Prediction Road Cracks on Norwegian Roads

Data Analysis

Norway
Usa
Czech
Republic
India
Japan
China

47 420 images
55000 instances

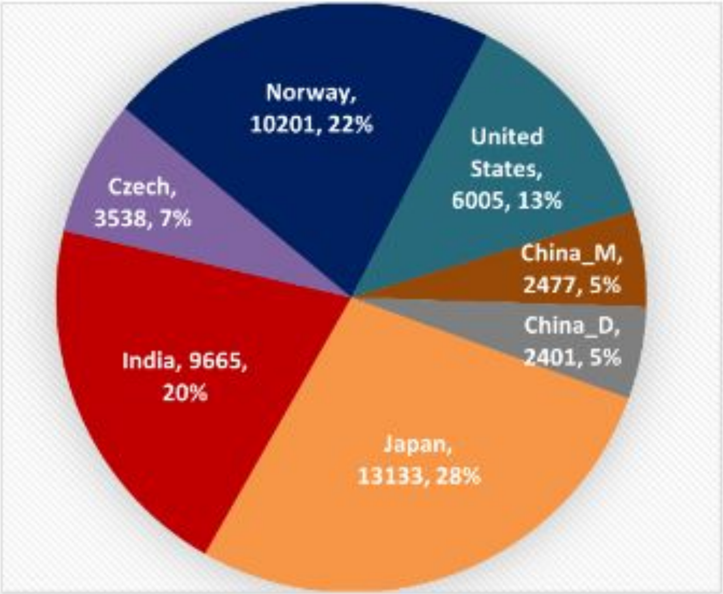


Fig. 1: Distribution of images across different countries in RDD2022

Data collected from source [1]

Data quality?

What preprocessing is necessary?

How can we finetune training to the data?

Data collected from datasett

China_motorbike	D00", "D10", "Repair", "D40", "D20"
China_drone	"Repair", "D10", "D00", "D20", "D40", "Block crack"
Czechia	"D10", "D00", "D40", "D20"
India	"D01", "D11", "D20", "D44", "D40", "D00", "D50", "D43","D10", "D0w0"
Japan	"D00", "D44", "D43", "D20", "D50", "D10", "D40"
USA	“D00", "D10", "D40", "D20"

Damage Type			***Detail***	***Class Name***
Crack	Linear Crack	Longitudinal	Wheel mark part	D00
			Construction joint part	D01
		Lateral	Equal interval	D10
			Construction joint part	D11
	Alligator Crack		Partial pavement, overall pavement	D20
Other Corruption			Rutting, bump, pothole, separation	D40
			Crosswalk blur	D43
			White line blur	D44

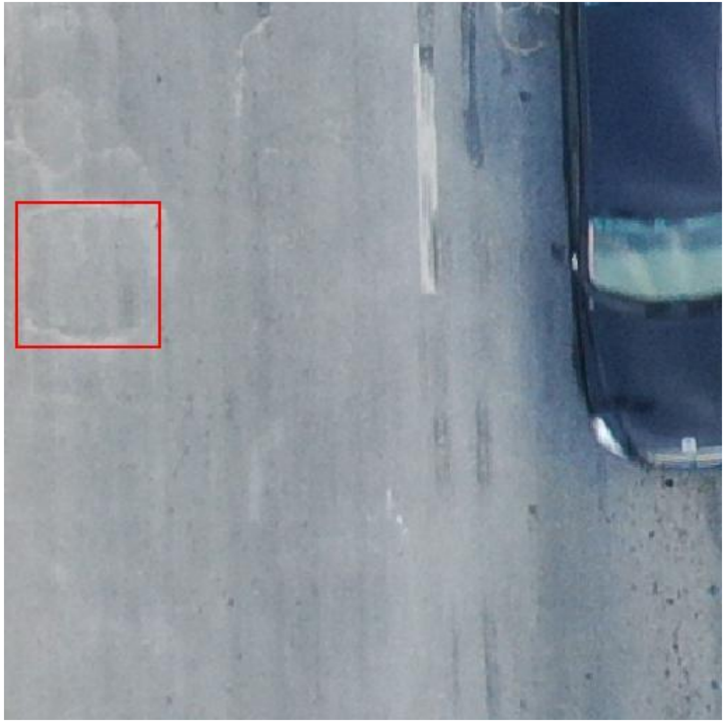
Figure collected from source [1]

D50, Repair, Block Crack and D0w0

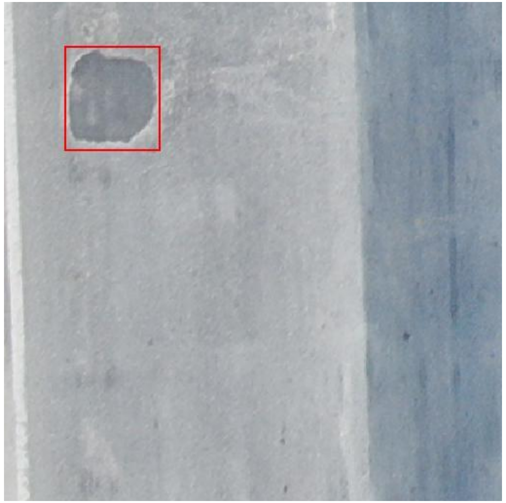
Visual inspection of D50

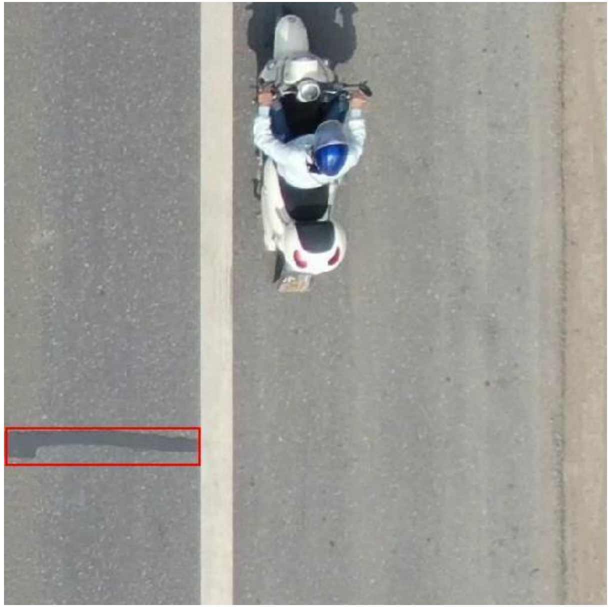
label 50 = manhole cover

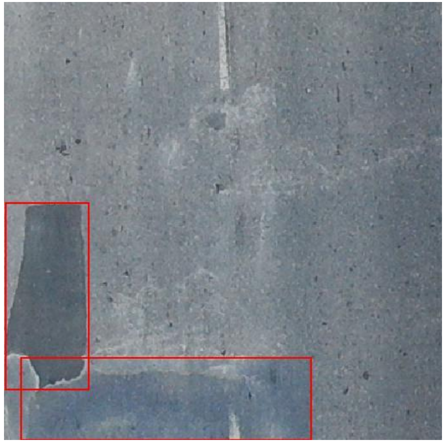
Inspection of Block Crack

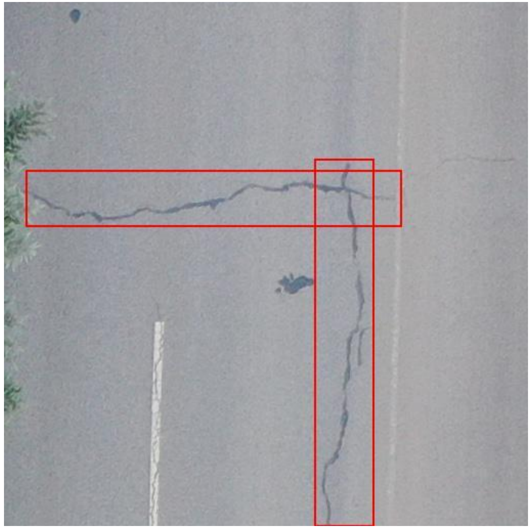


Visual Inspection of Repair









Class Names (All Countries)
D00
D01
D10
D11
D20
D40
D43
D44
D50
Repair
Block Crack
D0w0

Class Name (Norway)
D00
D10
D20
D40

Data collected from datasett

Should we include all the countries?

Quick comparison of the different countries and Norway

Conclusion: Include all countries

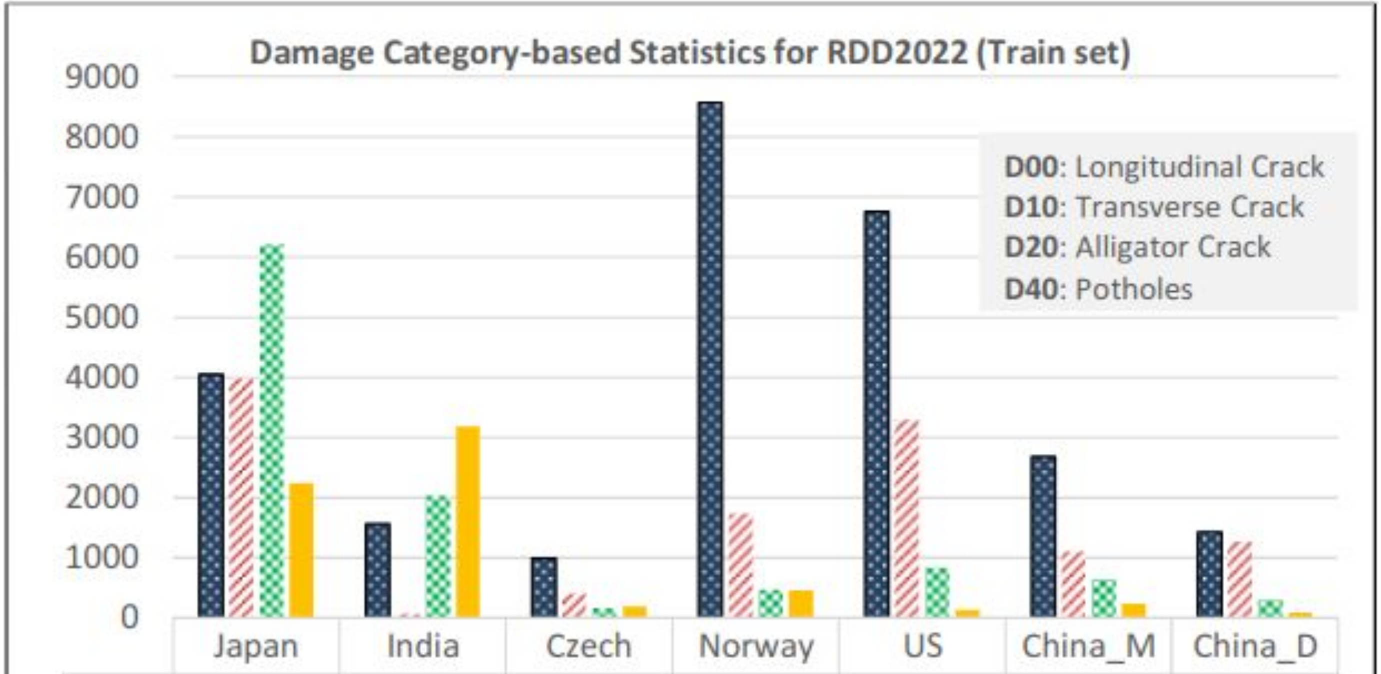
Different backgrounds

Different perspectives

Different Iluminations

More diversity

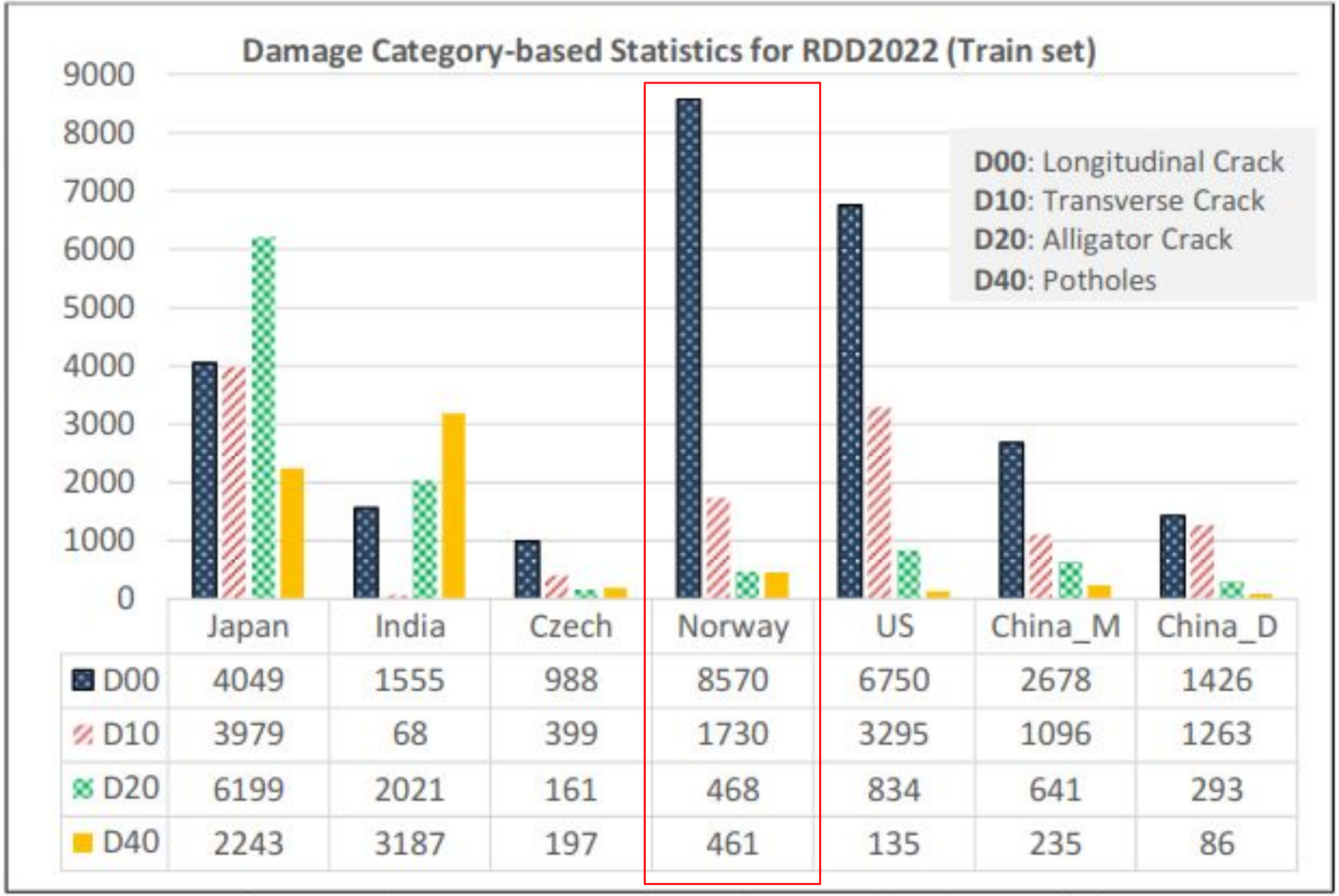
More robust towards overfitting



	Japan	India	Czech	Norway	US	China_M	China_D
D00	4049	1555	988	8570	6750	2678	1426
D10	3979	68	399	1730	3295	1096	1263
D20	6199	2021	161	468	834	641	293
D40	2243	3187	197	461	135	235	86

Figure 5: Damage Category-based data statistics for RDD2022

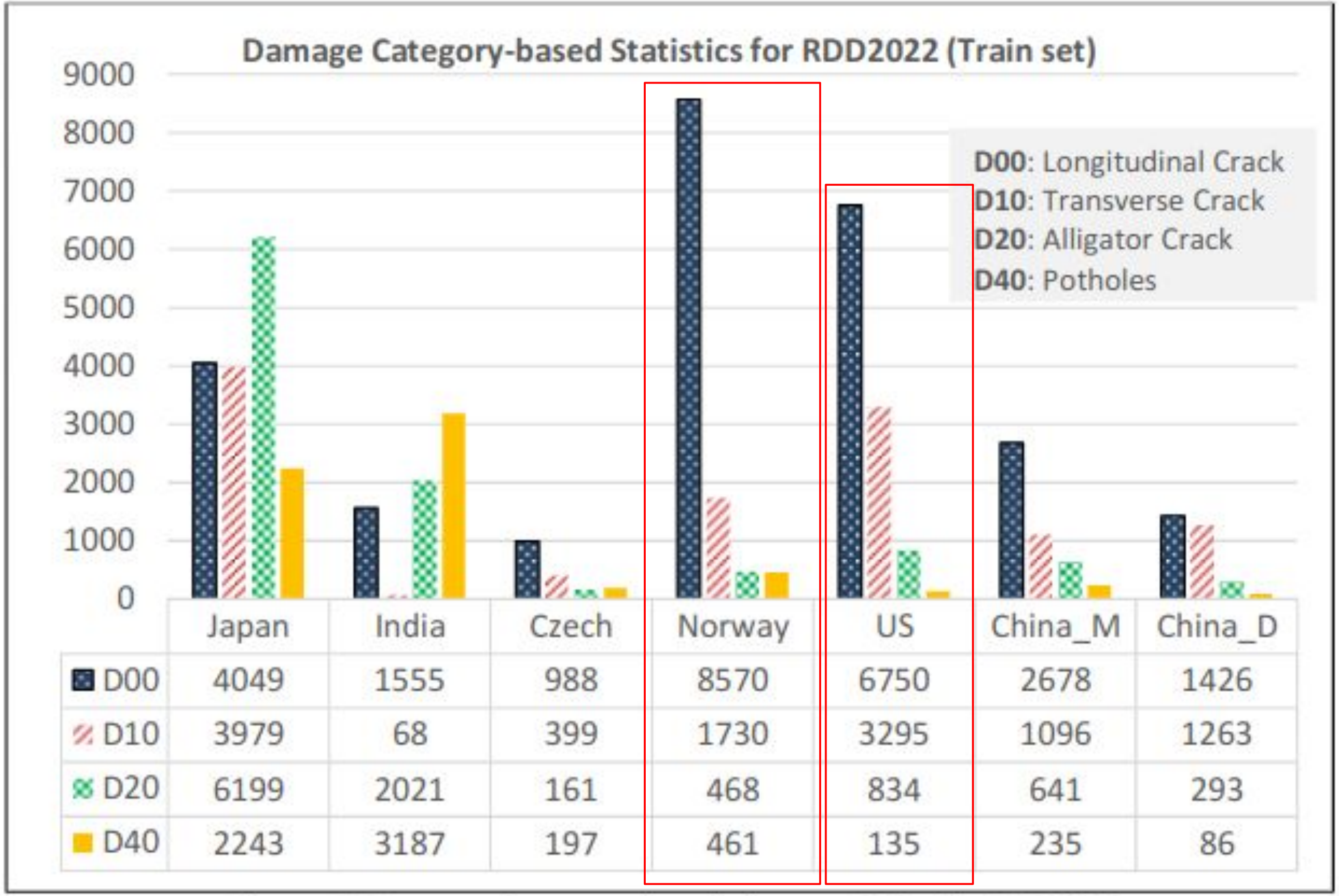
Figure collected from source [1]



	Japan	India	Czech	Norway	US	China_M	China_D
D00	4049	1555	988	8570	6750	2678	1426
D10	3979	68	399	1730	3295	1096	1263
D20	6199	2021	161	468	834	641	293
D40	2243	3187	197	461	135	235	86

Figure 5: Damage Category-based data statistics for RDD2022

Figure collected from source [1]



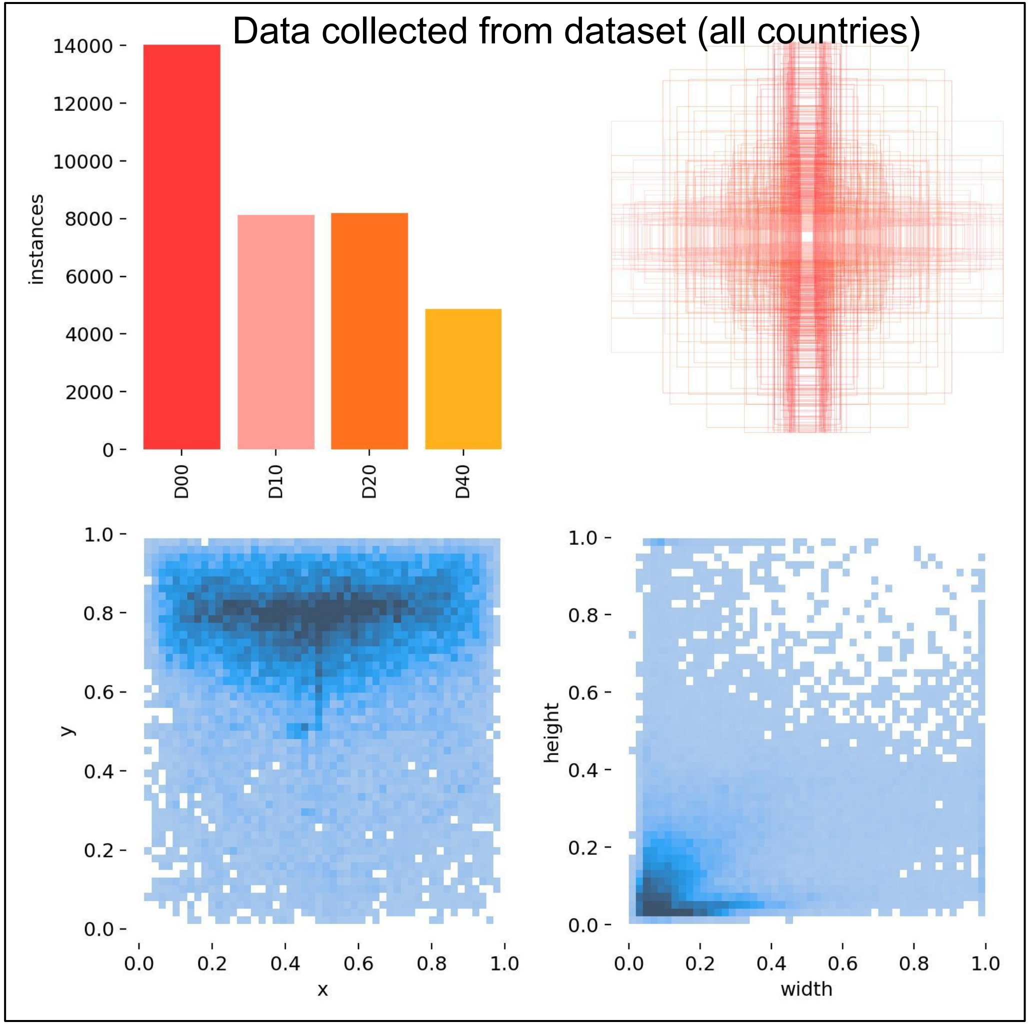
	Japan	India	Czech	Norway	US	China_M	China_D
D00	4049	1555	988	8570	6750	2678	1426
D10	3979	68	399	1730	3295	1096	1263
D20	6199	2021	161	468	834	641	293
D40	2243	3187	197	461	135	235	86

Figure 5: Damage Category-based data statistics for RDD2022

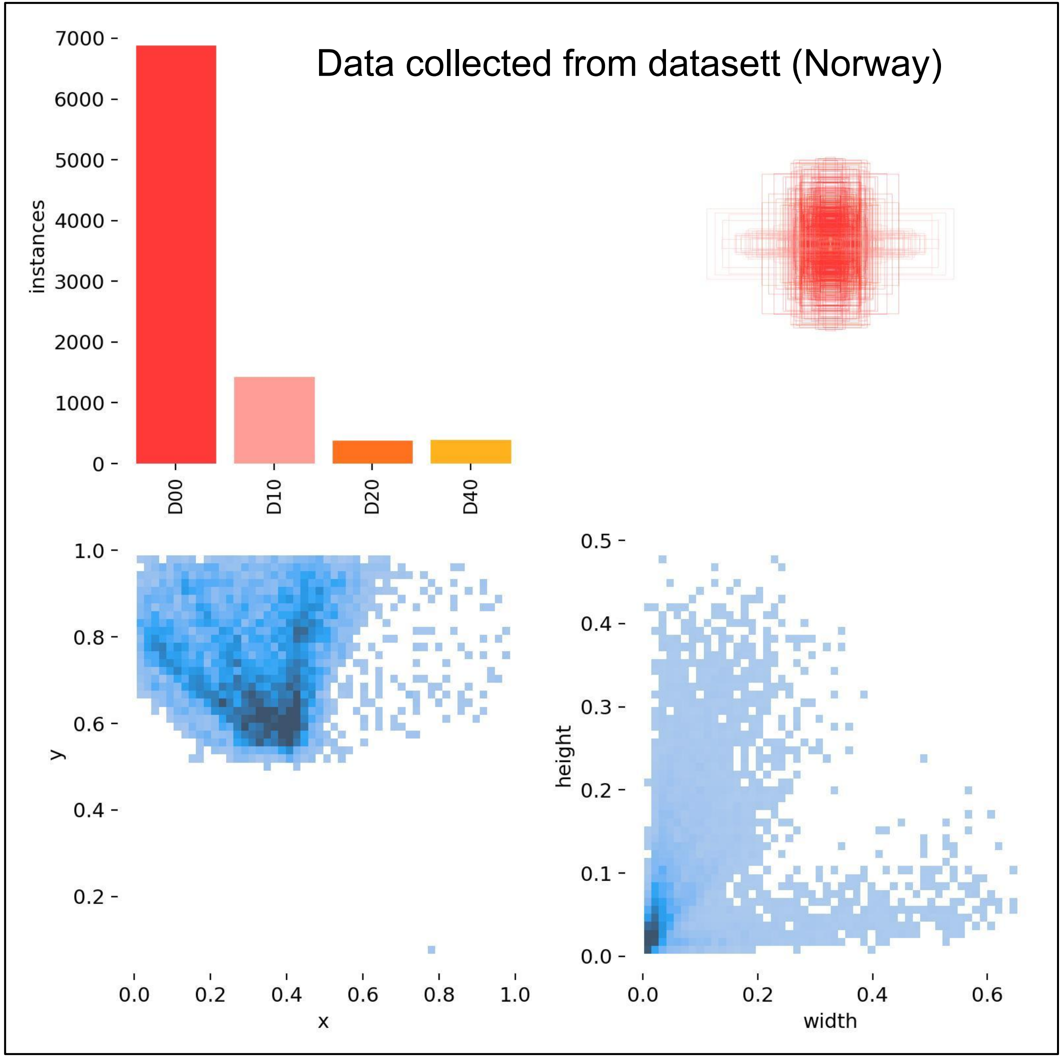
Figure collected from source [1]

Statistics

Norway
Usa
Czech
Republic
India
Japan
China

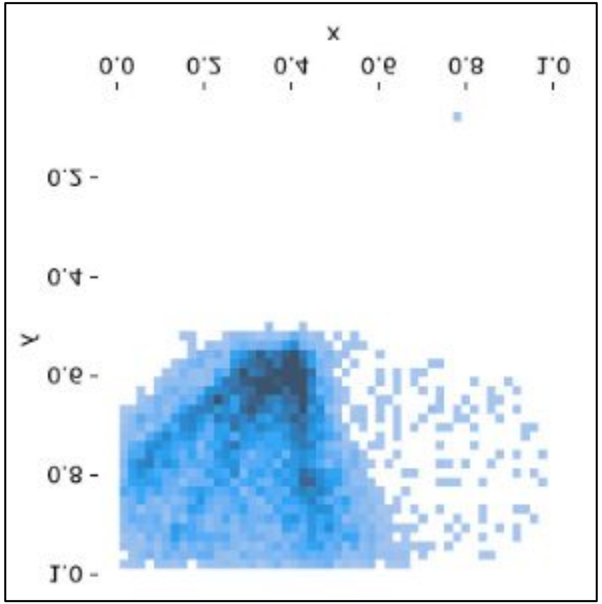


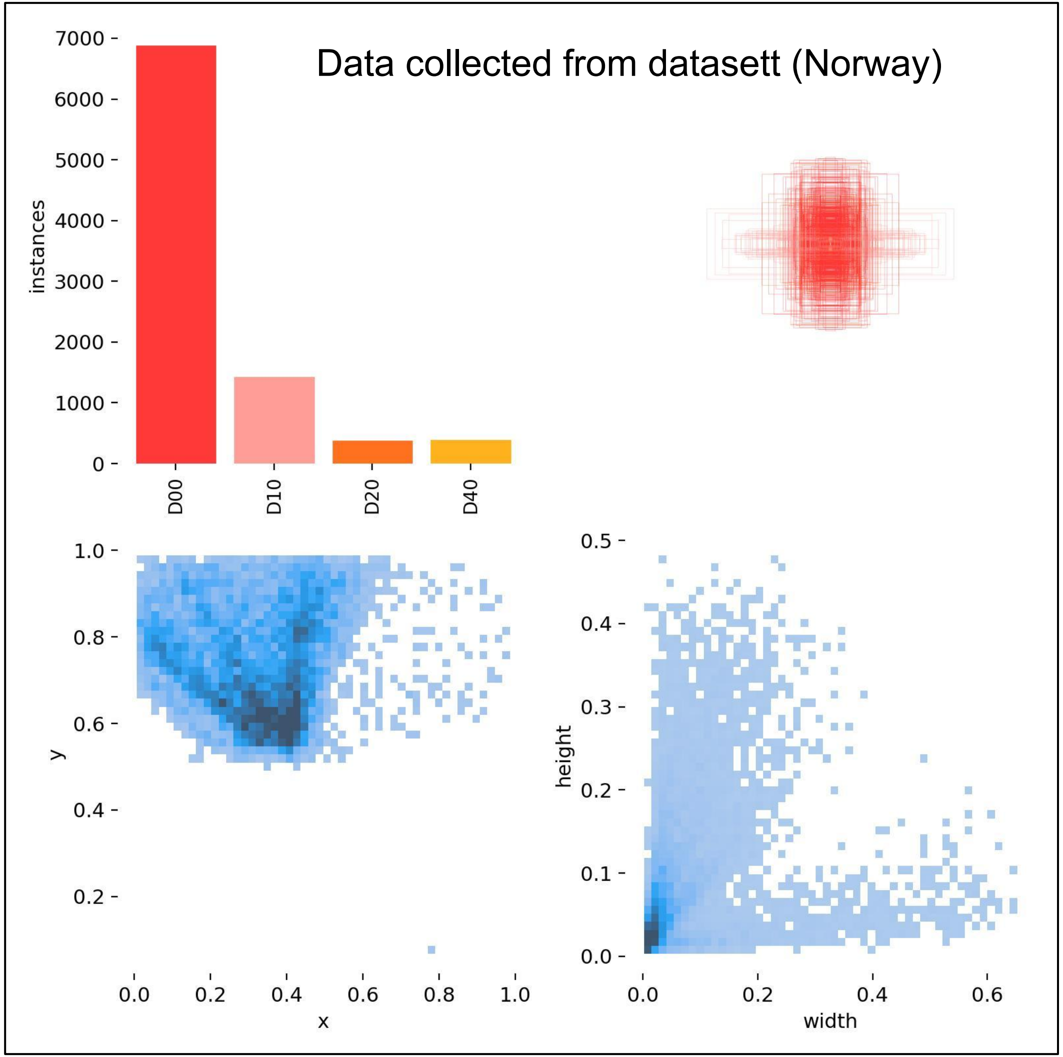
Statistics

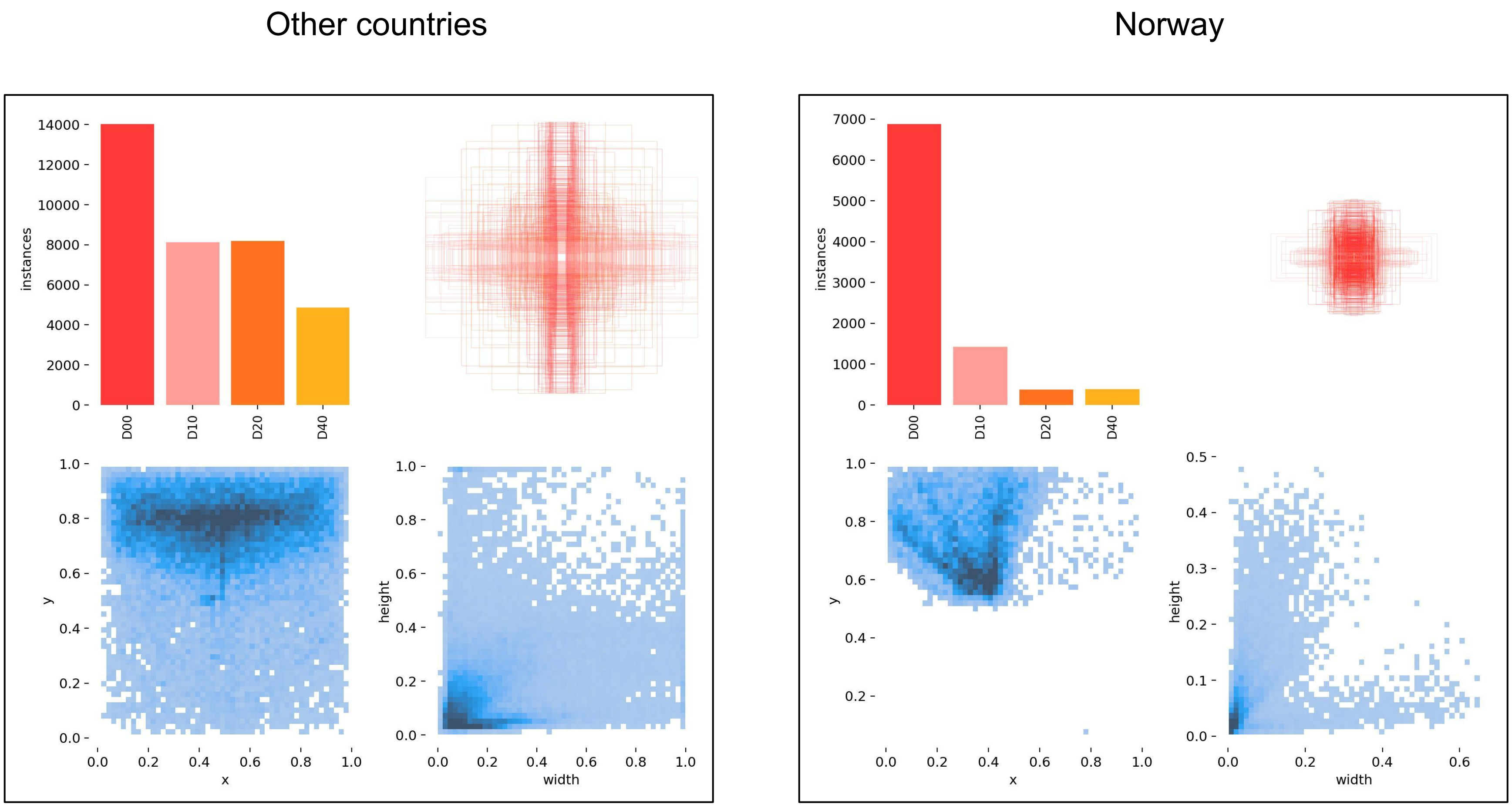

Data collected from datasett (Norway)
instances
7000
6000
5000
4000
3000
2000
1000
0
D00
D10
D20
D40
y
x
height
width
0.0
0.2
0.4
0.6
0.8
1.0
0.1
0.3
0.5

Statistics

Mirror image





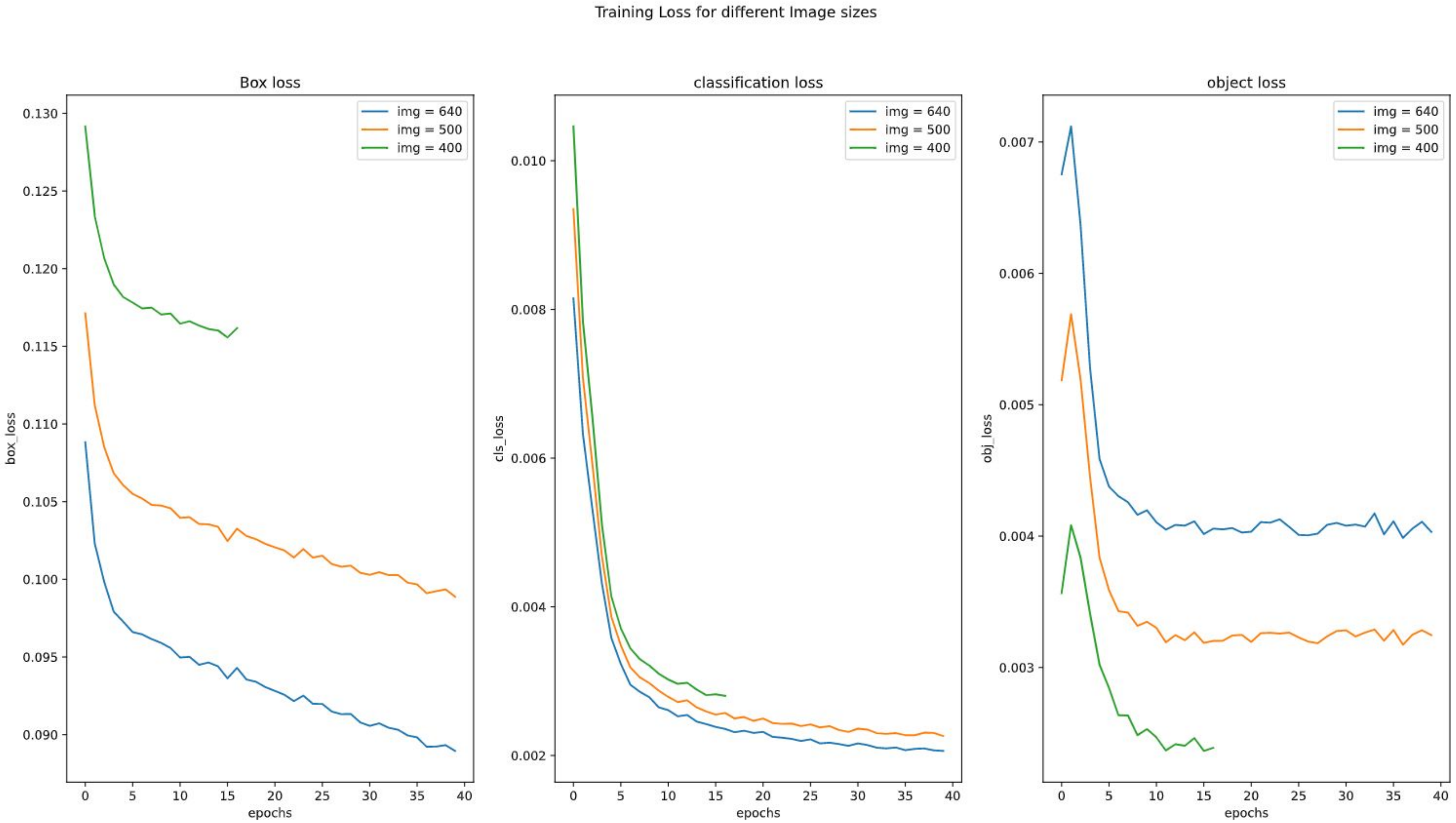


Other countries
Norway
instances
D00
D10
D20
D40
x
y
width
height

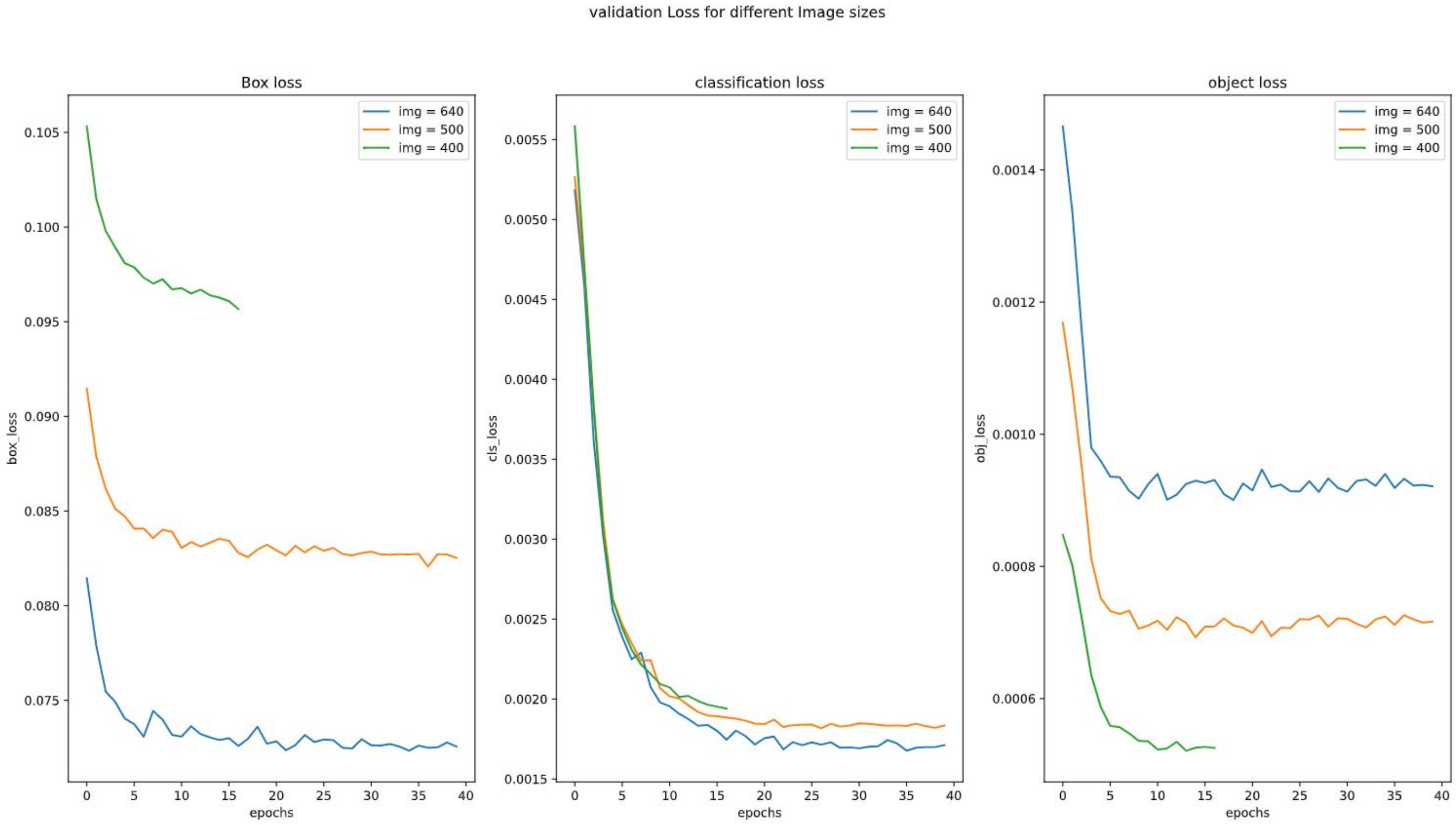
Resolution

	Norway	USA	Czech Republic	India	Japan	China
Resolution	3650x2044	640x640	600x600	720x720	600x600	512x512
Aquisition	Car	vehicle based (street view)	car	car	car	drones and bikes

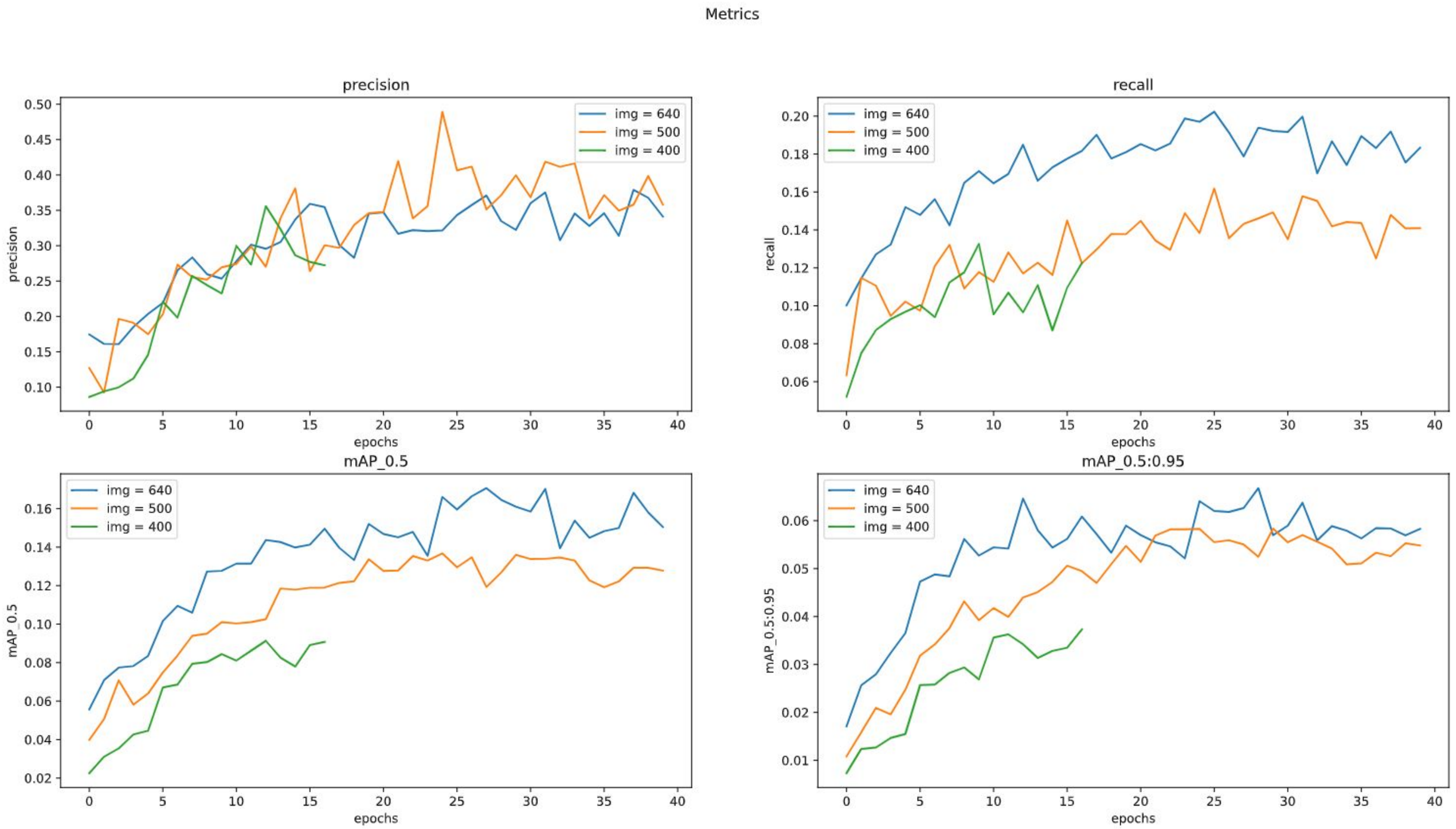
Data collected from source [1]



Training pretrained weights on data from Usa and China

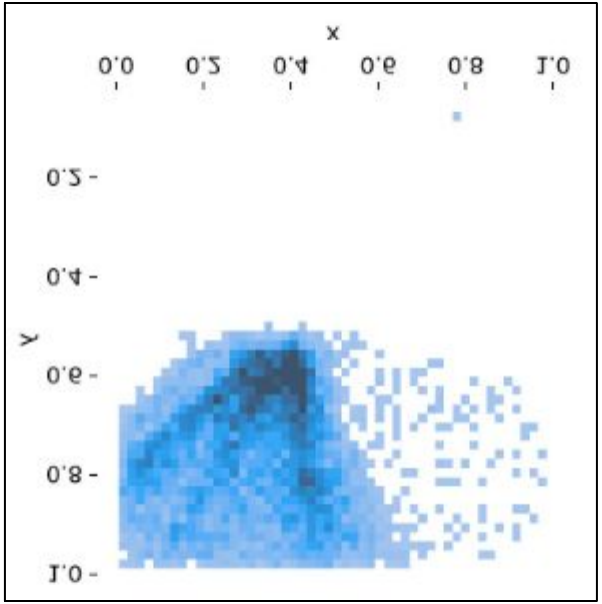


Training pretrained weights on data from Usa and China

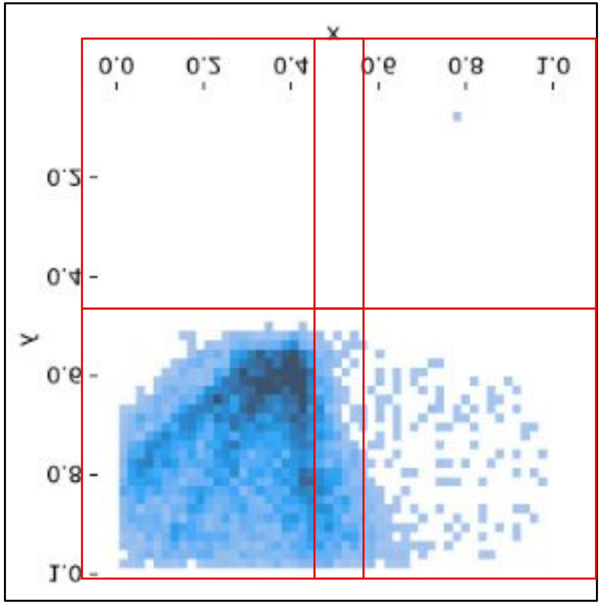


Training pretrained weights on data from Usa and China

Cropping - Increase the resolution.



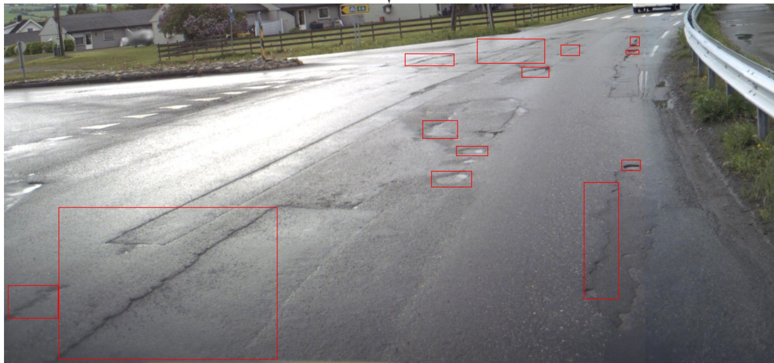
Cropping - Increase the resolution.



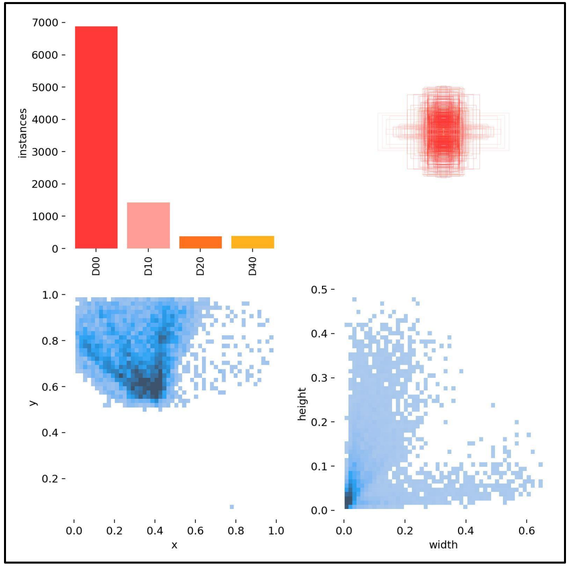
Dividing the photos into 4

	x_min_pixel / width	**x_max_pixel / width**	**y_min_pixel / height**	**y_max_pixel / height**
Lower right	0.3	1	0.6	1
Lower left	0	0.6	0.5	1
Upper left	0	0.6	0	0.6
Upper right	0.3	1	0.0	0.6

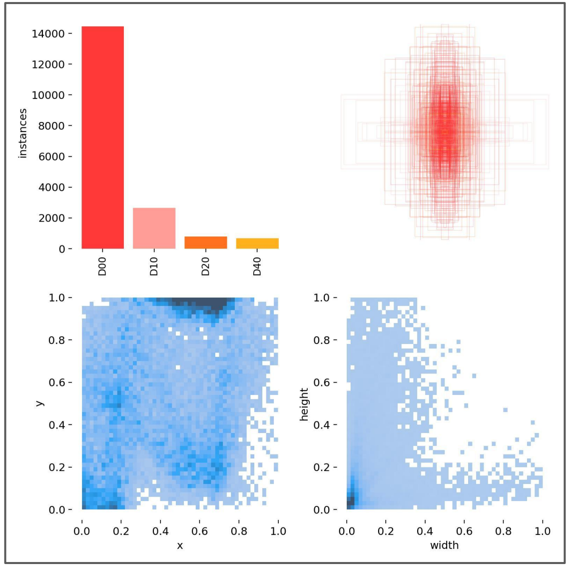
lower left



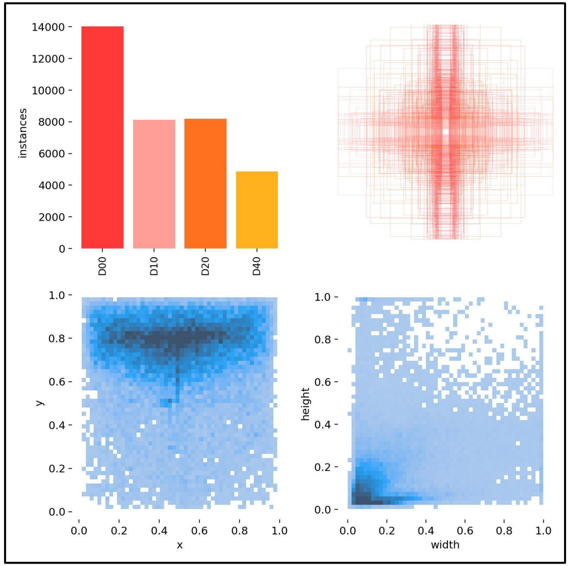
Label distribution after cropping



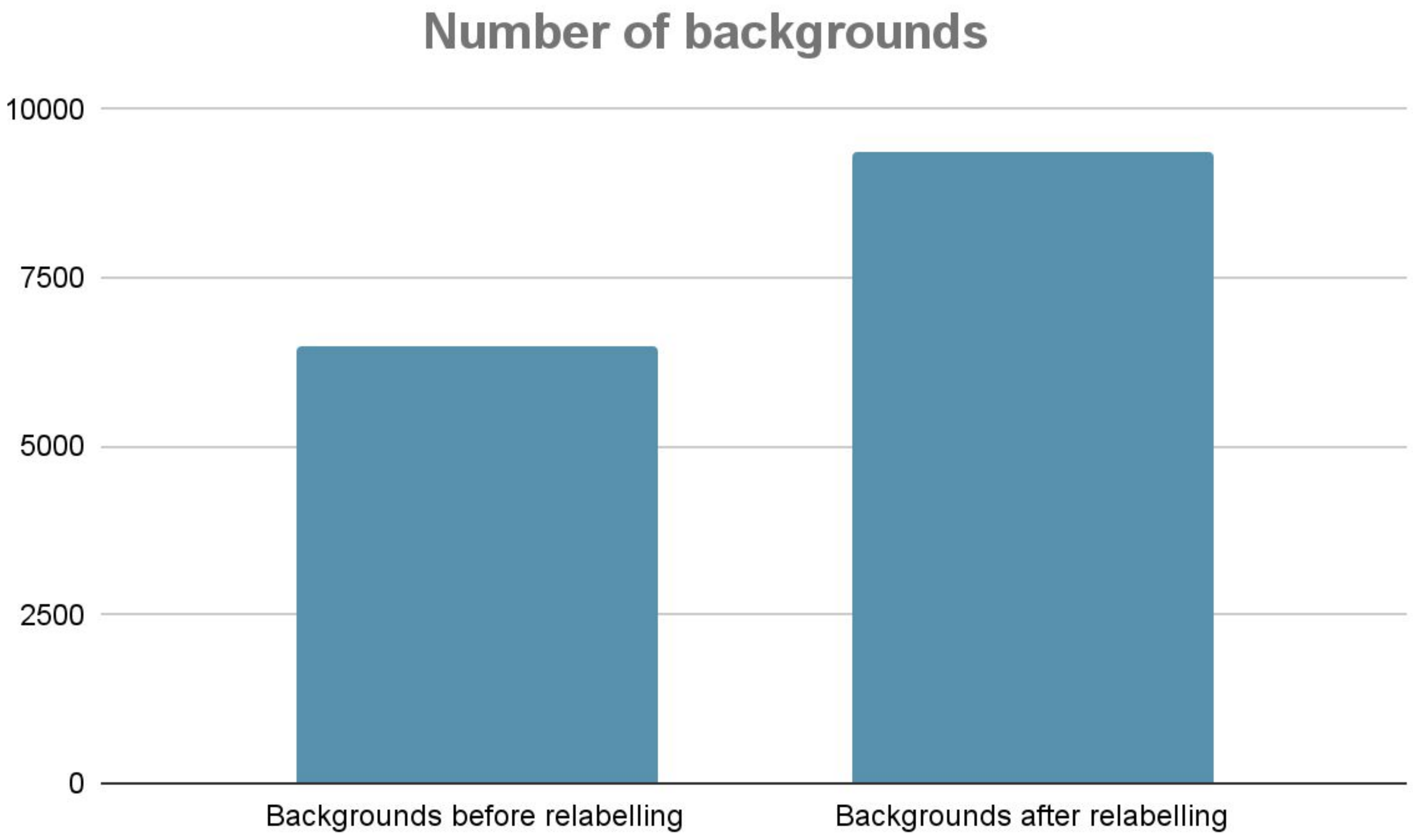
Before cropping (Norway)



After cropping (Norway)

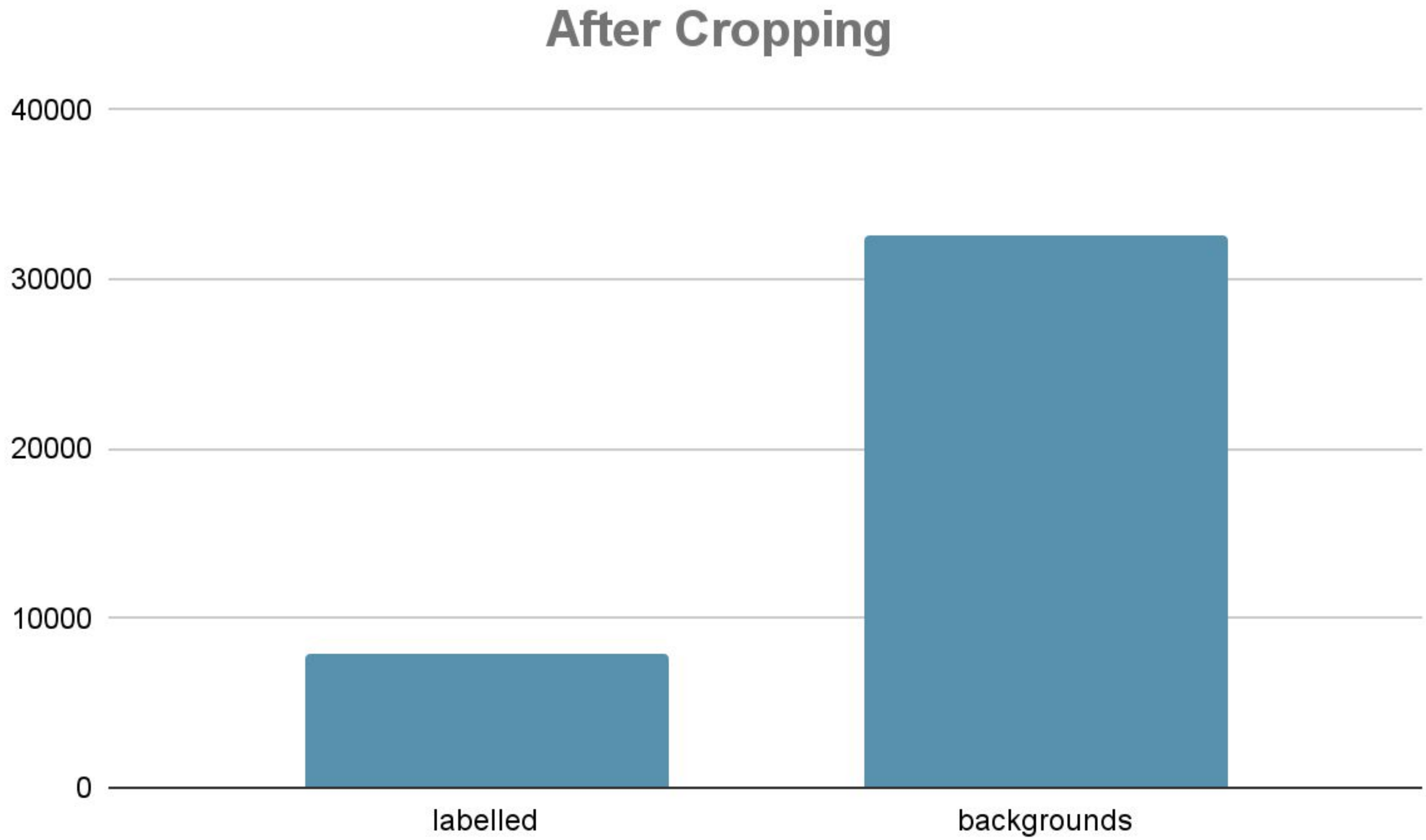


Merged dataset (all countries)



Data collected from datasett

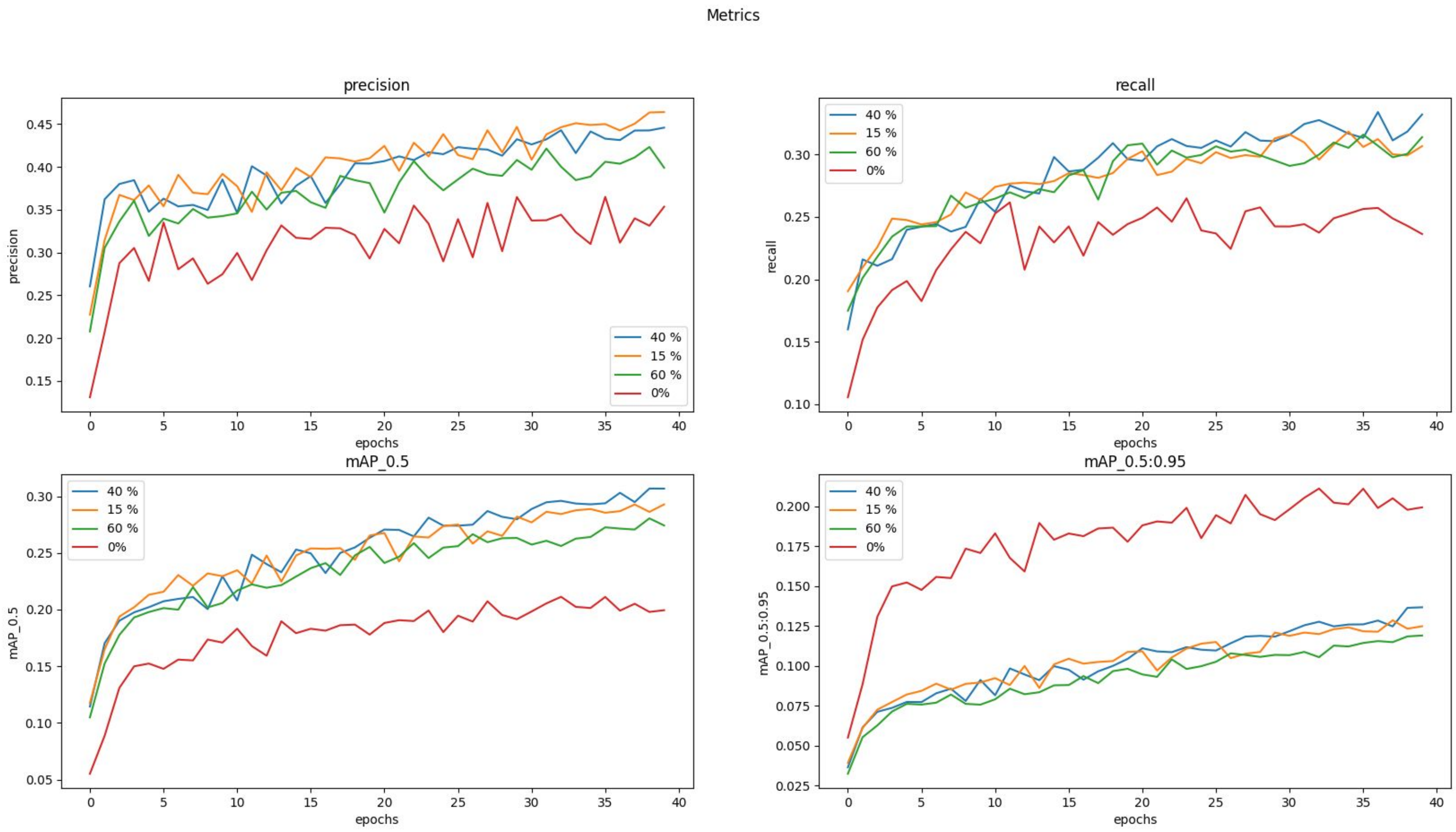
E


After Cropping
40000
30000
20000
10000
0
labelled
backgrounds

How many backgrounds should the dataset contain?

Background images are images with no objects that are added to a dataset to reduce False Positives (FP). We recommend about 0-10% background images to help reduce FPs (COCO has 1000 background images for reference, 1% of the total). No labels are required for background images. [3]

Testing:
– 0%
– 15%
– 40%
– 60%



Metrics
precision
recall
mAP_0.5
mAP_0.5:0.95
epochs
40 %
15 %
60 %
0%

Inference

0%	15%	40%	60%
0.079	0.098	0.091	0.080

Model Development

Yolov5

Fast

Easy to customise

Model	size	AP^{val}	AP^{test}	AP_{50}	$Speed_{V100}$	FPS_{V100}		params	GFLOPS
YOLOv5s	640	36.8	36.8	55.6	2.2ms	**455**		7.3M	17.0
YOLOv5m	640	44.5	44.5	63.1	2.9ms	345		21.4M	51.3
YOLOv5l	640	48.1	48.1	66.4	3.8ms	264		47.0M	115.4
YOLOv5x	640	**50.1**	**50.1**	**68.7**	6.0ms	167		87.7M	218.8
YOLOv5x + TTA	832	**51.9**	**51.9**	**69.6**	24.9ms	40		87.7M	1005.3

Model	size (pixels)	mAP^{val} 0.5:0.95	mAP^{test} 0.5:0.95	mAP^{val} 0.5	Speed V100 (ms)	params (M)	FLOPS 640 (B)
YOLOv5s6	1280	43.3	43.3	61.9	**4.3**	12.7	17.4
YOLOv5m6	1280	50.5	50.5	68.7	8.4	35.9	52.4
YOLOv5l6	1280	53.4	53.4	71.1	12.3	77.2	117.7
YOLOv5x6	1280	**54.4**	**54.4**	**72.0**	22.4	141.8	222.9
YOLOv5x6 TTA	1280	**55.0**	**55.0**	**72.0**	70.8	-	-

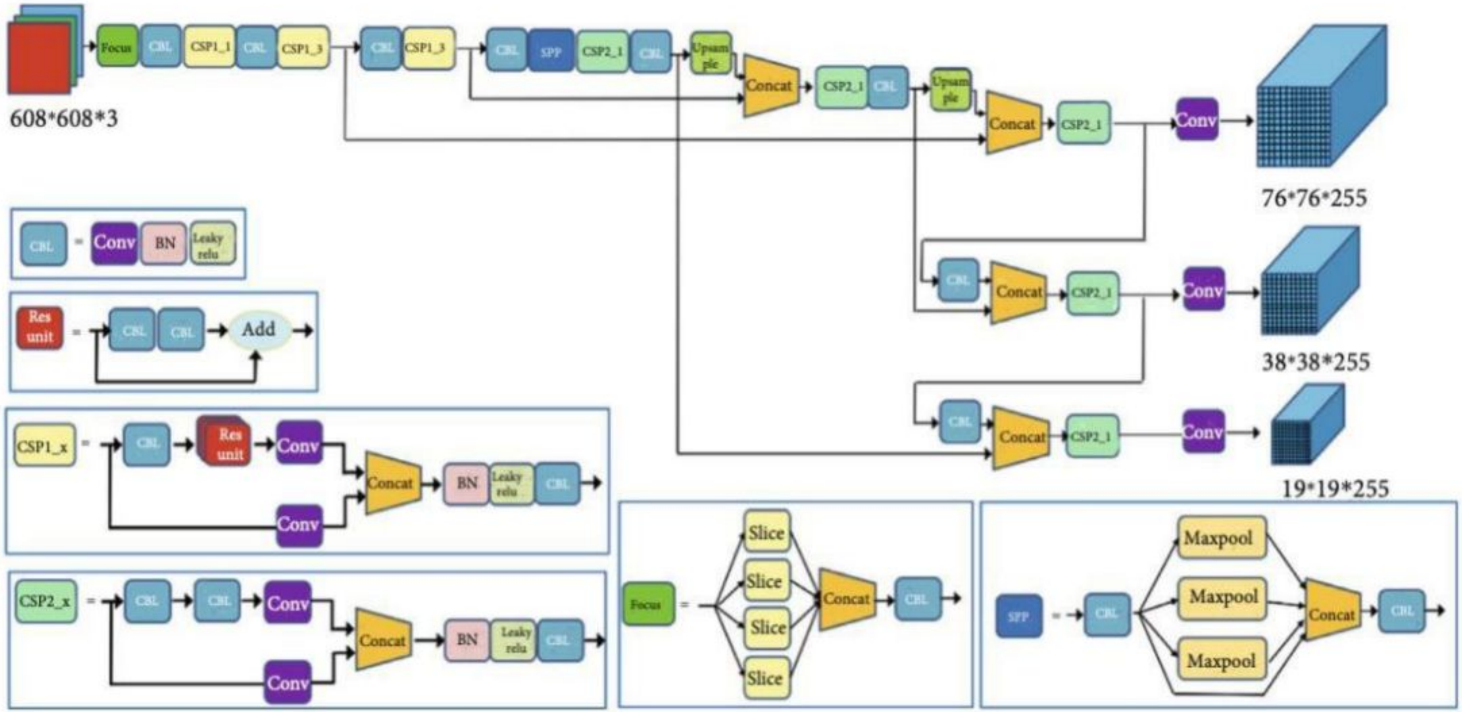


Figure 13. YOLOv5s6 architecture [43].

Source [7]

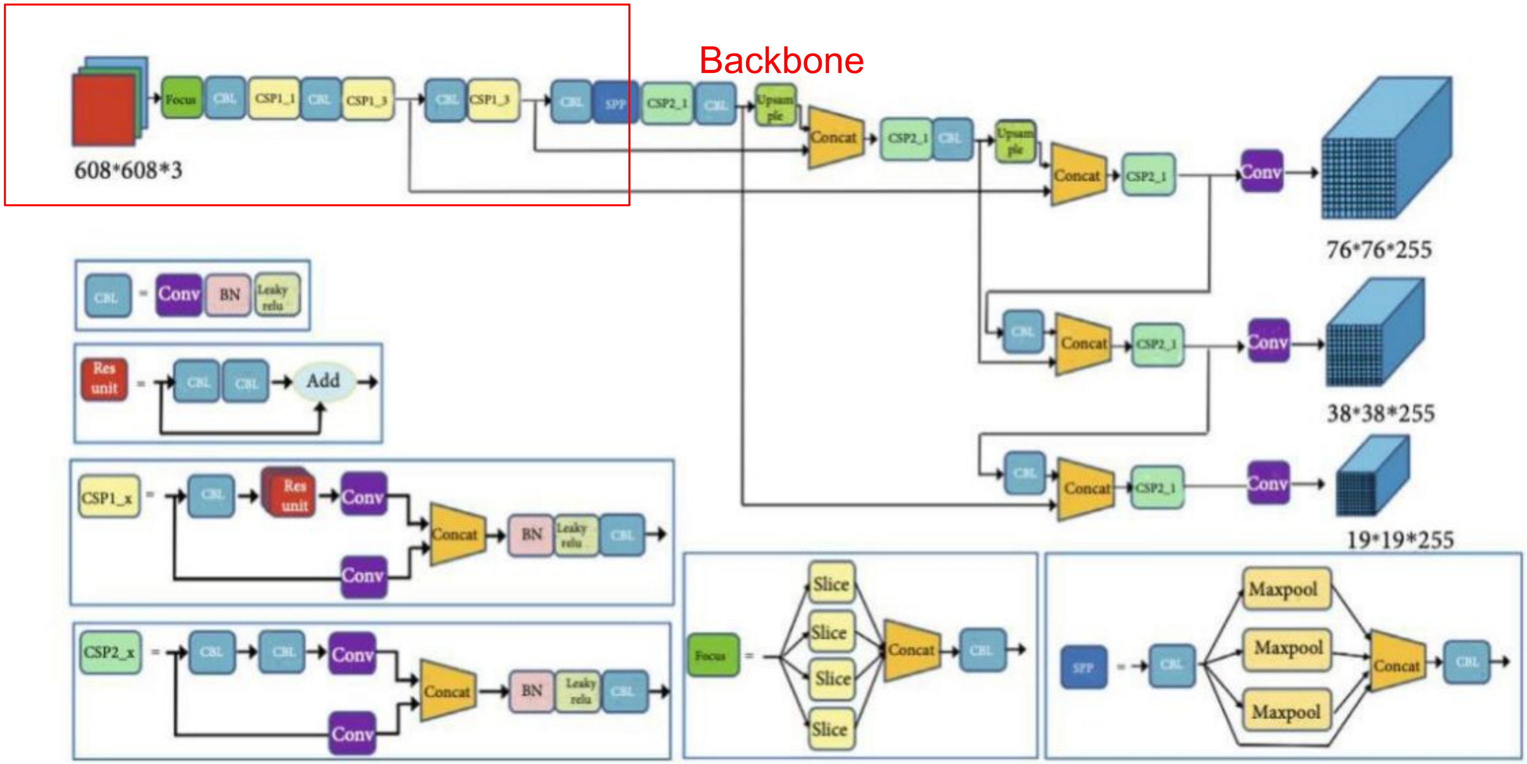


Figure 13. YOLOv5s6 architecture [43].

Source [7]

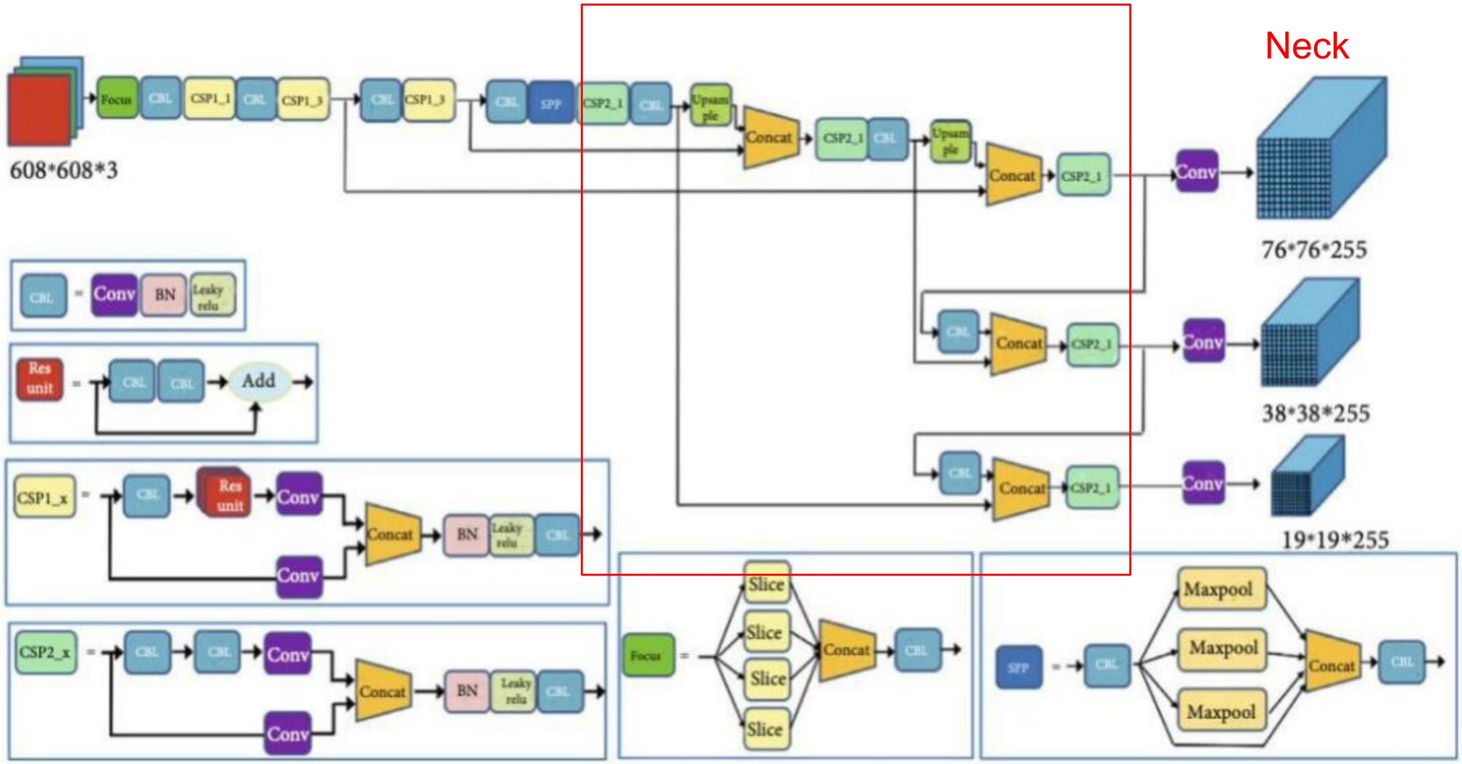


Figure 13. YOLOv5s6 architecture [43].

Source [7]

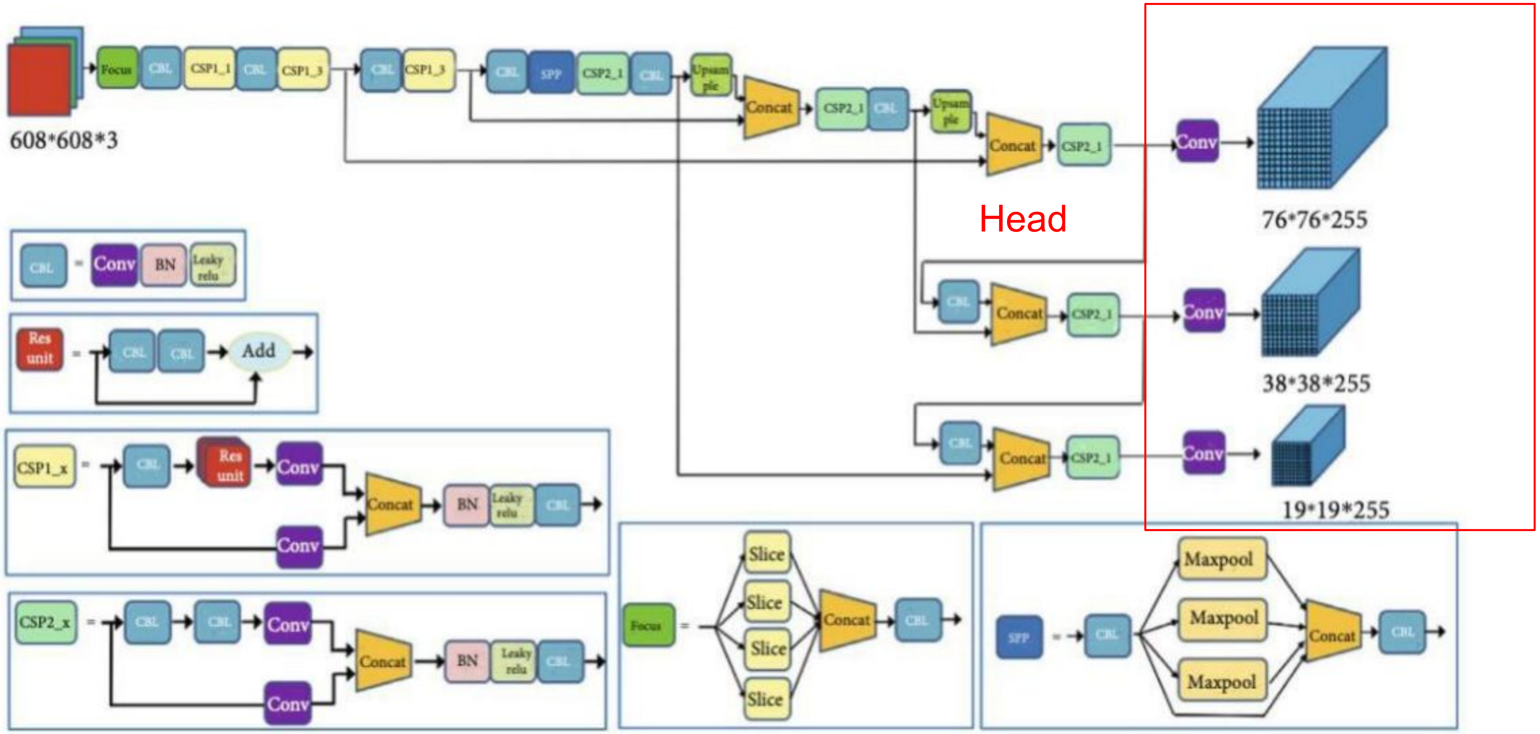


Figure 13. YOLOv5s6 architecture [43].

Source [7]

Pretraining COCO weights on Merged dataset

Epochs: 50

Configuration: yolov5s6.yaml

Weights: yolov5s6.pt

batch_size: 8

img: 1280

cache

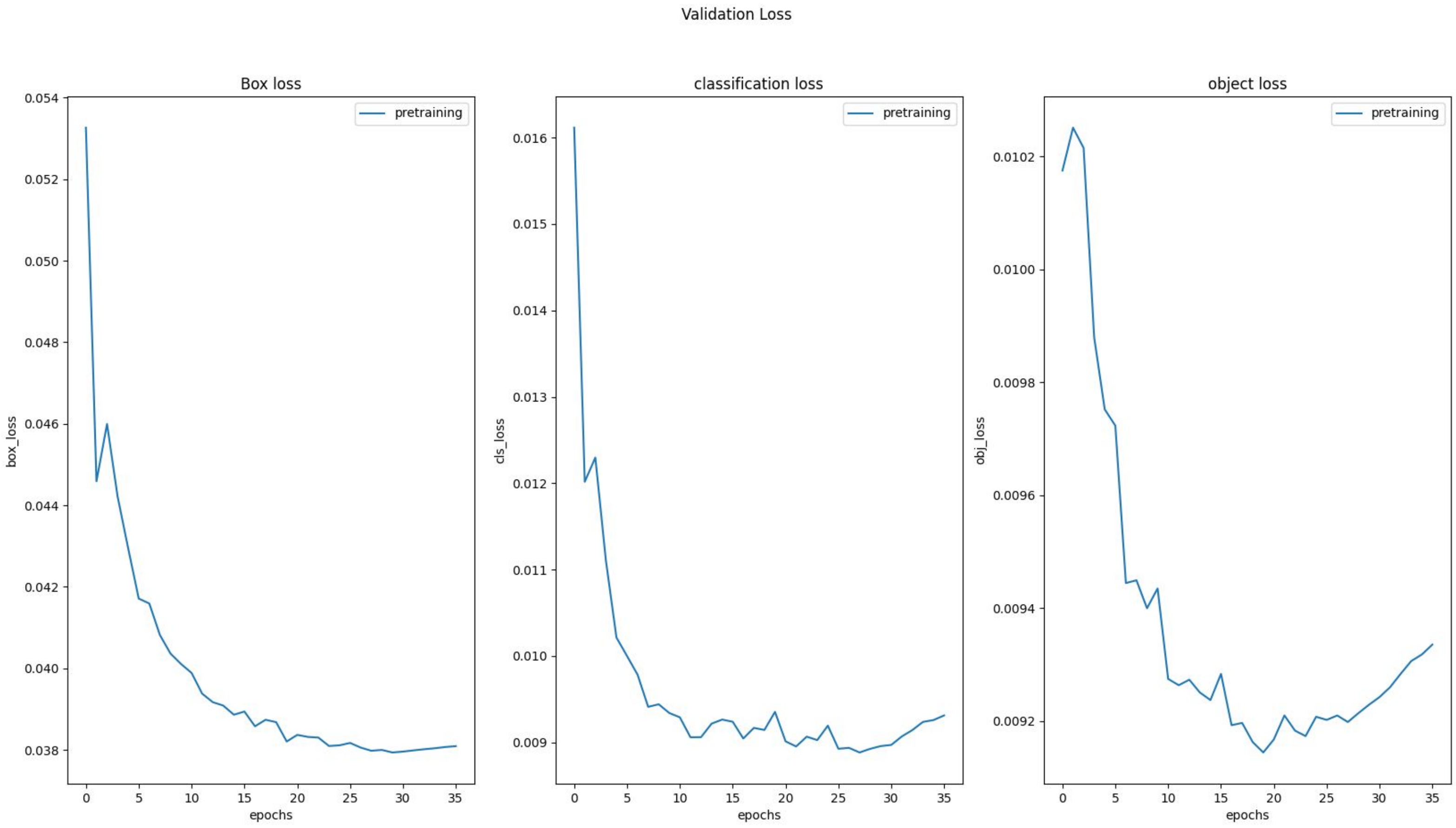
image_weights (imbalanced)

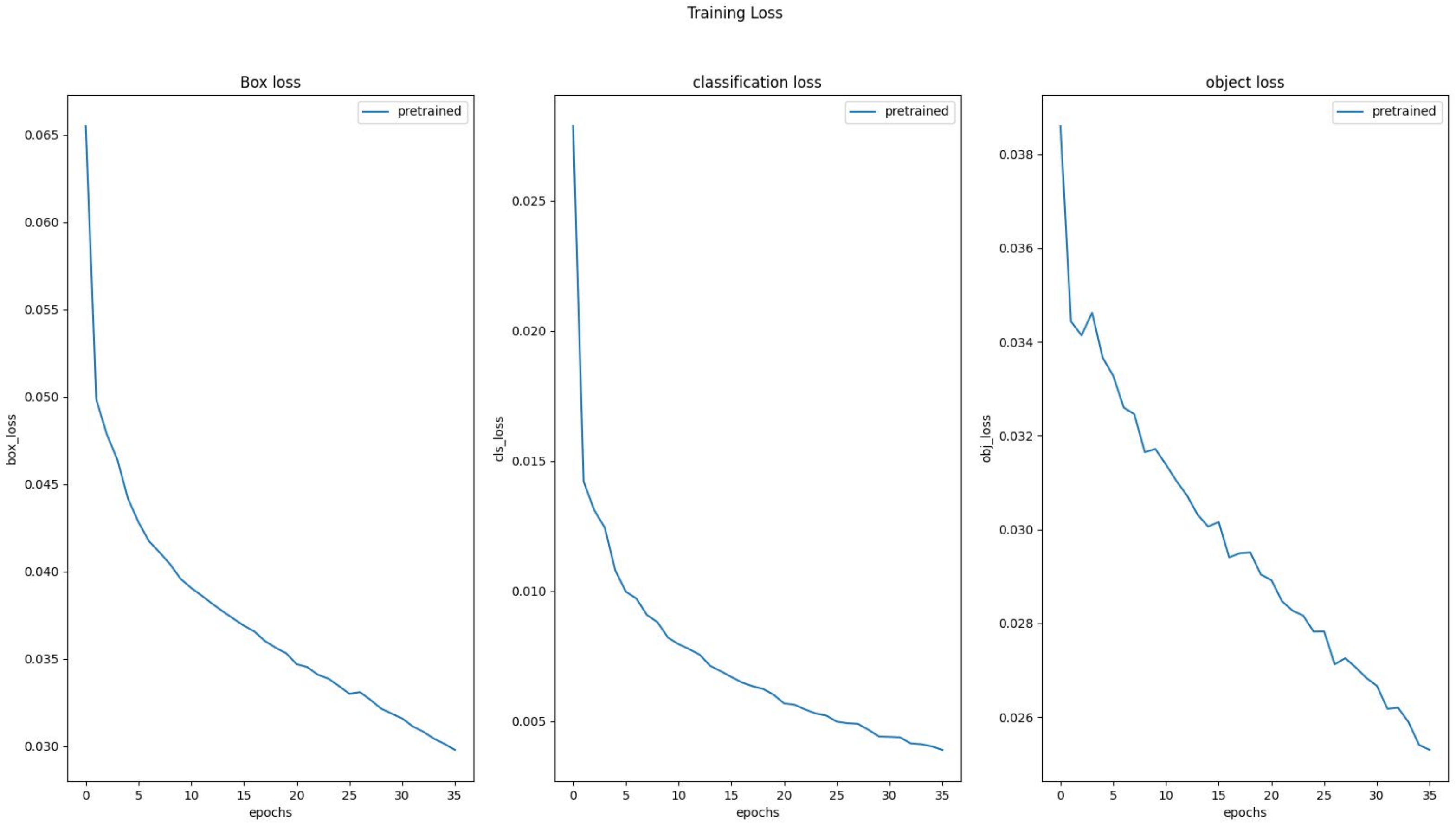
Runtime: 600 min

hyp: high augmentation yaml

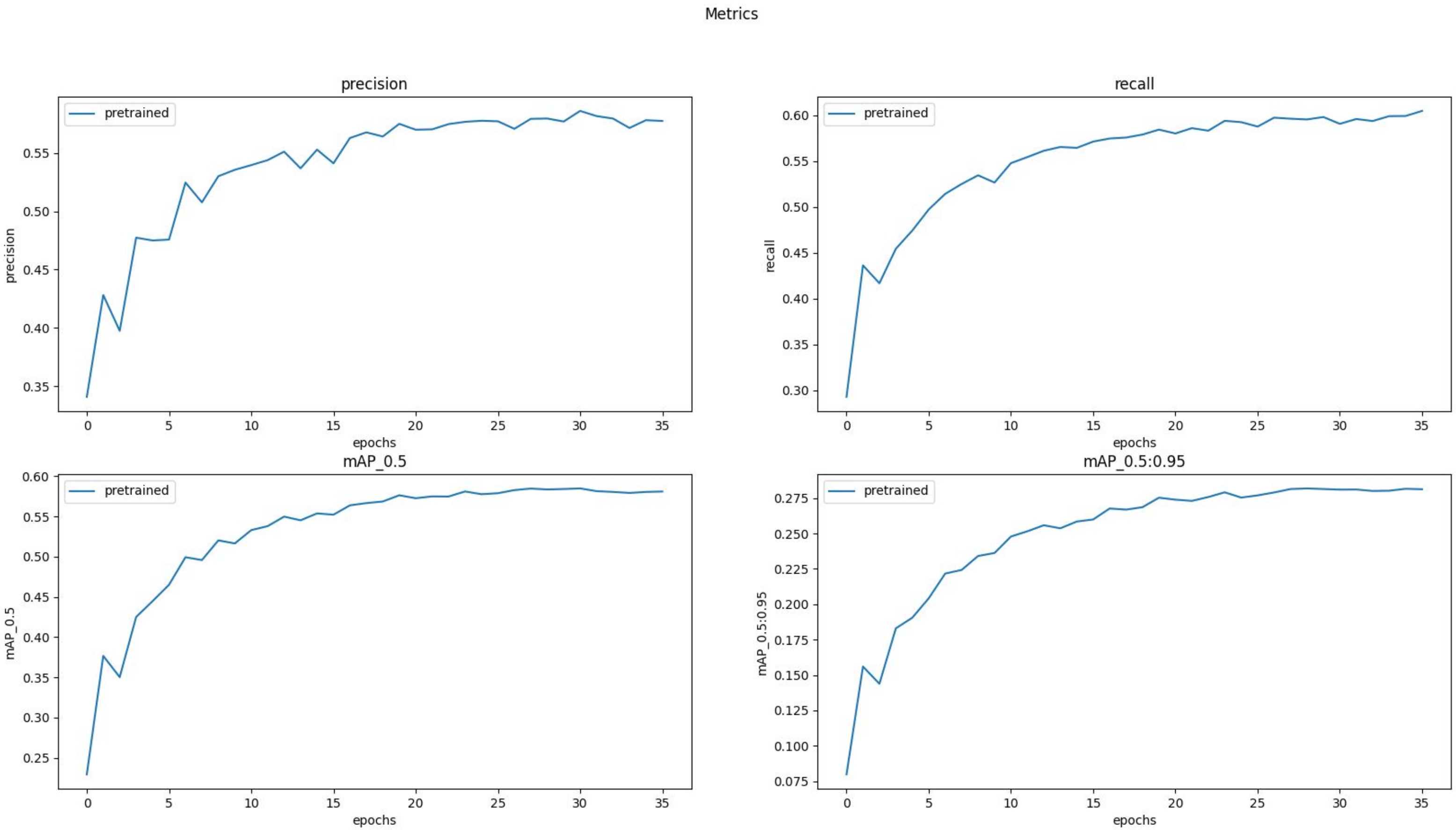
Hyper parameters

Problem	**Solution**	**Parameter**
Different backgrounds	Prevent overfitting on backgrounds	Illumination Augmentations copy paste perspective fliplr
Small object	Feature extractor to be good at extracting small features	Increase box loss scale augmentation
Imbalanced	Increase classification of minority classes	mosaic [6] fl_gamma


Validation Loss
Box loss
pretraining
box_loss
epochs
classification loss
cls_loss
object loss
obj_loss



Training Loss
Box loss
classification loss
object loss
pretrained
pretrained
pretrained
box_loss
cls_loss
obj_loss
epochs
epochs
epochs


Metrics
precision
pretrained
0.55
0.50
0.45
0.40
0.35
precision
0 5 10 15 20 25 30 35
epochs
recall
pretrained
0.60
0.55
0.50
0.45
0.40
0.35
0.30
recall
0 5 10 15 20 25 30 35
epochs
mAP_0.5
pretrained
0.60
0.55
0.50
0.45
0.40
0.35
0.30
0.25
mAP_0.5
0 5 10 15 20 25 30 35
epochs
mAP_0.5:0.95
pretrained
0.275
0.250
0.225
0.200
0.175
0.150
0.125
0.100
0.075
mAP_0.5:0.95
0 5 10 15 20 25 30 35
epochs

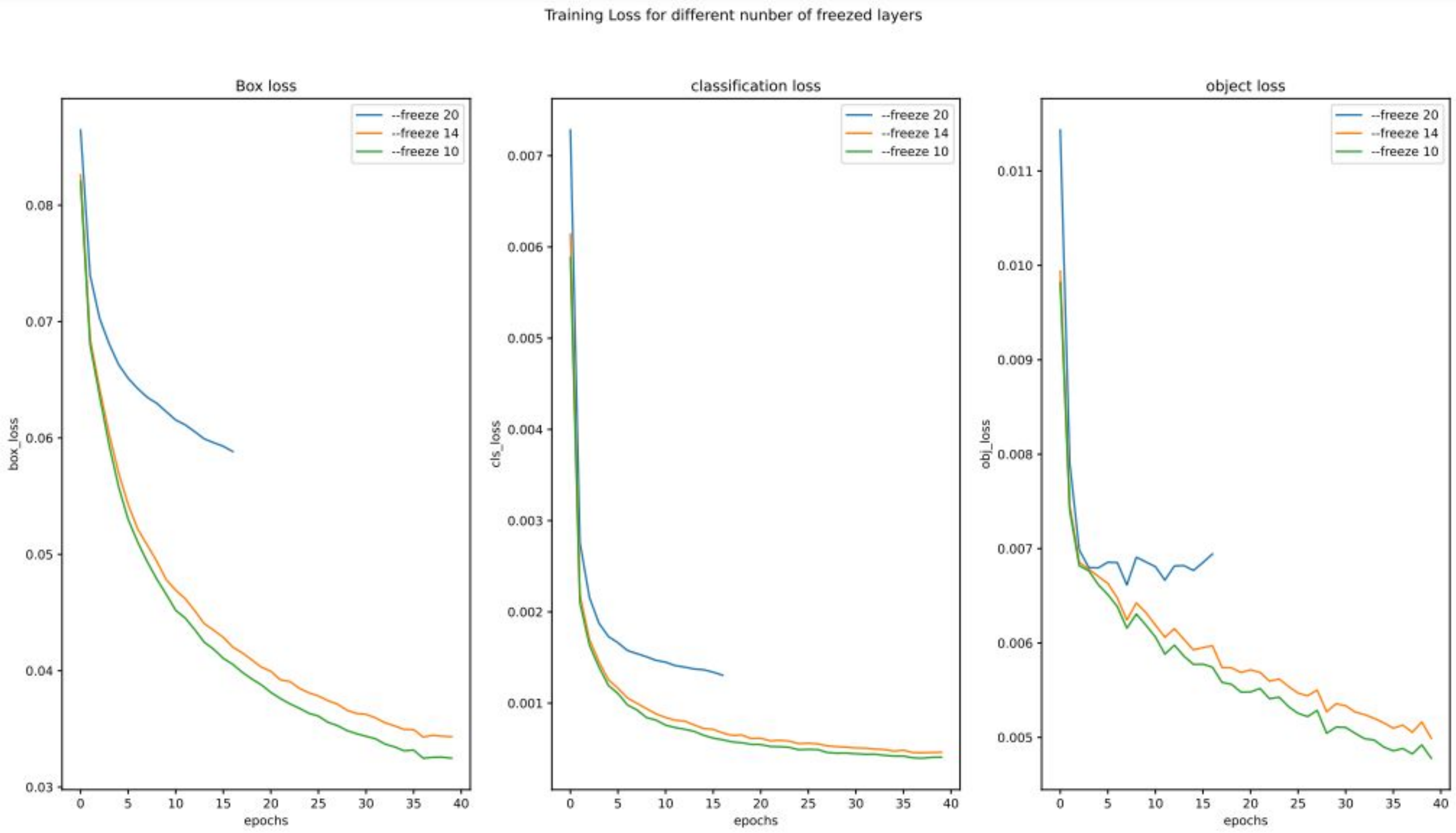
Finetuning to the Norwegian dataset

How many layers to freeze?

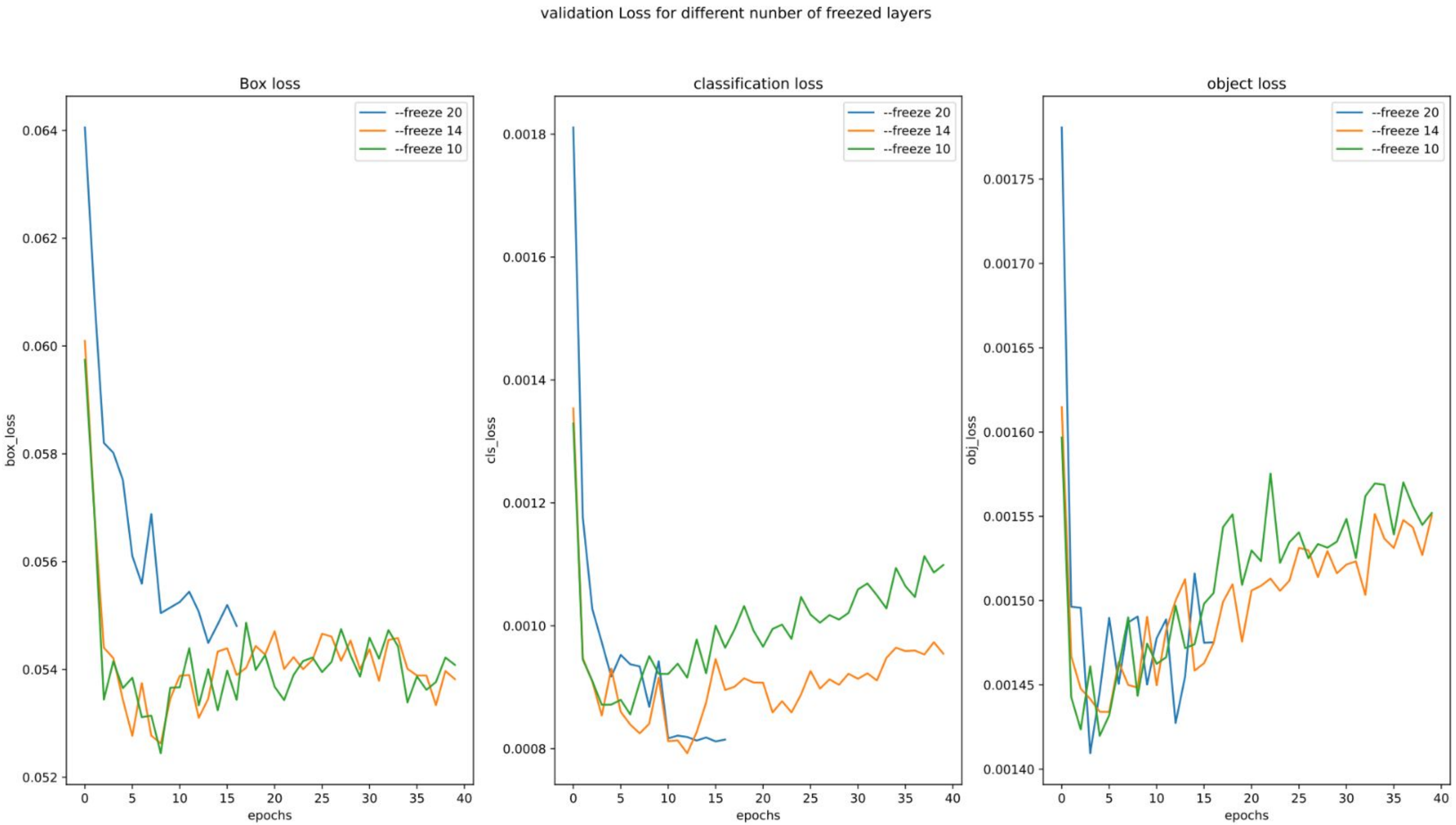
– 10

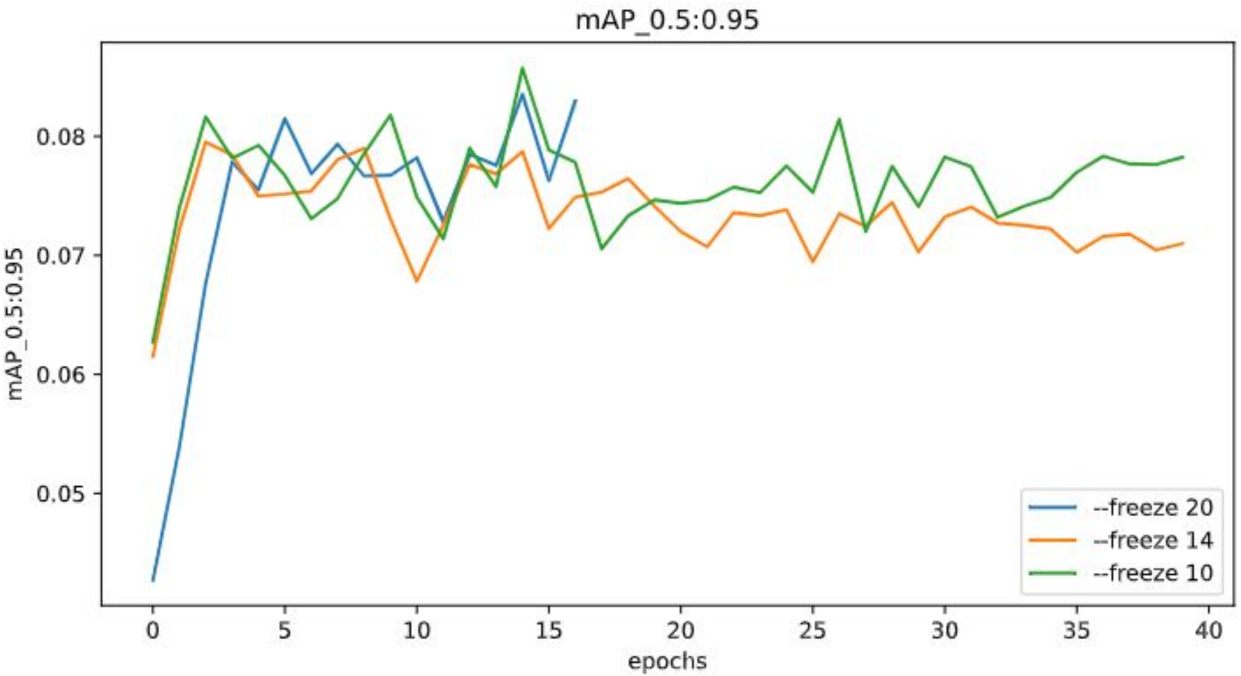
– 14

– 20



Training Loss for different nunber of freezed layers
Box loss
classification loss
object loss
--freeze 20
--freeze 14
--freeze 10
box_loss
cls_loss
obj_loss
epochs

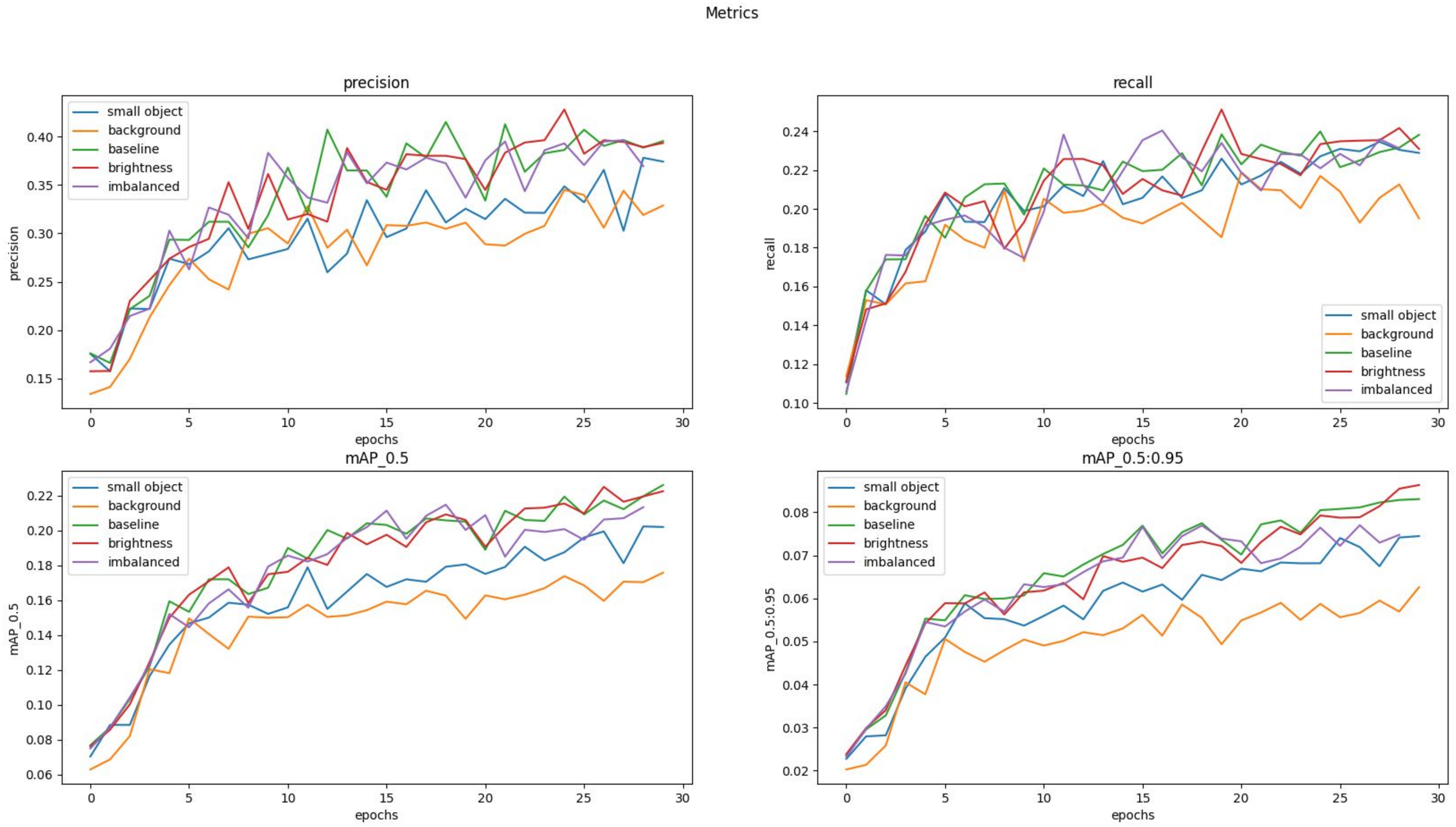

validation Loss for different nunber of freezed layers
Box loss
classification loss
object loss
--freeze 20
--freeze 14
--freeze 10
box_loss
cls_loss
obj_loss
epochs
0.064
0.062
0.060
0.058
0.056
0.054
0.052
0.0018
0.0016
0.0014
0.0012
0.0010
0.0008
0.00175
0.00170
0.00165
0.00160
0.00155
0.00150
0.00145
0.00140
0
5
10
15
20
25
30
35
40


mAP_0.5:0.95
mAP_0.5:0.95
0.08
0.07
0.06
0.05
0
5
10
15
20
25
30
35
40
epochs
--freeze 20
--freeze 14
--freeze 10

Tuning hyperparameters

Efficient Tuning - Clustering similar parameters

Imbalance	Brightness	Background	Small Object
cls_pw (up)	Hue	Copy paste	scale
cls (down)	Saturation	perspective	obj (up)
mosaic		fliplr	
fl gamma			


Metrics
precision
recall
mAP_0.5
mAP_0.5:0.95
small object
background
baseline
brightness
imbalanced
epochs

Inference

Baseline	Imbalance	Brightness	Background	Small objects
0.082	0.078	0.08	0.06	0.082

Inference

detect + augment

Final model

Pretraining

Model: yolov5s6

Weights: Pretrained on the Coco-dataset, yolov5s6.pt

batch_size: 8

img: 1280

Augmentations:

Runtime: 600 min

Dataset: All countries relabelled.

Finetuning

Model: yolov5s6

Weights: Pretrained on all countries

batch_size: 8

img: 1280

Augmentations:

Runtime: 600 min

Dataset: Norway, cropped, 15% backgrounds

Inference

Model: yolov5s6

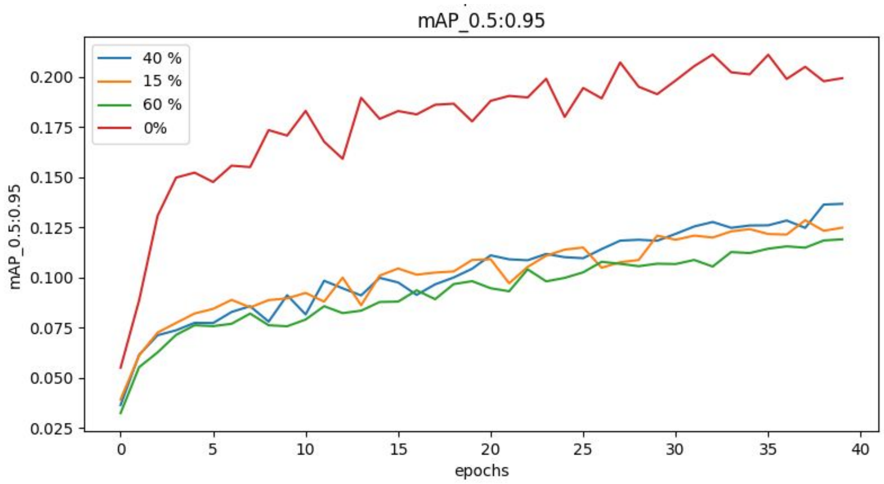
Weights: Finetuned on Norway

Test Time Augmentations

Accuracy: 0.098

Results

Best result was Map 950:95 = 0.098 training on data with 15% backgrounds.

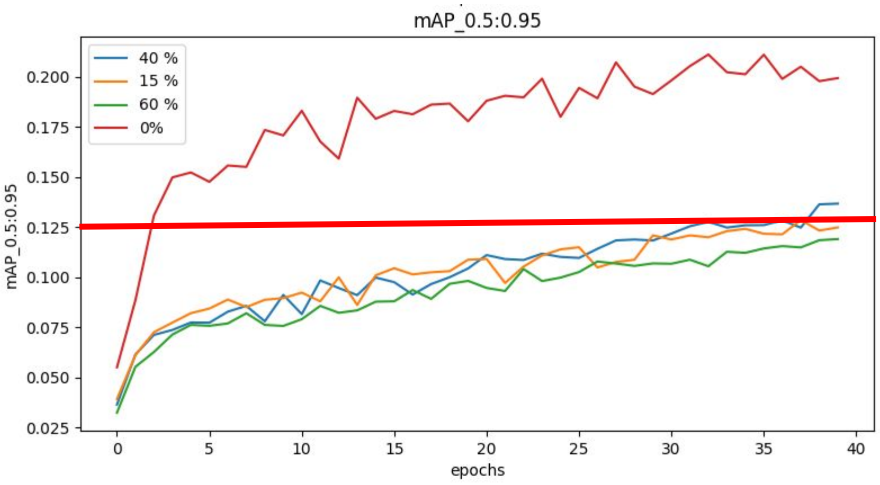


Results

Best result was Map 950:95 = 0.098 training on data with 15% backgrounds.

A reduction of over 2% in accuracy on test data.

Overfitting



Runtime Analysis

Used smaller resolution when finetuning.

Fast detection with yolo detect.py -nosave

Pretraining (all countries)	Finetuning (Norway)	Inference
600 min	300 min	4 min

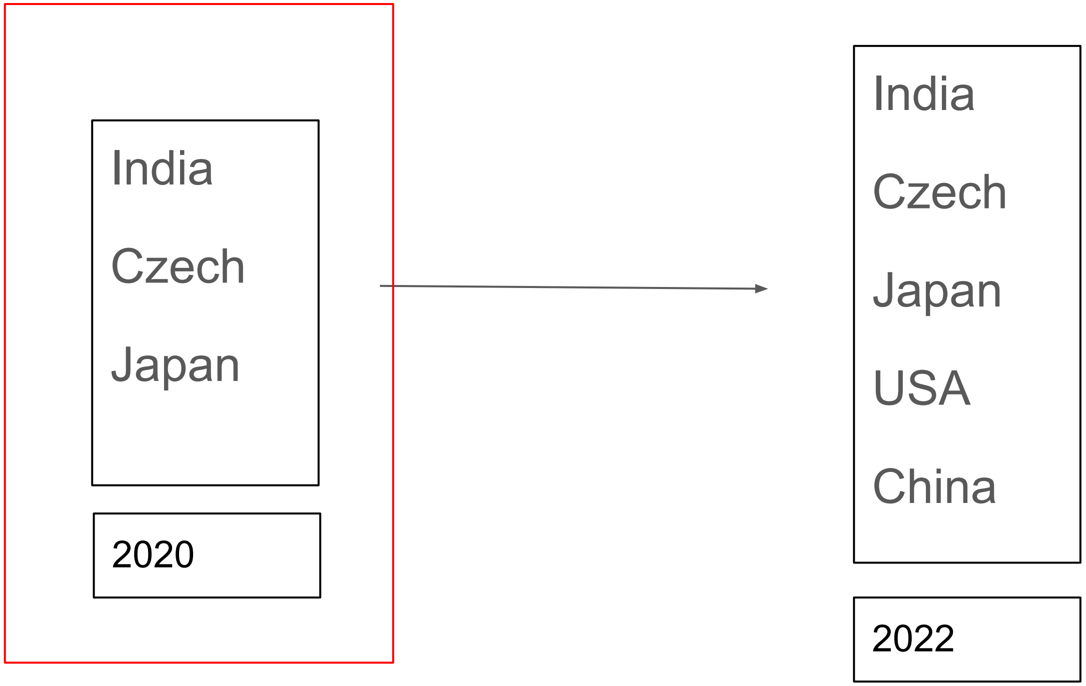
180 W * 18,7H = 3366 kWh

In Norway: 30g CO2eq/kWh

Carbon Footprint: 81,4kg

Discussion

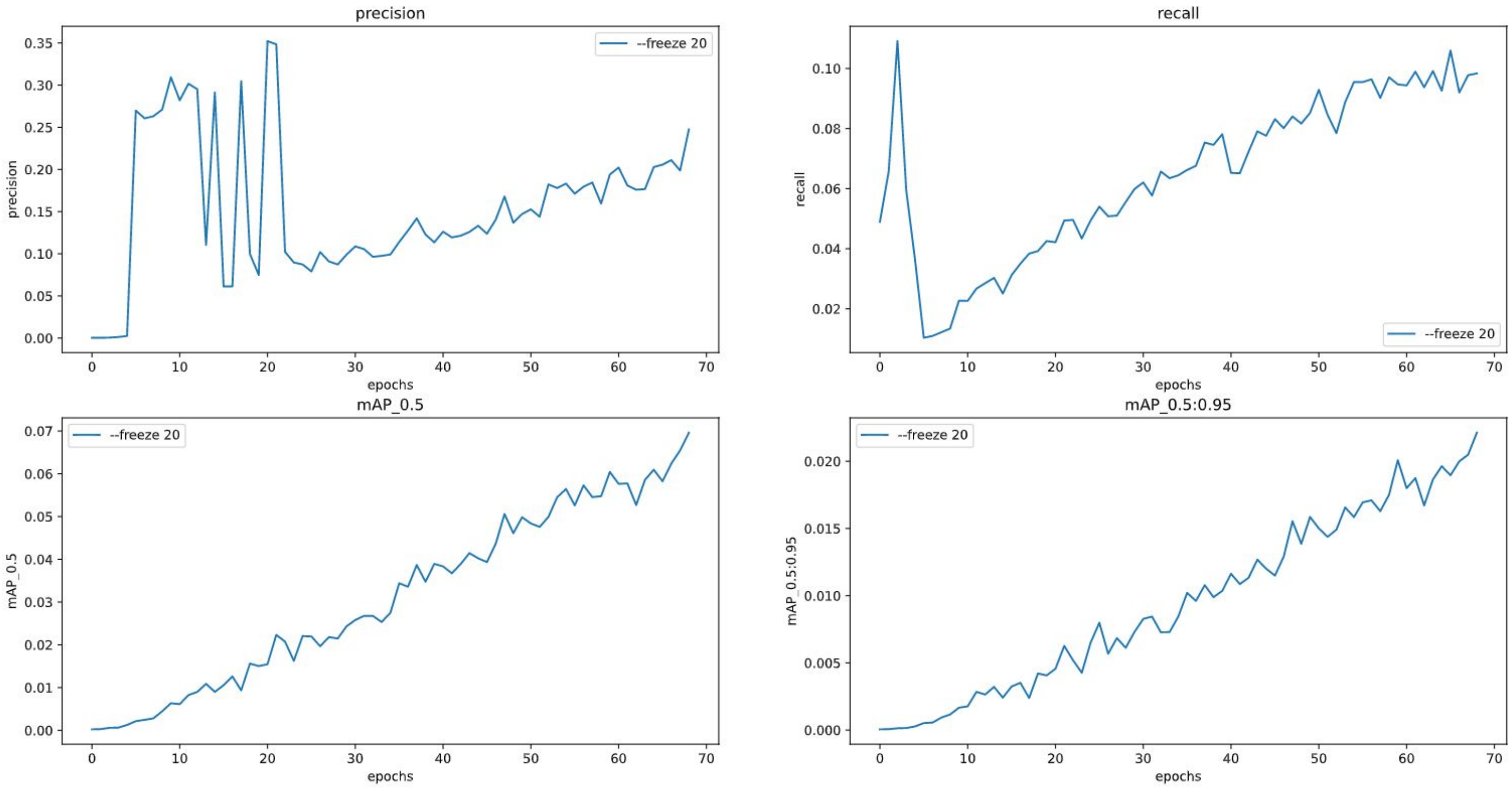
Using the winning Weights from 2020



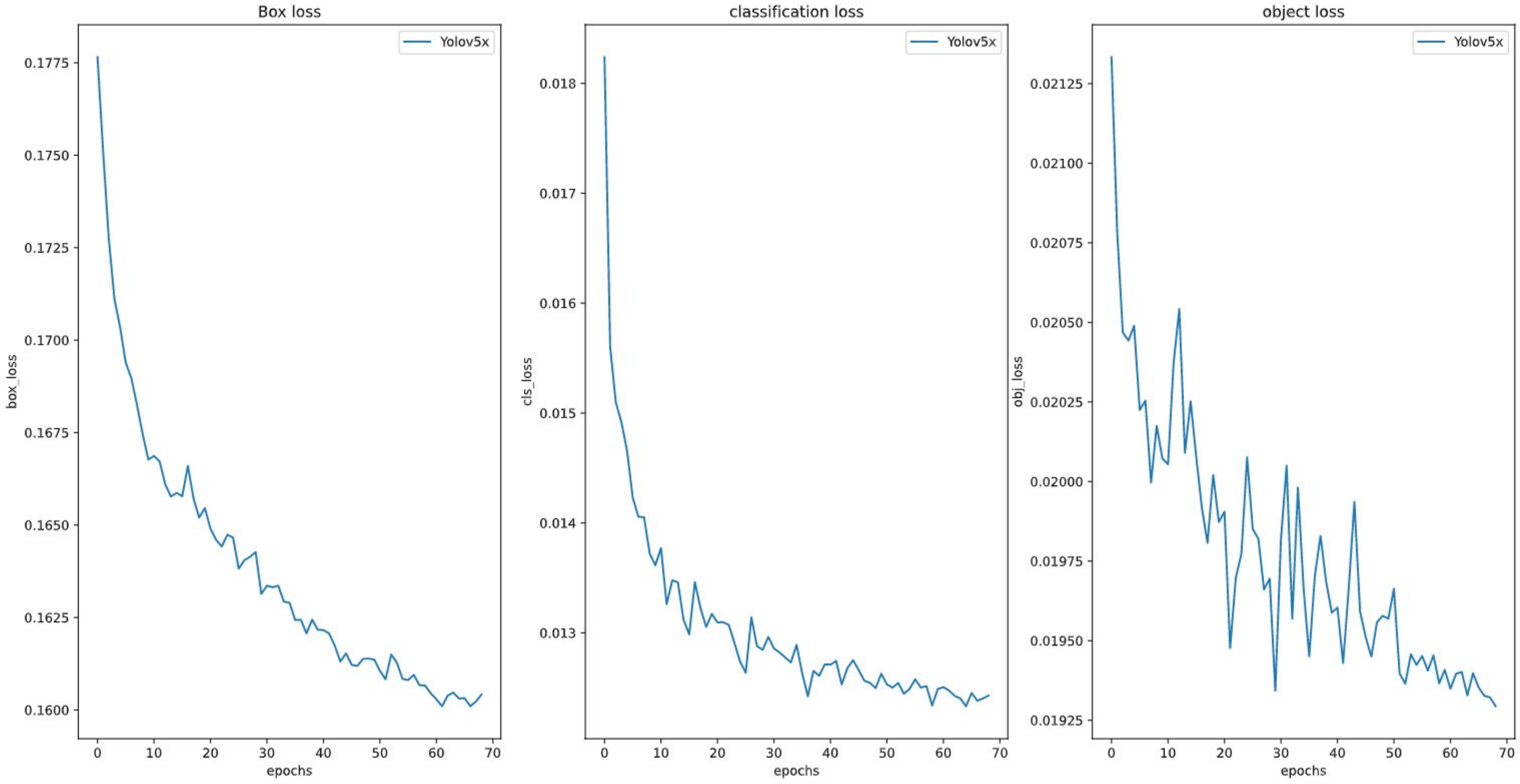
Finetuning winning weights from 2020

Model: Yolov5x
Runtime: 761 min
Epochs: 70
Best Map50:95 : 0.022125

Metrics


precision
--freeze 20
precision
epochs
recall
--freeze 20
recall
epochs
mAP_0.5
--freeze 20
mAP_0.5
epochs
mAP_0.5:0.95
--freeze 20
mAP_0.5:0.95
epochs

Validation Loss



Many weights - a lot of data

Sliding Window

Finetuning on cropped images ⇒ Inference using cropped images.

The accuracy didn't improve and detection time increased to 74 min.

Overfitting

More Regularisation: Add dropout

Weight decay

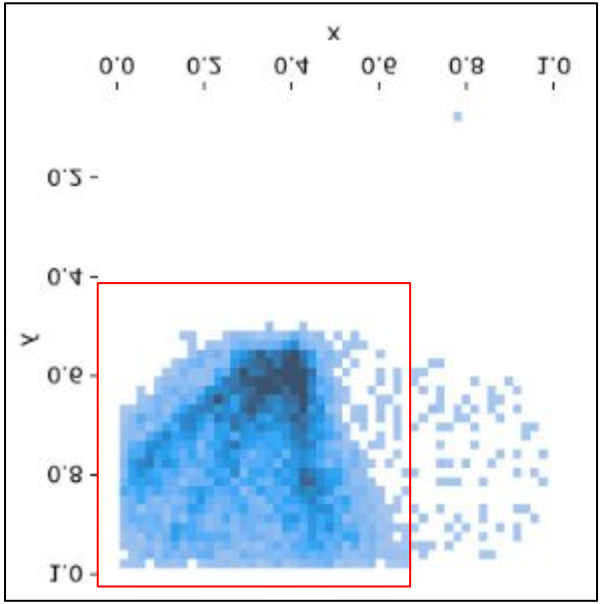
Use a less complex model

Sliding Window (Sahi)

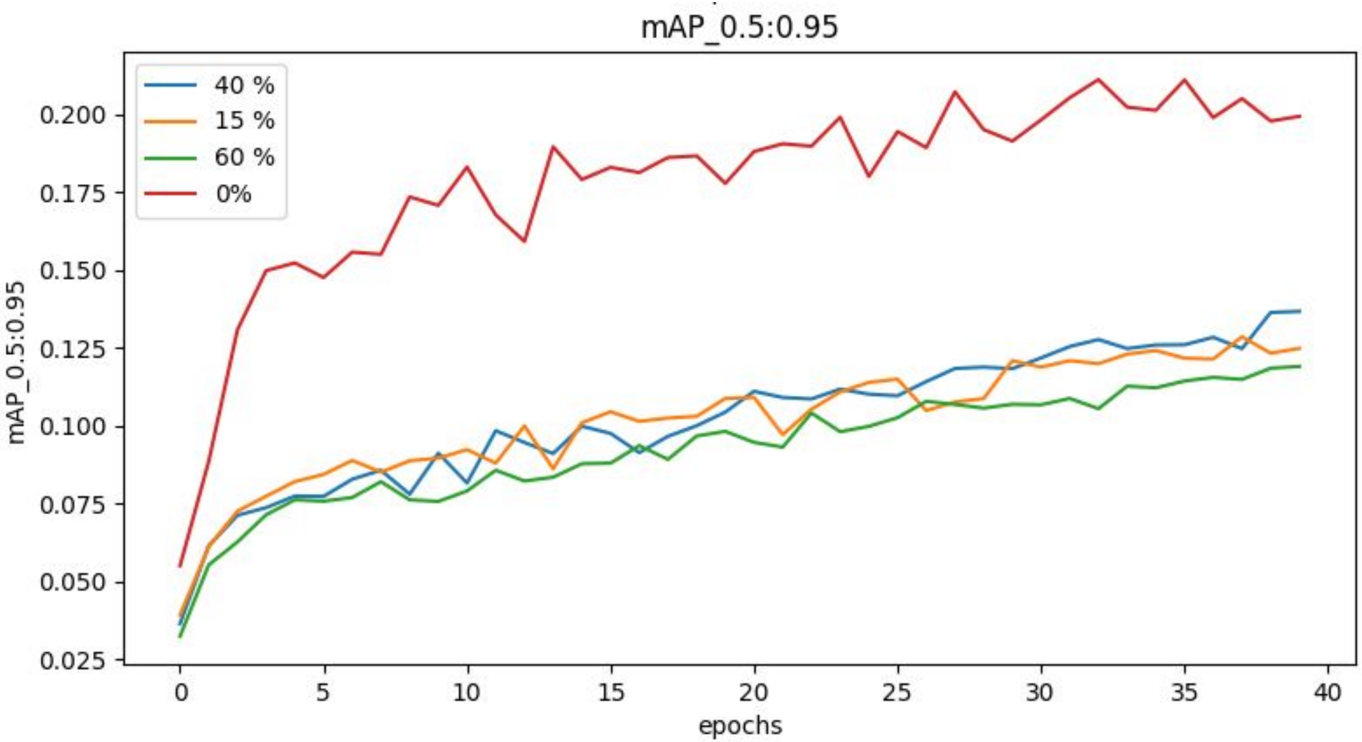
Runtime: 74 min

cropped training = cropped training

Increase robustness towards woods and skie == decrease amount of road crack predictions on roads.



Cropping test images before inference



Ensemble Learning

More robust model

Generalises ⇒ reduces overfitting

Increase accuracy and decreases variance

Thank you for listening

Sources

[1] RDD2022: A multi-national image dataset for automatic Road Damage Detection,

link: https://www.researchgate.net/publication/363668453_RDD2022_A_multi-national_image_dataset_for_automatic_Road_Damage_Detection

accessed: 07.03.2023

Authors Yoshihide Sekimoto, Durga Toshniwal, Sanjay Kumar Ghosh, Hiroya Maeda and Deeksha Arya

published: September 2022

Sources

[2] Global Road Damage Detection: State-of-the-art Solutions

link:
https://www.researchgate.net/publication/345989816_Global_Road_Damage_Detection_State-of-the-art_Solutions

accessed: 07.03.2023

Authors Yoshihide Sekimoto, Durga Toshniwal, Sanjay Kumar Ghosh, Hiroya Maeda , Hiroshi Omata, Takehiro Kashiyama and Deeksha Arya

published: November 2020

Sources

[3] Github

Author: Glenn Jocher.

Date: 2021 may 31

Accessed: 27.04.2023

Link: https://github.com/ultralytics/yolov5/issues/2844

Source

[4] SAHI

Github: https://github.com/obss/sahi

medium article: https://medium.com/codable/sahi-a-vision-library-for-performing-sliced-inference-on-large-images-small-objects-c8b086af3b80

published: 2021 jan 30

Accessed: 20.04

Source

[5] Dataset Quality

link: https://selectstar-ai.medium.com/diversity-accuracy-important-properties-of-your-dataset-e8b3072b29d6

accessed 14 April

published jul 20 2020

Source

[6] Mosaic

https://wandb.ai/iankelk/YOLOv5/reports/Part-II-Search-and-Rescue-Augmentation-and-preprocessing-on-drone-based-water-rescue-images-with-YOLOv5--VmlldzoyMDA0ODQ5

Yolov5s6 Architectures

[7]
https://dione.lib.unipi.gr/xmlui/bitstream/handle/unipi/14218/Giannios_mtn2003.pdf?sequence=3&isAllowed=y