Backbone:

Resnext50 (https://pytorch.org/vision/stable/models.html#id21), without fully connected layers.

Output size	# block	
250x250x64	0	Conv2d
	1	BatchNorm2d
	2	ReLU
125x125x64	3	MaxPool2d
125x125x256	4	Bottleneck0 Bottleneck1 Bottleneck2
63x63x512	5	Bottleneck0 Bottleneck1 Bottleneck2 Bottleneck3
32x32x1024	6	Bottleneck0 Bottleneck1 Bottleneck2 Bottleneck3 Bottleneck4 Bottleneck5
16x16x2048	7	Bottleneck0 Bottleneck1 Bottleneck2

Table 1: Backbone structure (sizes specified for input=500x500)

Optimizer

`Adam: lr= 10^-4, amsgrad=True`

Loss weight

Added in `ssd\modeling\box_head\loss.py` to `classification_loss`

`RDD2020: Weight = [1,k/0.27,k/0.15,k/0.32,k/0.27]; k=0.32`

`TDT4265: Weight = [1, 1, 6 ,6 ,6]`

Augmentation:

All augmentation that has been used for testing and final training can be found at: ssd/data/transforms/transforms.py

Final augmentation used:

- RandomRotation, imported from https://github.com/Paperspace/DataAugmentationForObjectDetection/blob/master/data_aug/data_aug.py and added probability for rotation to happen = 0.3 up to 2 degrees
- RandomMirror
- SampleCrop

Config – RDD2020

From: configs/rdd2020_server_resnext.yaml

Parameter		Comment
OUT_CHANNELS	[512, 1024, 2048]	
FEATURE_MAPS	[[63, 63], [32, 32], [16, 16]]	Input: 500x500
MIN_SIZES	[[10, 10], [80, 80], [150, 150]]	Input: 500x500
MAX_SIZES	[[80, 80], [150, 150], [250, 250]]	Input: 500x500
FEATURE_MAPS	[[38, 38], [19, 19], [10, 10]]	Input: 300x300
ASPECT_RATIOS	[[2,3], [2,3], [2,3,4]]	
BOXES_PER_LOCATION	[6, 6, 8]	
IMAGE_SIZE	[500, 500] or [300, 300]	
PIXEL_STD	[0.229, 0.224, 0.225]	
PIXEL_MEAN	[0.485, 0.456, 0.406]	
BATCH_SIZE	10	
LR	1e-4	

Note: Config has been modified such that train script loads model specified in config. Should be 'resnext' for all the training

Config – TDT4265

configs/tdt4265_server_resnext.yaml

Parameter		Comment
OUT_CHANNELS	[512, 1024, 2048]	
FEATURE_MAPS	[[80, 45], [40, 23], [20, 12]]	
MIN_SIZES	[[10, 10], [60, 60], [120, 120]]	
MAX_SIZES	[[60, 60], [120, 120], [220, 220]]	
ASPECT_RATIOS	[[2,3], [2,3], [2,3,4]]	
BOXES_PER_LOCATION	[6, 6, 8]	
IMAGE_SIZE	[640, 360]	
PIXEL_STD	[0.229, 0.224, 0.225]	
PIXEL_MEAN	[0.485, 0.456, 0.406]	
BATCH_SIZE	10	
LR	1e-4	

Transfer Learning

To get checkpoint from training on RDD2020 use function tranfer.py. This will get the model weights that can be used for training the model on TDT4265 dataset.

Results using different parameters

Run name	Augmentation	Weighted loss	Input size	Priors config	Aspect ratios	mAP
Org (pink)	Mirror	No	300x300	default	default	0.3219
Aug (black)	Mirror, Effect, Erasing	No	300x300	default	default	0.3018
Loss (purple)	Mirror	Yes	300x300	default	default	0.3242
Bigger (orange)	Mirror	Yes	500x500	default	default	0.3711
bigger_aug (green)	Mirror, Rotate, Crop	Yes	500x500	default	default	0.3775
bigger_priors (blue)	Mirror, Rotate, Crop	Yes	500x500	custom	custom	0.3748