp, 288, 294
 - ua, 294
 - v, 295
- VOCEVALSEG
 - VOCevalseg.m, 295
- VOCevalseg.m
 - VOCEVALSEG, 295
- VOCinit
 - create_segmentations_from_detections.m, 239
 - example_classifier.m, 244
 - example_detector.m, 246
 - example_layout.m, 247
 - viewanno.m, 255
- VOCinit.m
 - cwd, 296

- dataset, 296
- VOC2006, 296
- VOClabelcolormap.m
 - b, 297
 - CMAP, 297
 - cmap, 297
 - g, 298
 - i, 298
 - id, 298
 - indices, 298
 - j, 298
 - r, 299
- VOCreadrecxml.m
 - bbox, 301
 - database, 301
 - depth, 301
 - difficult, 301
 - hasparts, 301
 - height, 301
 - i, 302
 - imgname, 302
 - imgsize, 302
 - isfield, 300
 - label, 302
 - mask, 302
 - orglabel, 302
 - p, 303
 - part, 303
 - polygon, 303
 - rec, 303
 - segmented, 303
 - truncated, 304
 - view, 304
 - width, 304
 - x, 304
 - xmax, 304
 - xmin, 304
 - ymax, 305
 - ymin, 305
- VOCreadxml.m
 - f, 305
 - rec, 305
- VOCwritexml
 - VOCwritexml.m, 308
- VOCwritexml.m
 - ~iscell, 308
 - ~isempty, 308
 - error, 306
 - f, 309
 - fclose, 306
 - fprintf, 307
 - i, 309
 - ischar, 307
 - isnumeric, 307
 - j, 309
 - s, 309
 - VOCwritexml, 308
 - writexml, 308
 - xml, 309
 - VOCxml2struct.m
 - ~isempty, 310
 - ~strcmp, 310
 - function, 310
 - ind, 310
 - parent, 311
 - res, 311
 - tag, 311
 - weight_decay
 - utils.config.Config, 151
 - while
 - PASreadrectxt.m, 270
 - white
 - main, 87
 - widht
 - main, 87
 - width
 - convert_dr_yolo, 18
 - VOCreadrecxml.m, 304
 - width_n
 - convert_gt_yolo, 25
 - window_title
 - main, 88
 - writexml
 - VOCwritexml.m, 308
 - x
 - VOCreadrecxml.m, 304
 - x_c_n
 - convert_gt_yolo, 25
 - x_label
 - main, 88
 - x_max
 - convert_keras-yolo3, 30
 - x_min
 - convert_keras-yolo3, 30
 - xmax
 - VOCreadrecxml.m, 304
 - xmin
 - VOCreadrecxml.m, 304
 - xml
 - VOCwritexml.m, 309
 - xml_list
 - convert_gt_xml, 20
 - y_c_n
 - convert_gt_yolo, 25
 - y_max
 - convert_keras-yolo3, 30
 - y_min
 - convert_keras-yolo3, 30
 - ymax
 - VOCreadrecxml.m, 305
 - ymin
 - VOCreadrecxml.m, 305