GEORG-AUGUST-UNIVERSITÄT
GÖTTINGEN IN PUBLICA COMMODA SEIT 1737

Comparative Analysis of Neural Video Super-Resolution Methods for Object Detection on Mobile Devices

Mai Elshazly
mai.elshazly@stud.uni-goettingen.de

Supervisors: Prof. Dr. Xiaoming Fu, Dr. Tingting Yuan & Dr. Weijun Wang
23.05.2023

Relevant Background

- Adding more pixels to an image to enhance its quality is called **Super-Resolution (SR)**.
- We refer to quality by the vertical number of pixels (i.e. height) of an image.
- Neural super-resolution uses a **Deep Neural Network (DNN)** to achieve improved quality.
- Applying neural super-resolution to every frame of a video is called **Per-Frame Super-Resolution (Per-Frame SR)**.

Quality
1080p HD
720p
480p
360p
240p
144p

Size: 240p

Chunk0000 – Frame 0000

Size: 2160p

Chunk0000 – Frame 0000

Related Work

Some advances in enhancing video streaming using neural SR:

- M. Dasari, A. Bhattacharya, S. Vargas, P. Sahu, A. Balasubramanian, and S. R. Das. 2020. Streaming 360◦ Videos using Super-resolution. In Proceedings of the IEEE International Conference on Computer Communications (INFOCOM).

- Pan Hu, Rakesh Misra, and Sachin Katti. 2019. Dejavu: Enhancing Videoconfer- encing with Prior Knowledge. In Proceedings of the 20th International Workshop on Mobile Computing Systems and Applications. ACM, 63-68.

- Hyunho Yeo, Youngmok Jung, Jaehong Kim, Jinwoo Shin, and Dongsu Han. 2018. Neural adaptive content-aware internet video delivery. In 13th {USENIX } Symposium on Operating Systems Design and Implementation ({OSDI } 18). 645- 661.

- **H. Yeo, C. J. Chong, Y. Jung, J. Ye and D. Han, "NEMO: Enabling neural-enhanced video streaming on commodity mobile devices", Proc. 26th Annu. Int. Conf. Mobile Comput. Netw. (MobiCom), pp. 1-14, 2020.**

Motivation

- Smartphone/tablet video streaming accounts for 62% of viewership, and mobile devices account for more than 70% of YouTube video consumption*.

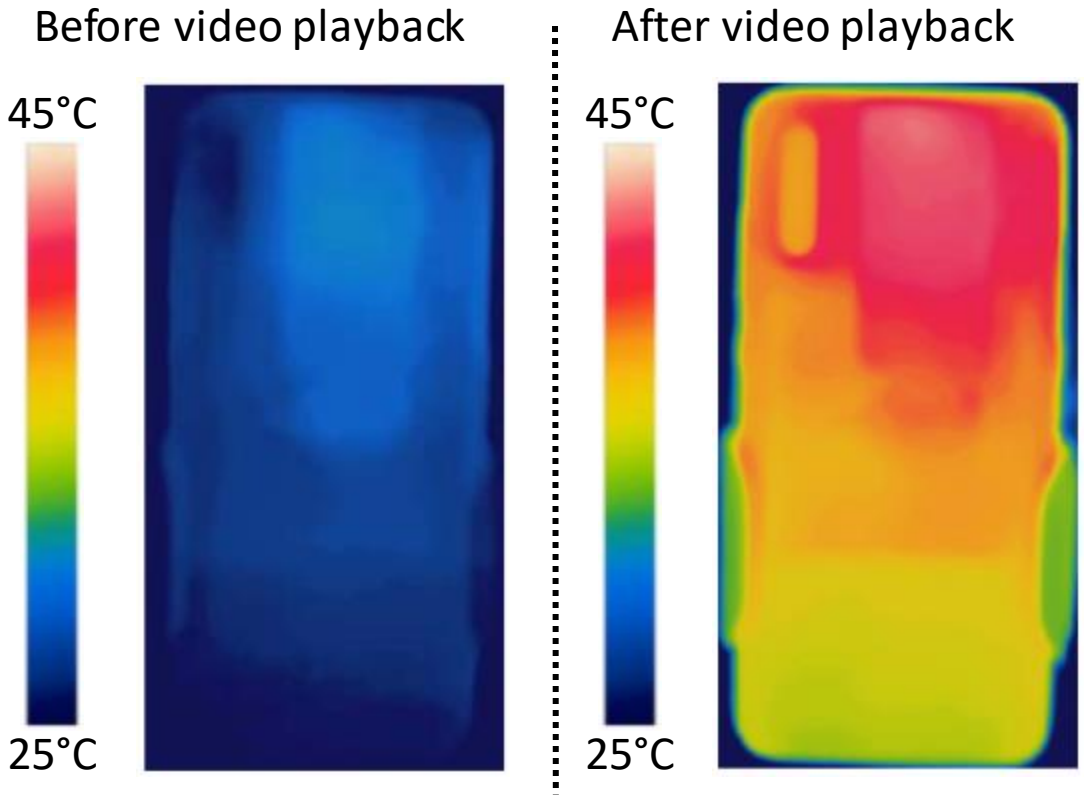
- Neural SR relies on client-side computation*, which makes per-frame SR impractical on mobile devices.

- Per-frame SR causes bad experiences for mobile users also due to high battery consumption and rise in device temperature.

* H. Yeo, C. J. Chong, Y. Jung, J. Ye and D. Han, "NEMO: Enabling neural-enhanced video streaming on commodity mobile devices", *Proc. 26th Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, pp. 1-14, 2020.

Motivation continue

Infrared thermal images of heat dissipation on a smartphone caused by per-frame DNN inference*



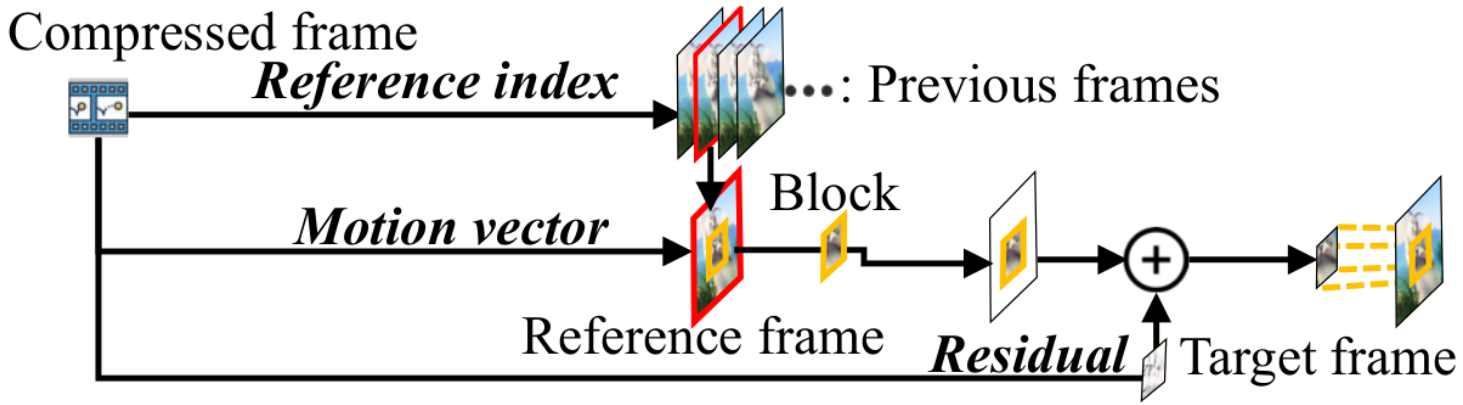
* H. Yeo, C. J. Chong, Y. Jung, J. Ye and D. Han, "NEMO: Enabling neural-enhanced video streaming on commodity mobile devices", *Proc. 26th Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, pp. 1-14, 2020.

NEMO – Neural-Enhanced Video Streaming on Mobile Devices

- NEMO selects only a few frames for neural SR; these frames are called **Anchor Points (APs)** while other frames are called **Non-Anchor Points**.

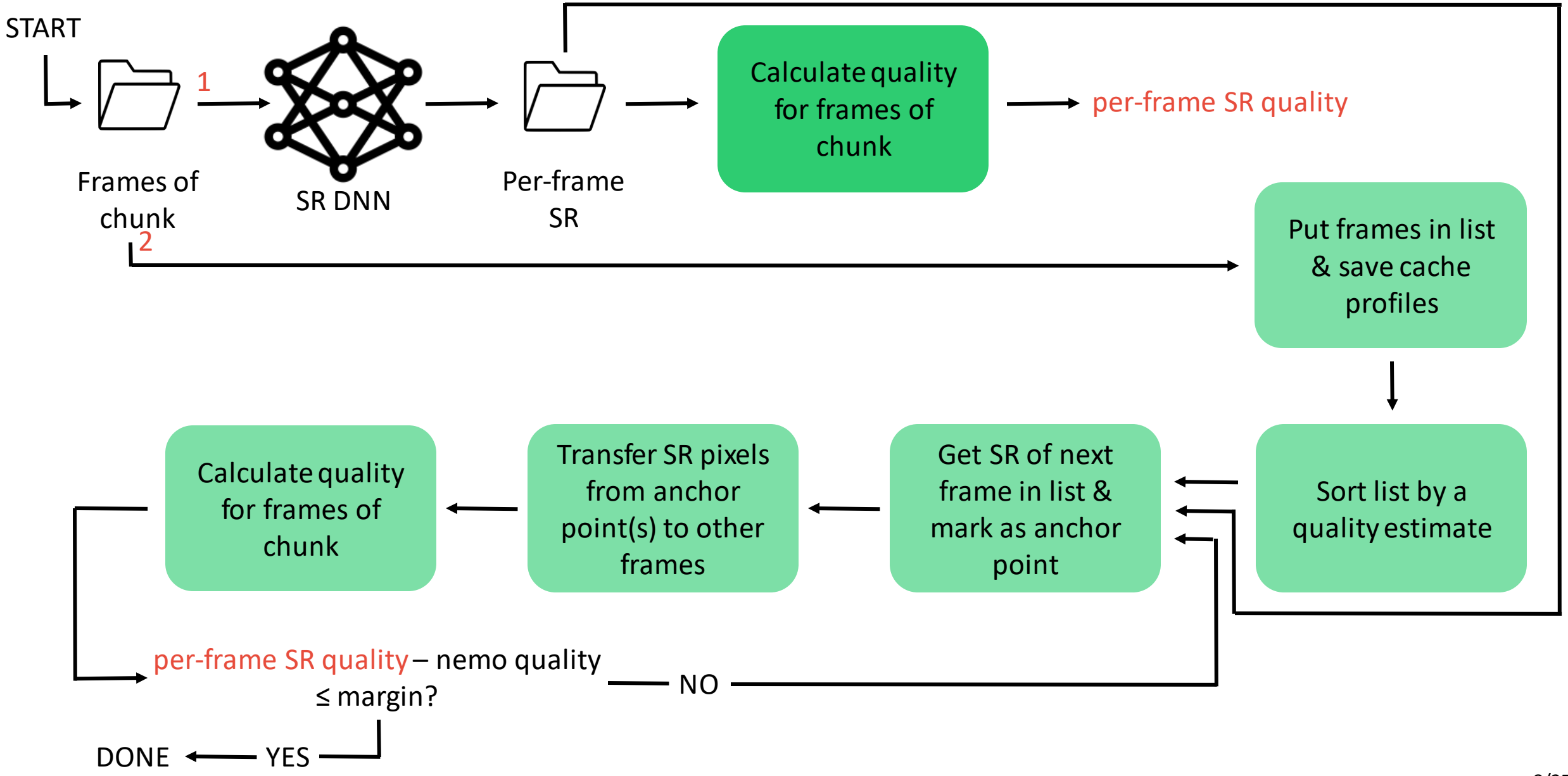
- NEMO uses a codec to get dependencies among frames so that non-APs benefit from APs.

Frame dependencies processed in a codec*



* H. Yeo, C. J. Chong, Y. Jung, J. Ye and D. Han, "NEMO: Enabling neural-enhanced video streaming on commodity mobile devices", *Proc. 26th Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, pp. 1-14, 2020.

Anchor Point Selection in a Video Chunk in NEMO

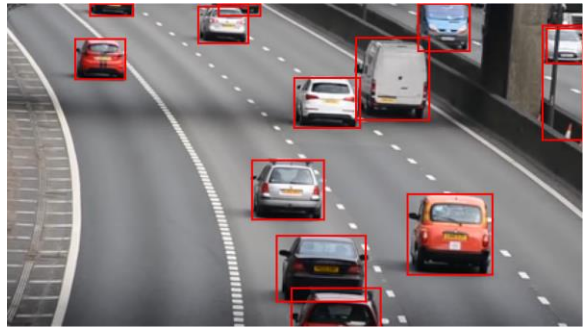


Introduction to this Research Project

- **Objective:** Creation of ground truth bounding boxes (BBs) from upscaled video on mobile devices for further object detection.

- **Target Metrics:**
  - SR DNN time required (seconds)
  - Accuracy against per-frame SR (f1-score)

Chunk0009 – Frame 0059

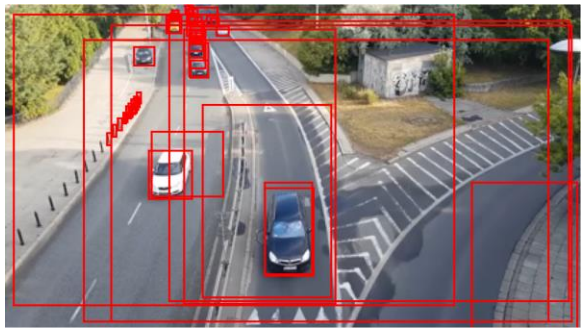


- **Questions to investigate:**
  - Can NEMO achieve the objective?
  - Does **applying SR DNN only to objects in APs (per-bb SR)** instead of complete APs achieve the objective with shorter time?
  - How about investigating only small objects in APs? What are small objects?

Object Detector

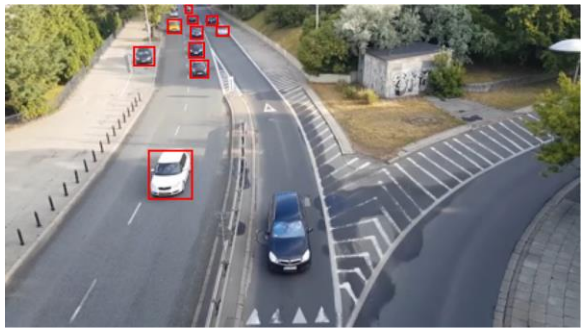
- Faster R-CNN with ResNet50 pretrained on COCO as backbone (weights found on Kaggle*).

- Can detect:
  - **Vehicle –** car, bus, train, truck
  - **Persons –** person, bicycle, motorcycle
  - **Roadside-objects –** traffic light, fire hydrant, stop sign, parking meter



Chunk0020 – Frame 0022

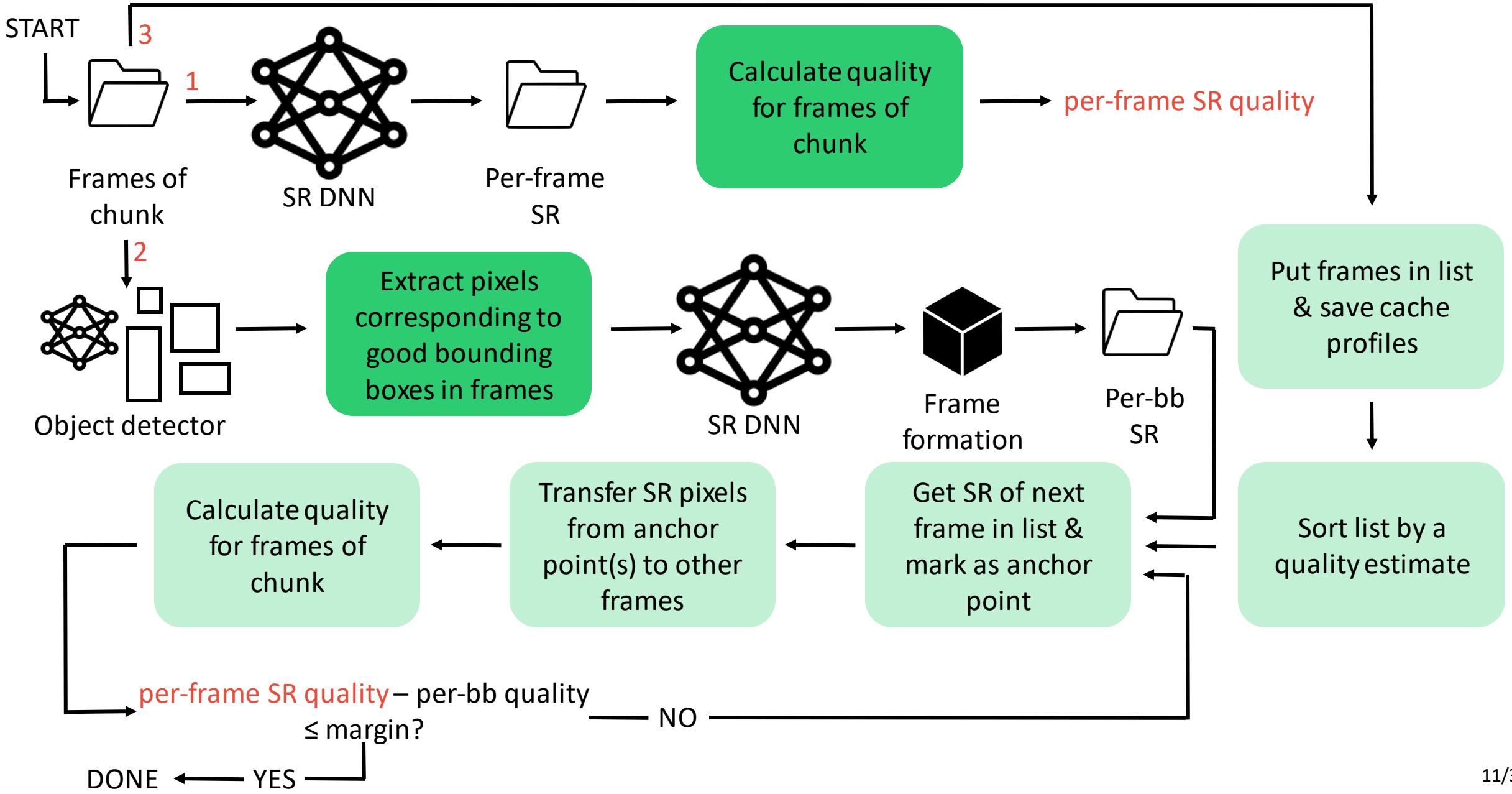
Filter with BBs with 50% confidence



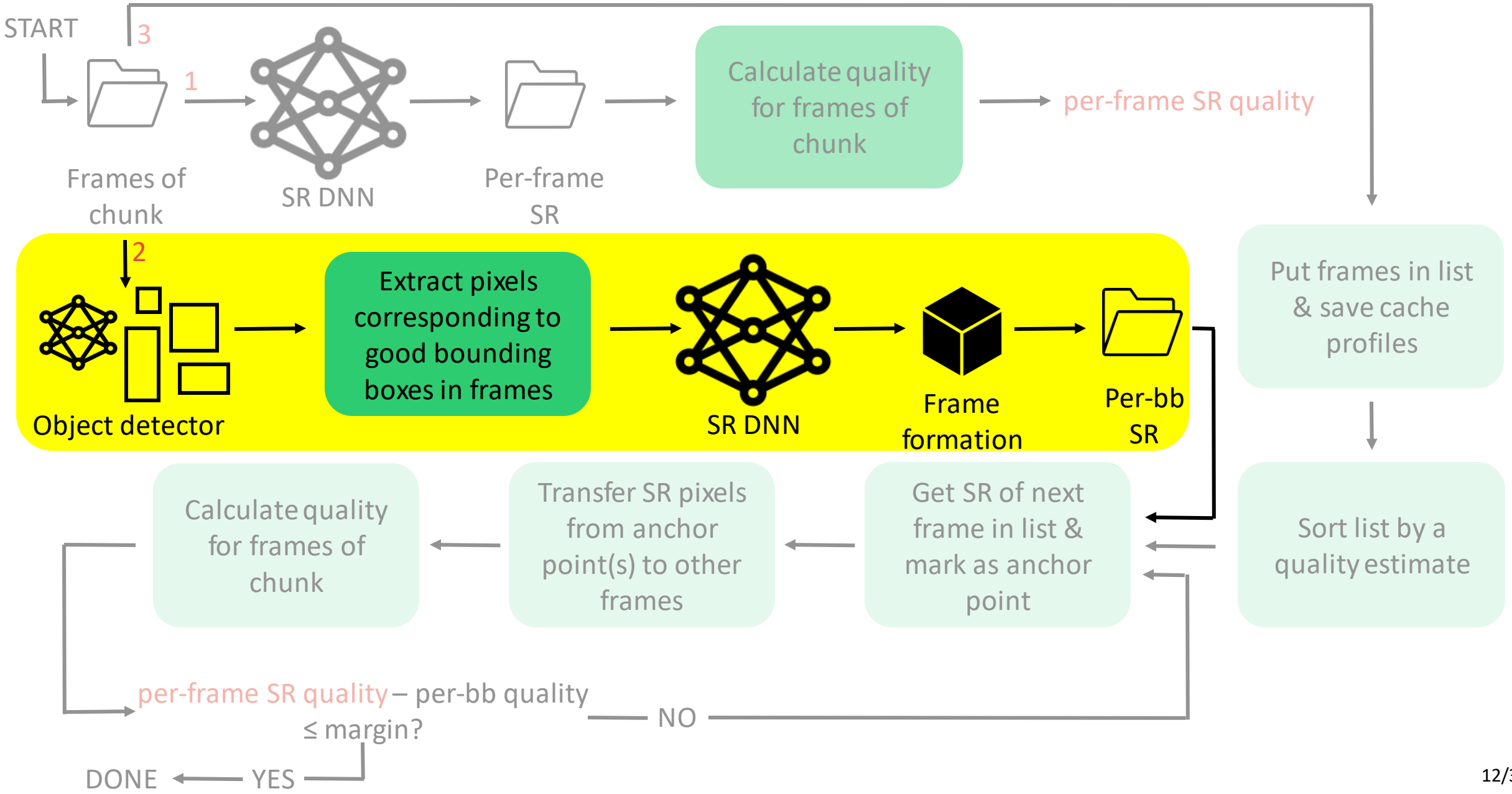
Chunk0020 – Frame 0022

* https://www.kaggle.com/datasets/n1t1nk/fasterrcnn-resnet50-fpn-coco?resource=download

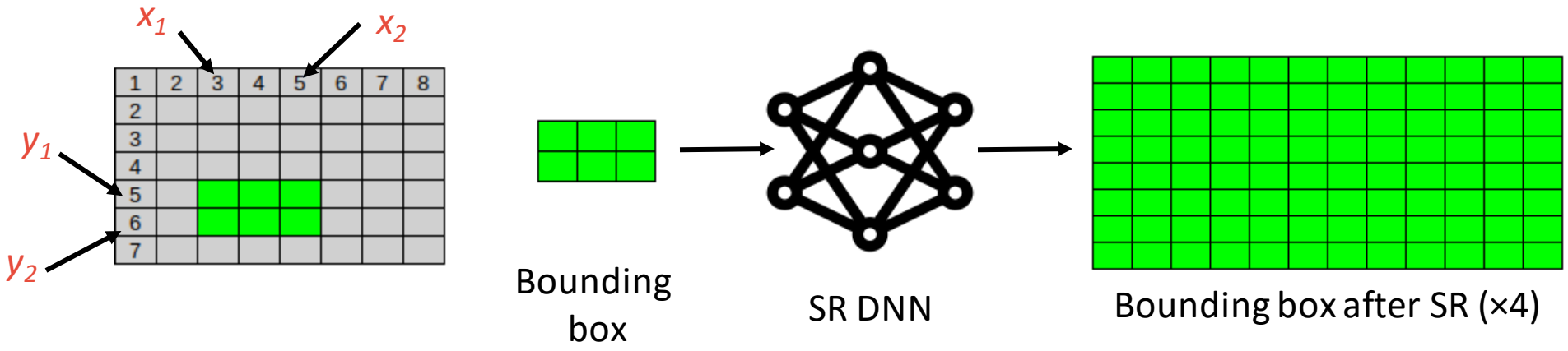
Anchor Point Selection in a Video Chunk in Our Method



Anchor Point Selection in a Video Chunk in Our Method



Frame Formation as a Tensor



0-based indices of BB in upscaled tensor:

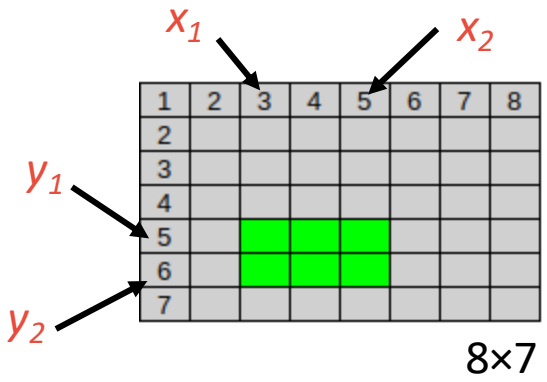
$r_1 = (y_1 - 1) \times scale$
$= (5 - 1) \times 4 = 16$

$r_2 = r_1 - 1 + ((y_2 - y_1 + 1) \times scale)$
$= 16 - 1 + ((6 - 5 + 1) \times 4) = 23$

$c_1 = (x_1 - 1) \times scale$
$= (3 - 1) \times 4 = 8$

$c_2 = c_1 - 1 + ((x_2 - x_1 + 1) \times scale)$
$= 8 - 1 + ((5 - 3 + 1) \times 4) = 19$

Frame Formation as a Tensor continue



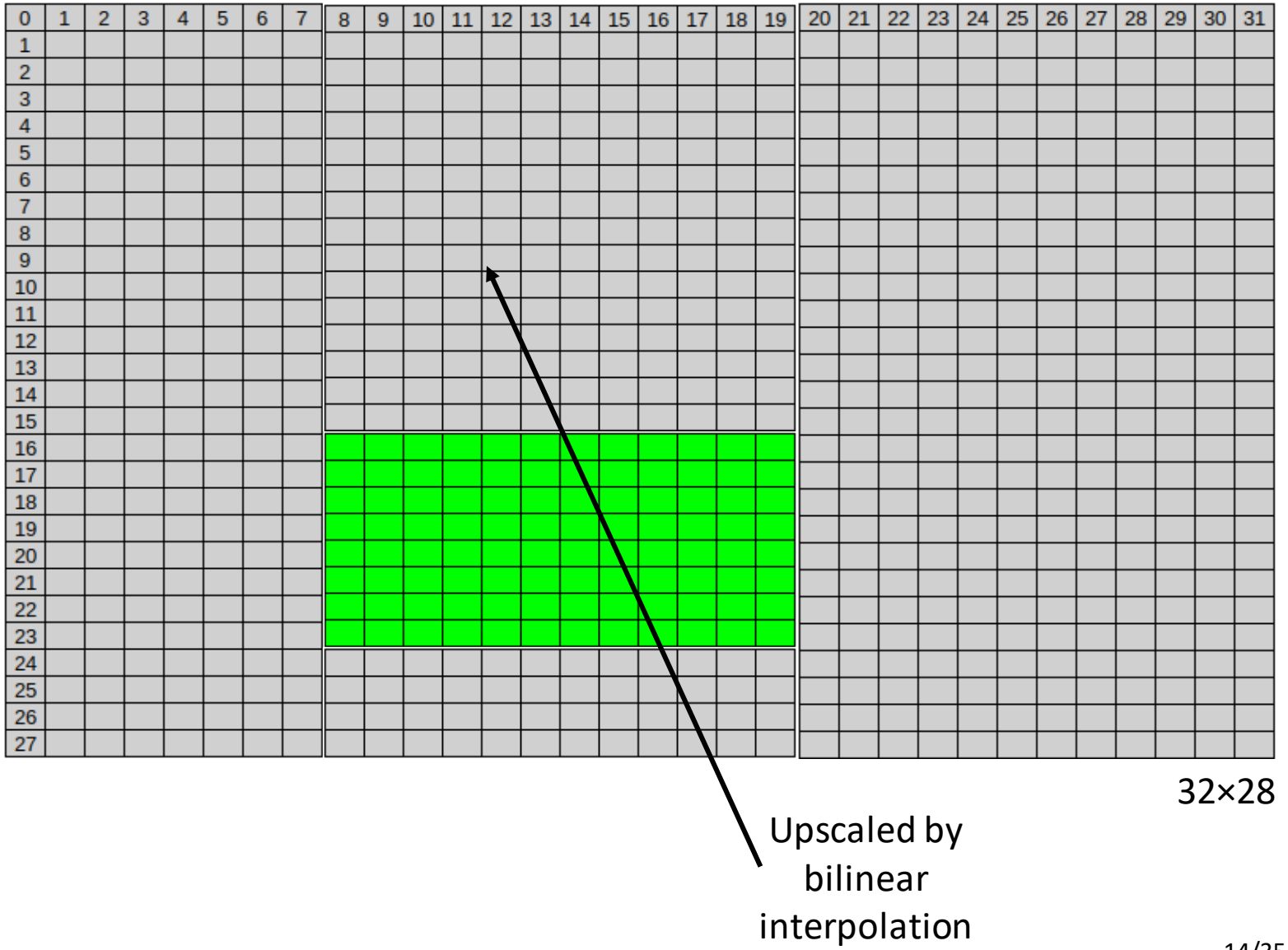
0-based indices of BB in upscaled tensor:

$r_1 = (y_1 - 1) \times scale$
$= (5 - 1) \times 4 = 16$

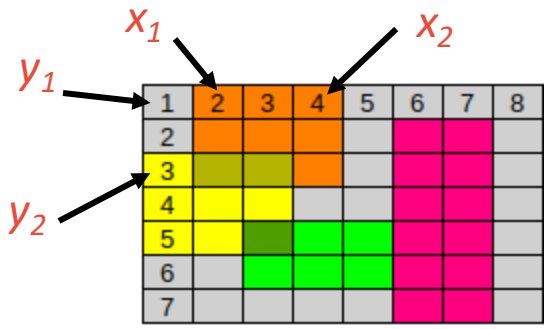
$r_2 = r_1 - 1 + ((y_2 - y_1 + 1) \times scale)$
$= 16 - 1 + ((6 - 5 + 1) \times 4) = 23$

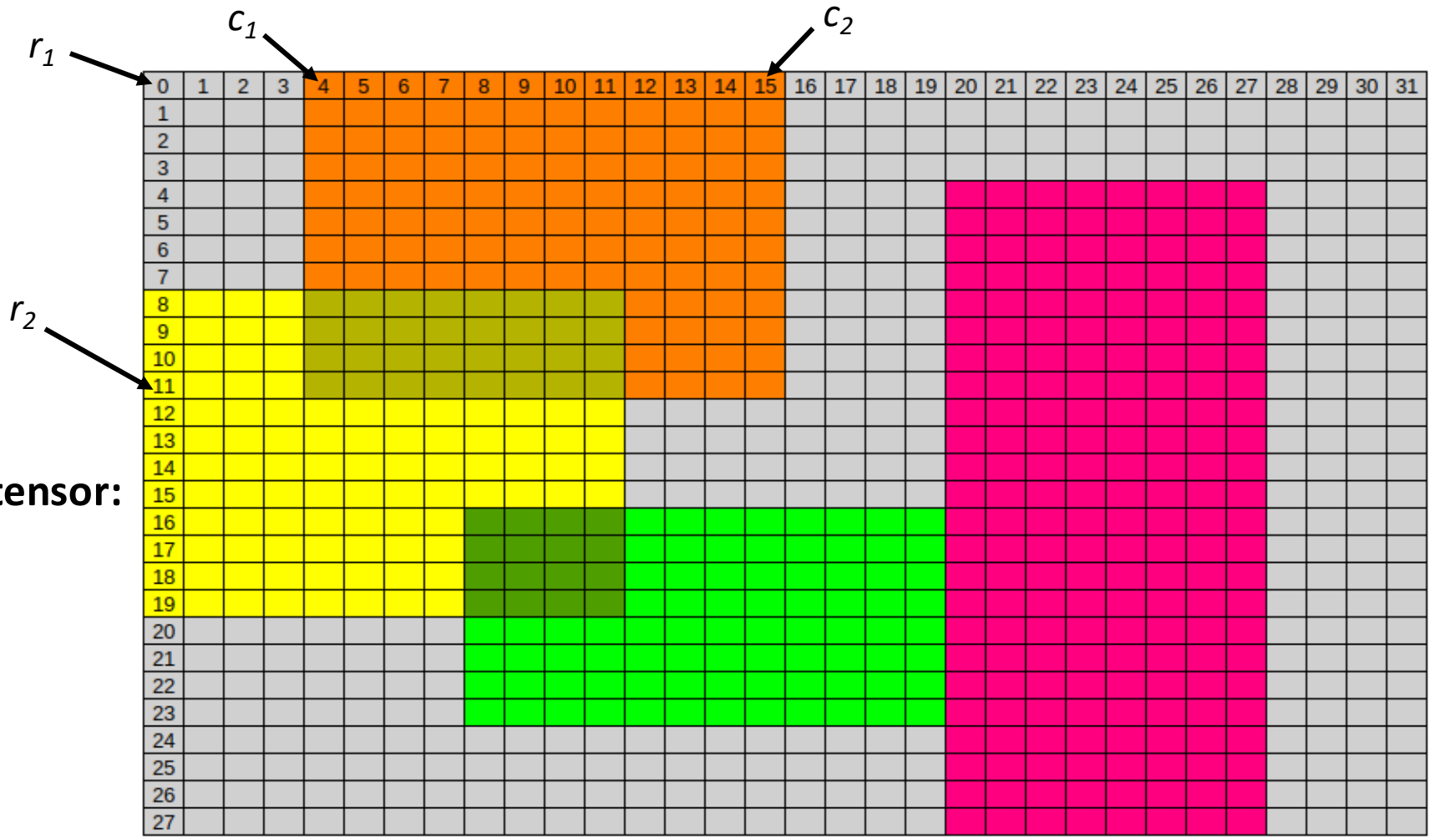
$c_1 = (x_1 - 1) \times scale$
$= (3 - 1) \times 4 = 8$

$c_2 = c_1 - 1 + ((x_2 - x_1 + 1) \times scale)$
$= 8 - 1 + ((5 - 3 + 1) \times 4) = 19$



Frame Formation as a Tensor continue





0-based indices of BB in upscaled tensor:

$r_1 = (y_1 - 1) \times scale$
$= (1 - 1) \times 4 = 0$

$r_2 = r_1 - 1 + ((y_2 - y_1 + 1) \times scale)$
$= 0 - 1 + ((3 - 1 + 1) \times 4) = 11$

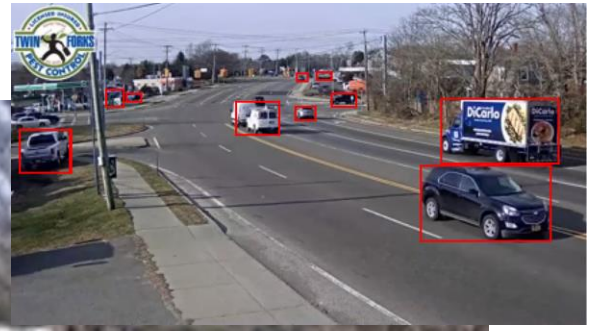
$c_1 = (x_1 - 1) \times scale$
$= (2 - 1) \times 4 = 4$

$c_2 = c_1 - 1 + ((x_2 - x_1 + 1) \times scale)$
$= 4 - 1 + ((4 - 2 + 1) \times 4) = 15$

Size: 1080p – Per-Frame SR

Chunk0001 – Frame 0018

Size: 1080p – Anchor Point in Per-BB SR



Chunk0001 – Frame 0018

Size: 1080p – Non-Anchor Point in Per-BB SR

Chunk0001 – Frame 0010

Quality, Anchor Points, and DNN Latency Results

Chunk	Per-Frame SR		AP SR (NEMO)			AP Per-BB SR (Our Method)		
	PSNR (dB)	Time (s)	PSNR (dB)	APs Count	Time (s)	PSNR (dB)	APs Count	Time (s)
0	30.03	57.23	29.56	6	2.86	26.87	8	4.16
1	31.19	54.85	30.77	3	1.37	27.68	8	2.91
2	32.02	54.63	31.61	2	0.91	28.35	8	2.46
3	32.53	54.62	32.23	1	0.46	29.11	8	3.20
4	32.56	54.67	32.45	6	2.73	29.85	8	4.68
5	32.08	54.52	31.93	2	0.91	29.30	8	3.58
6	31.71	54.56	31.51	5	2.27	28.93	8	4.55
7	31.68	55.61	31.49	1	0.46	29.11	8	5.13

120 frames/chunk, 25 chunks in total

Results of all 25 chunks are available here: https://drive.google.com/drive/folders/1vEaI4w8Lj5xC6_aGokVjaTzkL6XBIZ_h?usp=share_link

Per-BB SR Yields Very Low PSNR Unlike NEMO

Chunk	Per-Frame SR		AP SR (NEMO)			AP Per-BB SR (Our Method)		
	PSNR (dB)	Time (s)	**PSNR (dB)**	APs Count	Time (s)	**PSNR (dB)**	APs Count	Time (s)
0	**30.03**	57.23	**29.56**	6	2.86	**26.87**	8	4.16
1	**31.19**	54.85	**30.77**	3	1.37	**27.68**	8	2.91
2	**32.02**	54.63	**31.61**	2	0.91	**28.35**	8	2.46
3	**32.53**	54.62	**32.23**	1	0.46	**29.11**	8	3.20
4	**32.56**	54.67	**32.45**	6	2.73	**29.85**	8	4.68
5	**32.08**	54.52	**31.93**	2	0.91	**29.30**	8	3.58
6	**31.71**	54.56	**31.51**	5	2.27	**28.93**	8	4.55
7	**31.68**	55.61	**31.49**	1	0.46	**29.11**	8	5.13

Allowed quality margin = 0.5

DNN Latency of Per-BB SR is Longer Than Expected

Chunk	Per-Frame SR		AP SR (NEMO)			AP Per-BB SR (Our Method)		
	PSNR (dB)	Time (s)	PSNR (dB)	APs Count	**Time (s)**	PSNR (dB)	APs Count	**Time (s)**
0	30.03	57.23	29.56	6	**2.86**	26.87	8	**4.16**
1	31.19	54.85	30.77	3	**1.37**	27.68	8	**2.91**
2	32.02	54.63	31.61	2	**0.91**	28.35	8	**2.46**
3	32.53	54.62	32.23	1	**0.46**	29.11	8	**3.20**
4	32.56	54.67	32.45	6	**2.73**	29.85	8	**4.68**
5	32.08	54.52	31.93	2	**0.91**	29.30	8	**3.58**
6	31.71	54.56	31.51	5	**2.27**	28.93	8	**4.55**
7	31.68	55.61	31.49	1	**0.46**	29.11	8	**5.13**

APs of per-bb SR are not much more in chunk 0

time(NEMO) << time(per-bb SR)

DNN Latency in Per-BB SR Includes Additional Processing

Chunk	BBs Count	AP Per-BB SR Raw DNN Time (s)	AP Per-BB SR Total Processing Time (s)
0	1623	**2.83**	**4.16**
1	1134	**1.86**	**2.91**
2	953	**1.56**	**2.46**
3	1365	**2.19**	**3.20**
4	2225	**3.35**	**4.68**
5	1353	**2.30**	**3.58**
6	1922	**3.38**	**4.55**
7	2519	**3.82**	**5.13**

8 anchor points each

BBs wait in a queue to get SR applied; can't be processed in batches

F1-Scores of Bounding Boxes Against Per-Frame-SR

Chunk	240p	AP SR (NEMO)	AP Per-BB SR (Our Method)
0	0.66	0.88	0.66
1	0.85	0.92	0.85
2	0.80	0.91	0.82
3	0.85	0.89	0.86
4	0.86	0.89	0.86
5	0.78	0.87	0.80
6	0.74	0.84	0.77
7	0.81	0.87	0.86

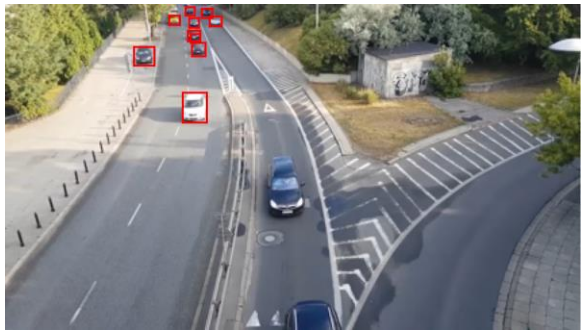
Results of all 25 chunks are available here: https://drive.google.com/drive/folders/1vEaI4w8Lj5xC6_aGokVjaTzkL6XBIZ_h?usp=share_link

How About Per-Small-BB SR? What is a Small BB?

- In COCO*, a bounding box with area < 32^2 is considered small. Such BBs occupy 41% of objects.

- This should be interpreted in a way that suits our frames.

- In our frames, a bounding box with area < 20^2 is considered small. Such BBs occupy 59.01% of objects.

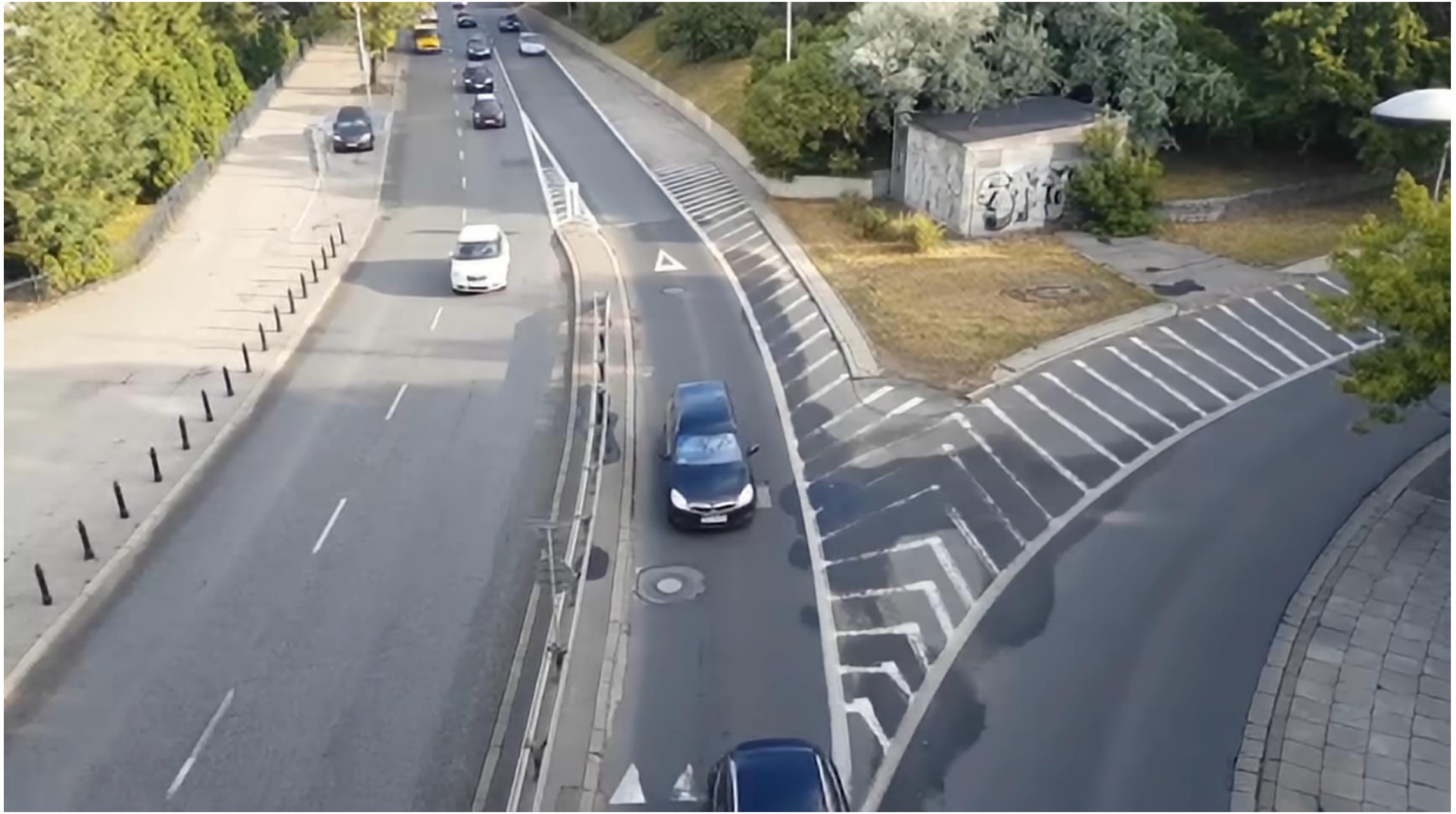


Chunk0020 – Frame 0000

Try the code here to set your own threshold and see how much % its BBs occupy: https://drive.google.com/drive/folders/1QIUvpVJsWzHyT0R0wNeW4yP3VnjjqfJF?usp=share_link

* https://cocodataset.org/#detection-eval

Size: 1080p – Per-Frame SR



Chunk0020 – Frame 0000

Size: 1080p – Anchor Point in Per-Small-BB SR

Chunk0020 – Frame 0000

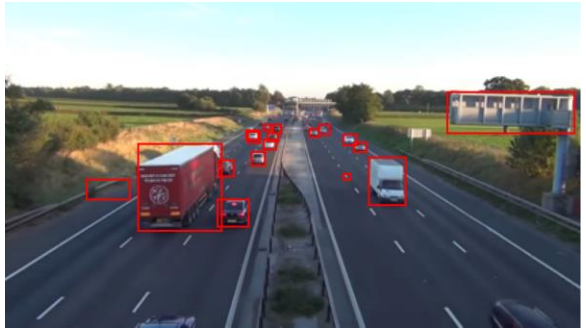
Size: 1080p – Anchor Point in Per-BB SR

Chunk0020 – Frame 0000

DNN Latency in Per-Small-BB SR as Compared to Per-BB SR

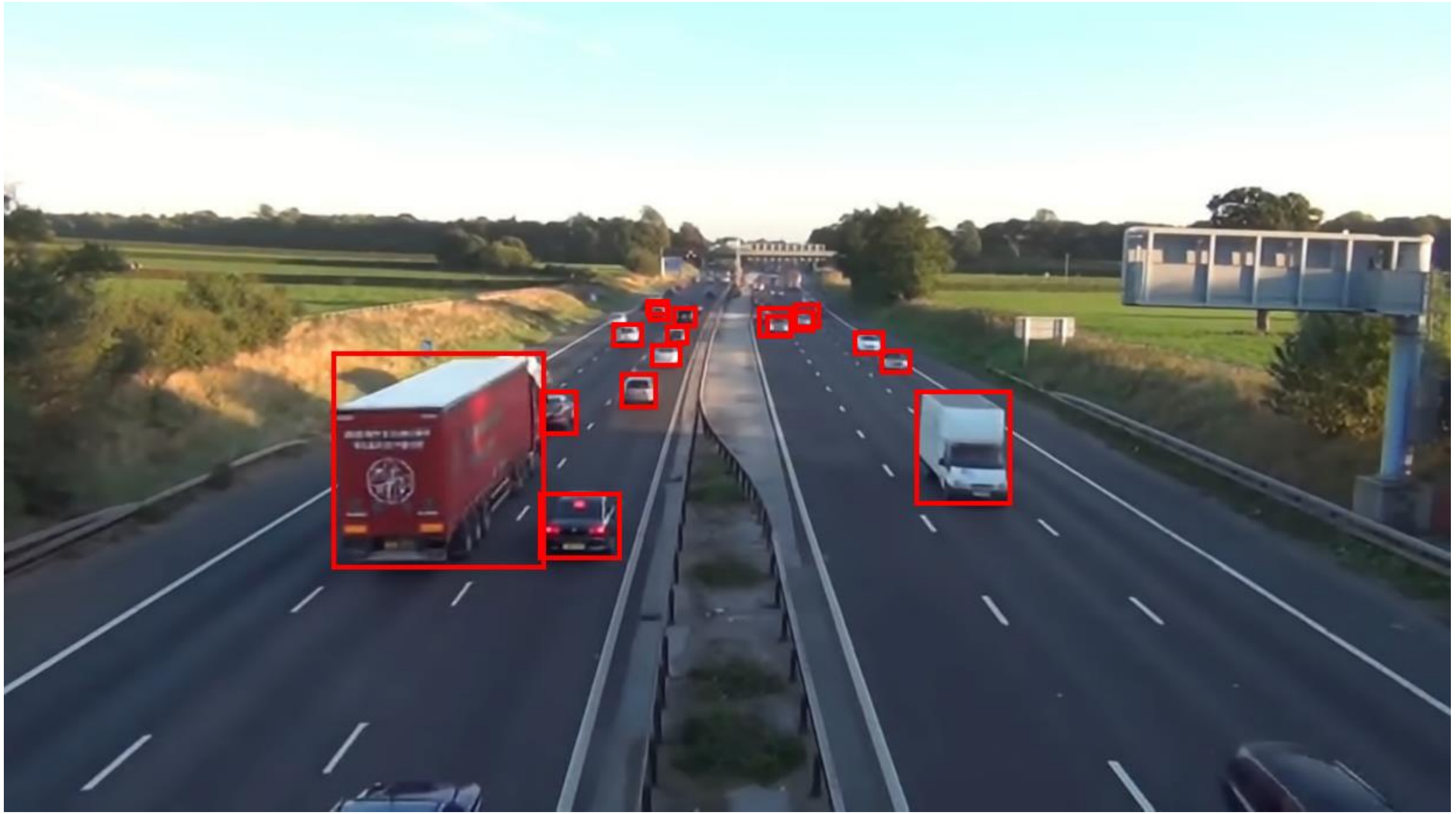
Chunk	Small BBs Count	AP Per-Small-BB SR Total Processing Time (s)	AP Per-BB SR Total Processing Time (s)
0	938	**2.35**	**4.16**
1	863	**1.86**	**2.91**
2	734	**1.50**	**2.46**
3	1060	**1.79**	**3.20**
4	1616	**2.40**	**4.68**
5	873	**1.88**	**3.58**
6	1203	**2.19**	**4.55**
7	1731	**2.59**	**5.13**

Size: 240p – Input Frame with Bounding Boxes



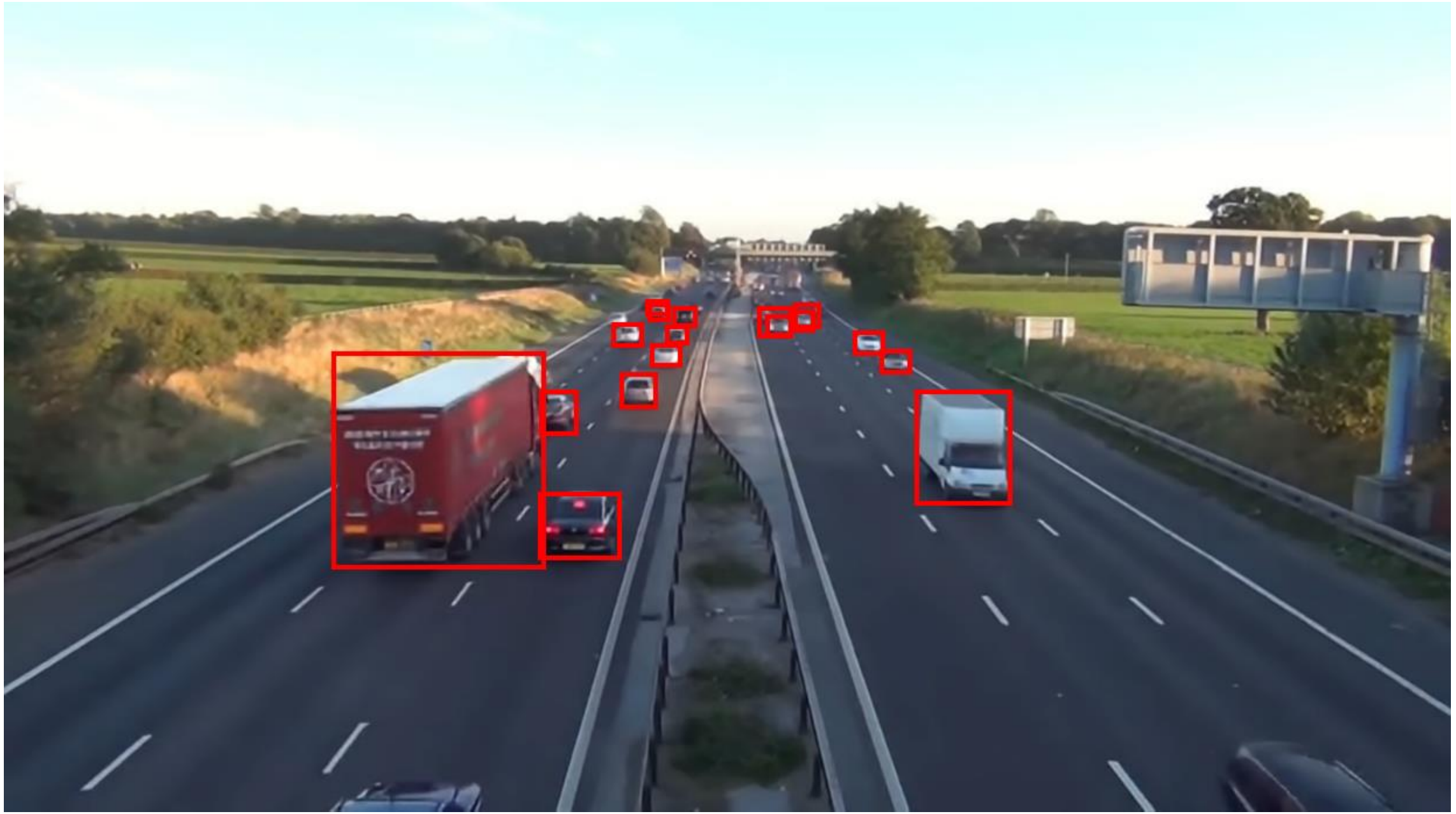
Chunk0004 – Frame 0012

Size: 1080p – BBs After Per-Frame SR



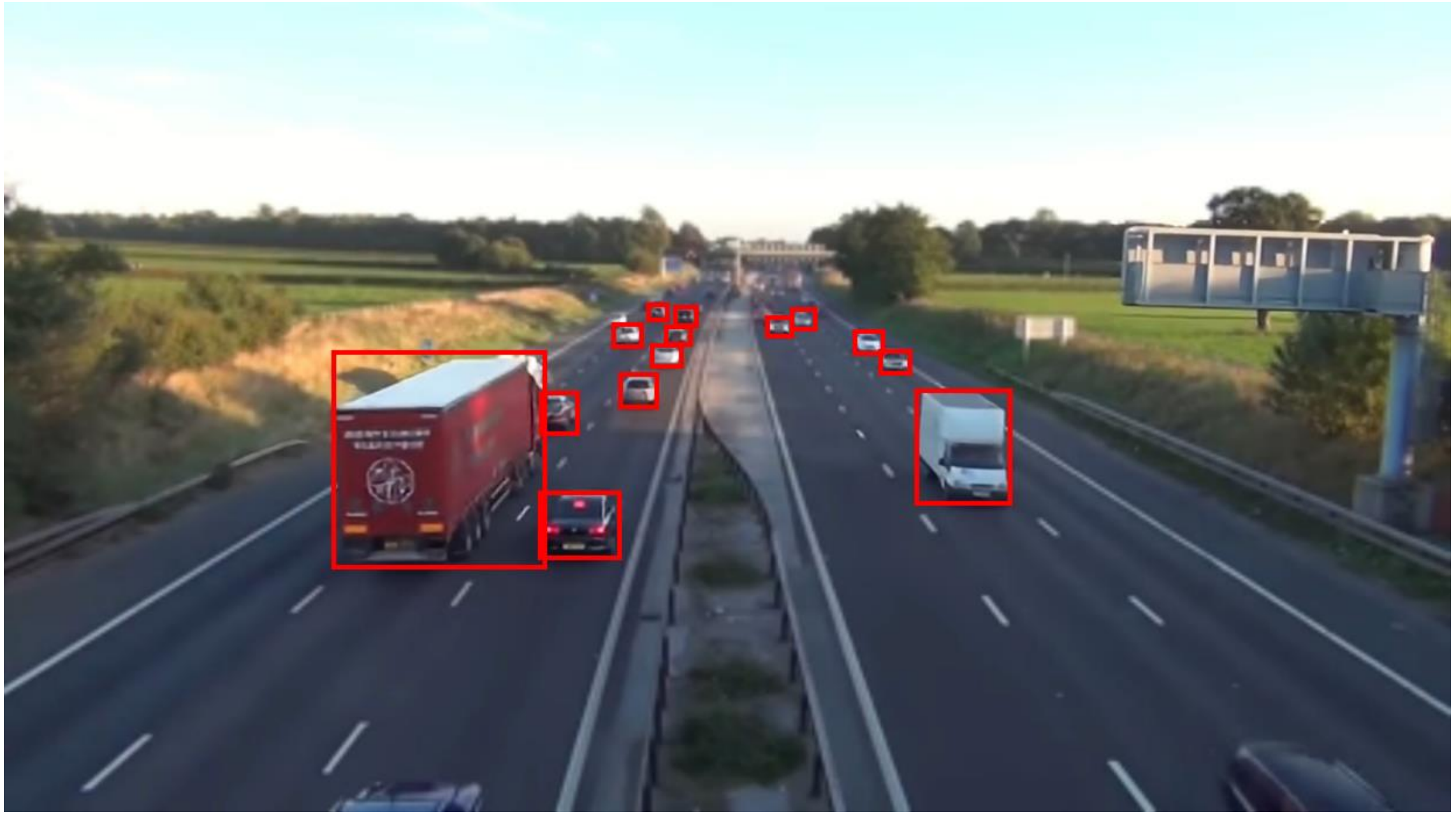
Chunk0004 – Frame 0012

Size: 1080p – BBs After NEMO



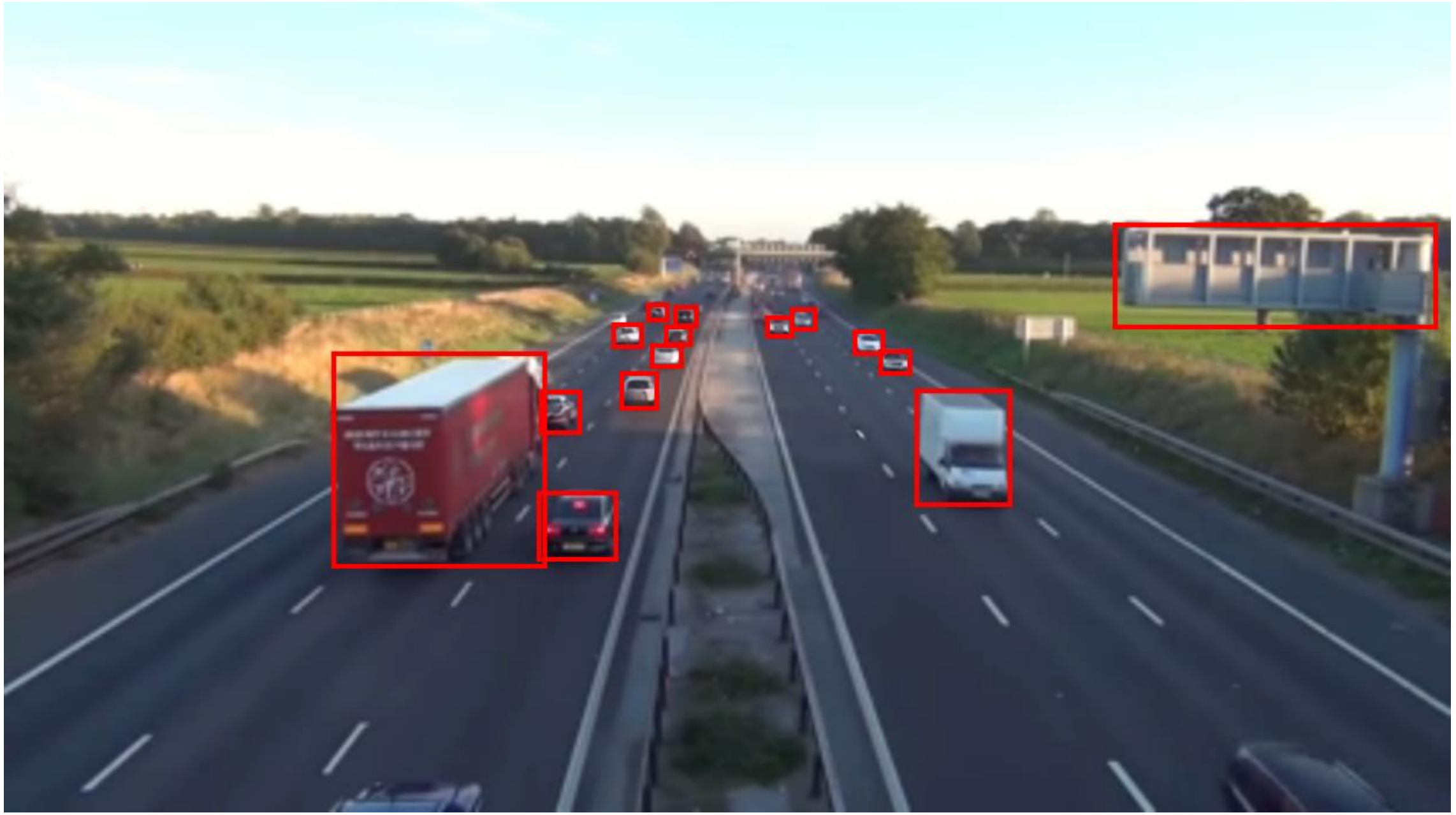
Chunk0004 – Frame 0012

Size: 1080p – BBs After Per-BB SR



Chunk0004 – Frame 0012

Size: 1080p – BBs After Per-Small-BB SR



Chunk0004 – Frame 0012

F1-Scores of Bounding Boxes Against Per-Frame-SR

Method	Average F1-Score in Video
240p	0.84
AP SR (NEMO)	0.90
AP Per-BB SR (Our Main Method)	0.87
AP Per-Small-BB SR (Additional Method)	0.86

Conclusion

- NEMO achieves ground truth BBs with high accuracy and within acceptable DNN latency.
- Per-bb SR takes longer time than NEMO since BBs wait in a queue to get neural SR applied.
- BBs cannot be processed in a batch due to their different sizes.
- Per-small-bb SR takes way shorter time than per-bb SR with close accuracy.
- **Suggestions for future work for per-bb SR DNN time:**
  - Change anchor points limit
  - Change quality margin
  - Change threshold of confidence score
  - Choose a fixed size for BBs and work on them as a batch

Thank you