Recognition, feature space representation, tracking and performance in DCNN driven safety systems.

Coursework submission 2020

Novosibirsk State University

15 December 2020

Mukul Kumar Vishwas

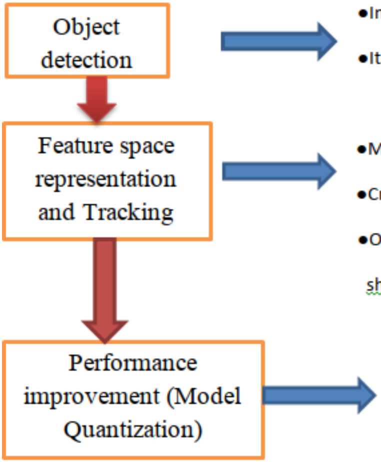
Advisor : Aleksey Okunev

Content

Problem Statement

- COVID-19 pandemic changed our life. It is deadly and cost's more than 1.2 million of lives to date. On the other hand, following some simple rules can help to control the infection. The goal of this work is to create a model and train neural network to discriminate peoples who follow the sanitary rules from those who are violating them...

Project plan



- Implement an object detection model, fine tune the model.
- It should work on images and live video feed.

- Model should able to track the objects in the video.
- Create feature space representation of objects.
- Objects should have individual representation and it should not change while the object is in the frame.

- Implement quantization of the network to improve performance.
- Smaller storage size, Smaller download size, Less memory usage, latency reduction and accelerator compatibility.

What is MMdetection

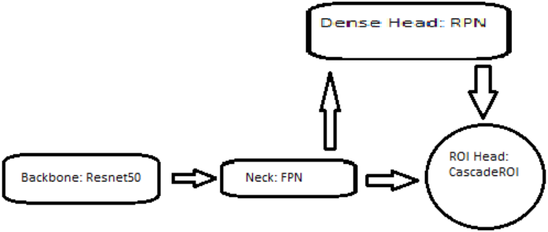
- MMdetection is an open source object detection toolbox based on PyTorch.
- It is faster in computation.
- Different detection framework's can be customize our model.
- It support multiple Datasets like XML style, COCO, PASCAL.

Why MMdetection

	MMDetection	maskrcnn-benchmark	Detectron	SimpleDet
Fast R-CNN	✓	✓	✓	✓
Faster R-CNN	✓	✓	✓	✓
Mask R-CNN	✓	✓	✓	✓
RetinaNet	✓	✓	✓	✓
DCN	✓	✓	✓	✓
DCNv2	✓	✓		
Mixed Precision Training	✓	✓		✓
Cascade R-CNN	✓		*	✓
Weight Standardization	✓	*		
Mask Scoring R-CNN	✓	*		
FCOS	✓	*		
SSD	✓			
R-FCN	✓			
M2Det	✓			
GHM	✓			
ScratchDet	✓			
Double-Head R-CNN	✓			
Grid R-CNN	✓			
FSAF	✓			
Hybrid Task Cascade	✓			
Guided Anchoring	✓			
Libra R-CNN	✓			
Generalized Attention	✓			
GCNet	✓			
HRNet	✓			
TridentNet [17]				✓

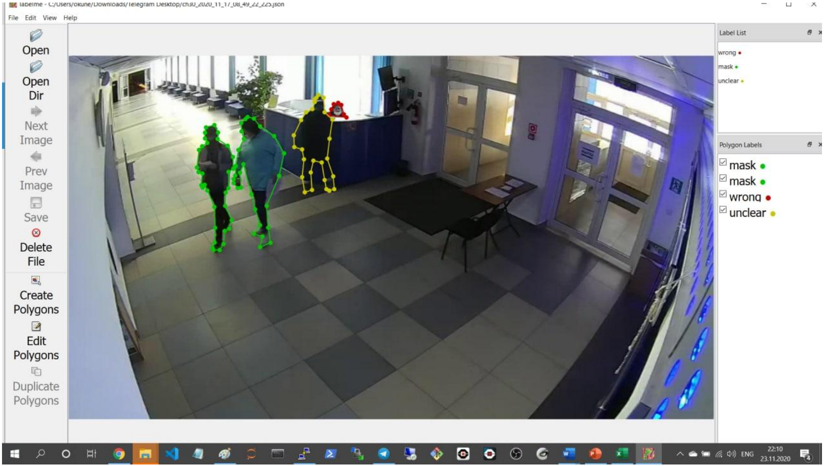
Model

- This model used resnet50 for feature extraction without the last fully connected layer.
- Feature pyramid N/w do refinements of the raw feature extracted by backbone.
- Dense head (RPN) operates on dense location of feature map.
- RoIHead is the part that takes RoI (Region of Interest) features as input and make RoI-wise task specific predictions, such as bounding box classification or regression (Shared2FCB BOX Head) , mask prediction (FCN Mask Head).

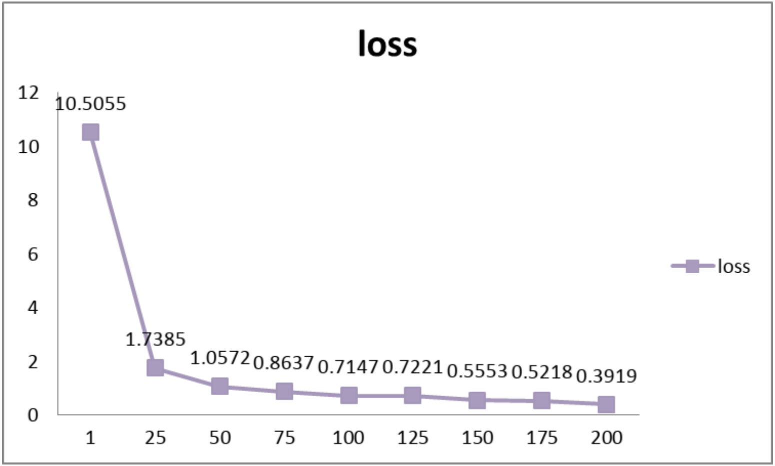


Annotation

- In this model ImageMe tool was used for the Image annotation.



Regression Loss for objection detection



Training and testing

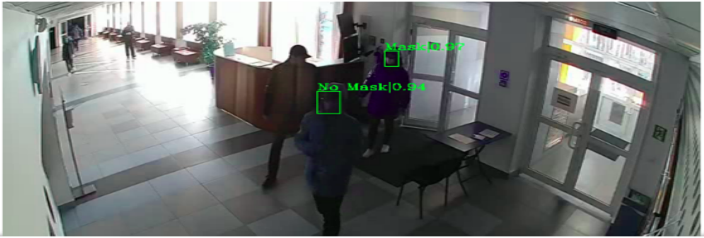
- Optimizer= SGD, lr= 0.02
- Process of Training.

```
train_pipeline = [
dict(type = 'LoadImageFromFile'),
dict(type = 'LoadAnnotations', with_bbox = True),
dict(type = 'Resize', img_scale = (1333,800),
keep_ratio = True,
dict(type = 'RandomFlip',flip_ratio = 0.5),
dict(type = 'Normalize', **img_norm_cfg),
dict(type = 'Pad', size_divisor = 32),
dict(type = 'DefaultFormatBundle'),
dict(type = 'Collect',
keys = ['img' ,'gt_bboxes', 'gt_labels'])
]
```

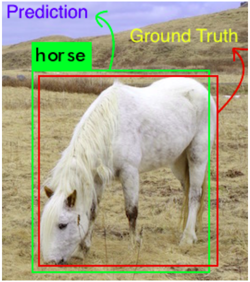
- Process of Testing.

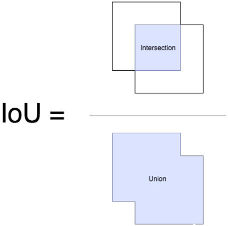
```
test_pipeline = [
dict(type = 'LoadImageFromFile'),
dict(type = 'MultiScaleFlipAug,
img_scale = (1333,800), flip = False,
transform = [
dict(type = 'Resize', keep_ratio = True),
dict(type = 'RandomFlip'),
dict(type = 'Normalize' **img_norm_cfg),
dict(type = 'Pad', size_divisor = 32),
dict(type = 'ImageToTensor', keys = ['img']),
dict(type = 'Collect', keys = ['img'])]
]
```

Result



How performance calculated for object detection



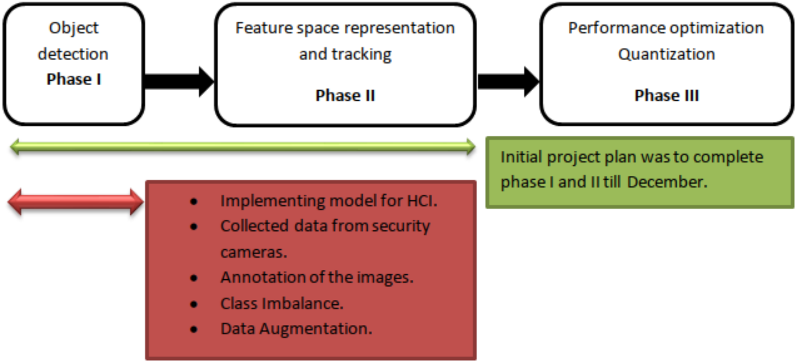


Comparison of mAP and mAR

- The performance calculated in mAP(mean Average Precision) and mAR(mean avarage Recall)
- mAP@[.5:.95] corresponds to the average AP for IoU from 0.5 to 0.95 with a step size of 0.05.

Metric	IoU	Old Val	New Val	MMdetection
mAP	@[IoU=0.50:0.95]	0.215	0.436	0.42
mAP	@[IoU=0.50]	0.289	0.631	0.63
mAP	@[IoU=0.75]	0.252	0.522	0.46
mAR	@[IoU=0.50:0.95]	0.228	0.531	
mAR	@[IoU=0.50]	0.228	0.531	
mAR	@[IoU=0.75]	0.228	0.513	

Current Progress



Future Task

- Train the model with more data to increase the accuracy.
- Feature space representation of each object so we can identify and track them on live camera video.
- Quantization of the network so it will decrease power consumption and can run of small devices like Jetsun.

Thank you for your attention.