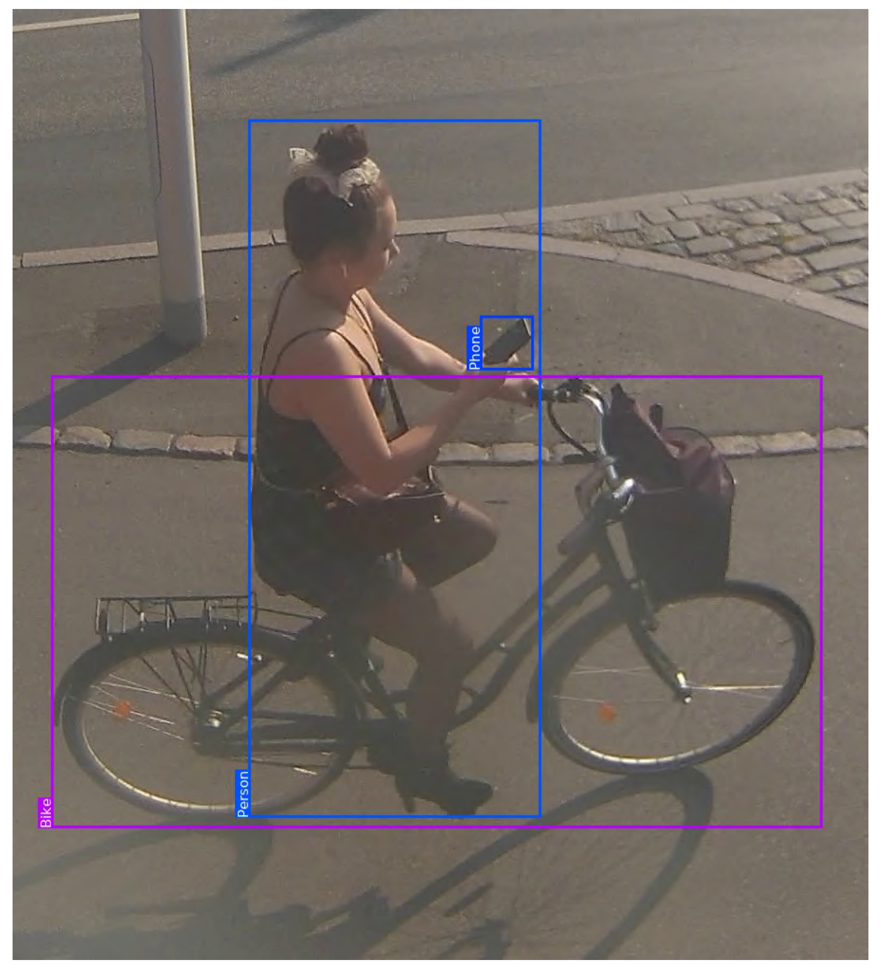
# Bikes and helmets detection using DETR

*Apolline Dersy, Gabriel Fournier, Jasmine Truchot*

DTU Compute, Technical University of Denmark

## Introduction

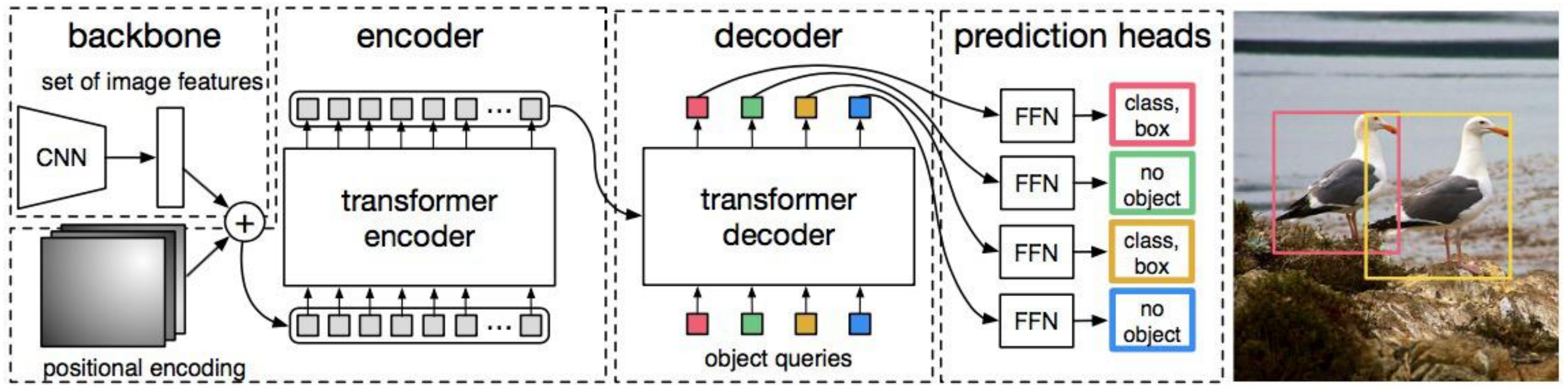
Object detection aims at locating and identifying objects on an image. In this project, we have used the DETR model made by Facebook AI to detect bikes, persons, helmets and phones on frames from a video dataset taken in Copenhagen.



**Steps :**

1. **Split** the videos into frames
2. Select and **label** the frames
3. **Use the DETR** model on these frames
4. **Improve** the model with augmentation

## Model : DEtection with Transformers (DETR) [1, 2]



### Loss function

**1. Pair-match predictions and real bbox**

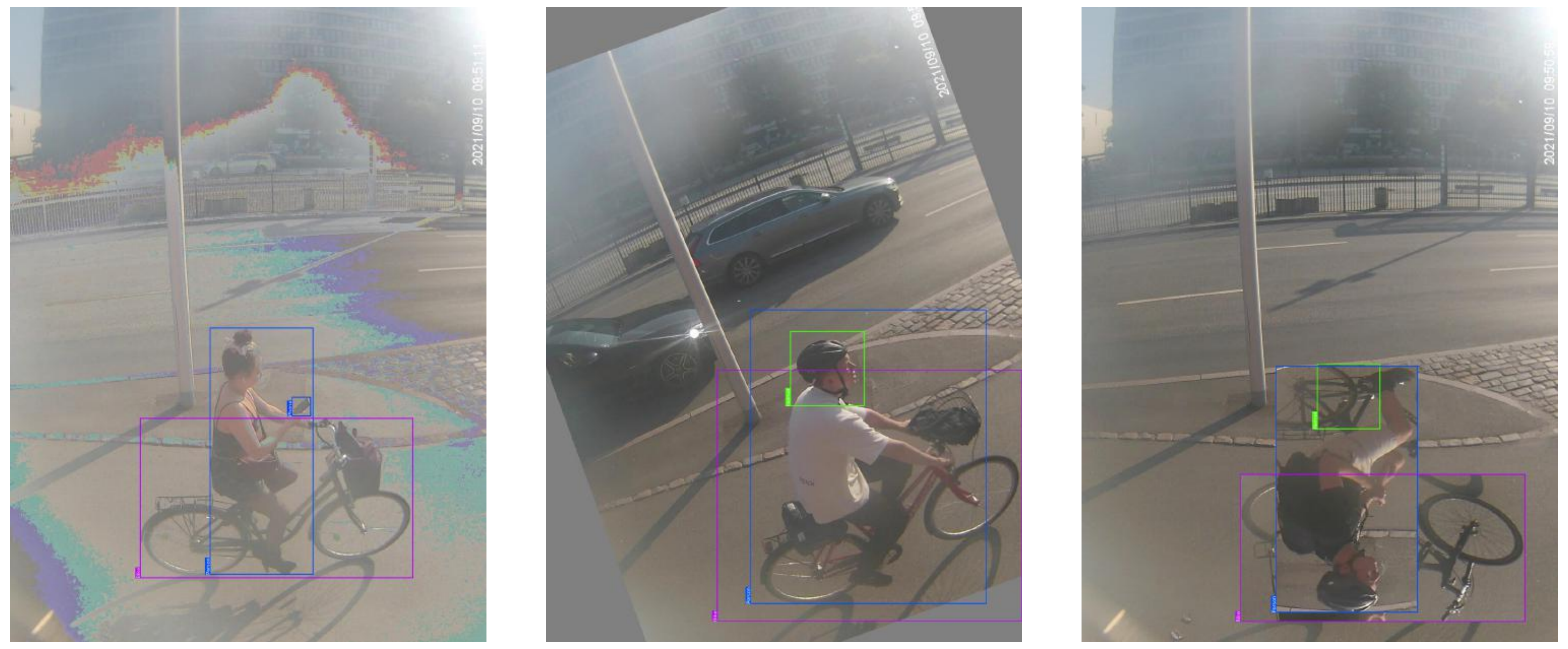
$$\hat{\sigma} = \underset{\sigma \in \mathfrak{S}_N}{\arg\min} \sum_{i}^{N} \mathcal{L}_{\text{match}}(y_i, \hat{y}_{\sigma(i)})$$

with $\mathcal{L}_{\text{match}}(y_i, \hat{y}_{\sigma(i)}) = -\mathbb{1}_{\{c_i \neq \varnothing\}} \hat{p}_{\sigma(i)}(c_i) + \mathbb{1}_{\{c_i \neq \varnothing\}} \mathcal{L}_{\text{box}}(b_i, \hat{b}_{\sigma(i)})$

**2. Calculate the loss for each pair**

$$\mathcal{L}_{\text{Hungarian}}(y, \hat{y}) = \sum_{i=1}^{N} \left[ -\log \hat{p}_{\hat{\sigma}(i)}(c_i) + \mathbb{1}_{\{c_i \neq \varnothing\}} \mathcal{L}_{\text{box}}(b_i, \hat{b}_{\hat{\sigma}}(i)) \right]$$

## Data set



Various proportions of non-annotated frames :

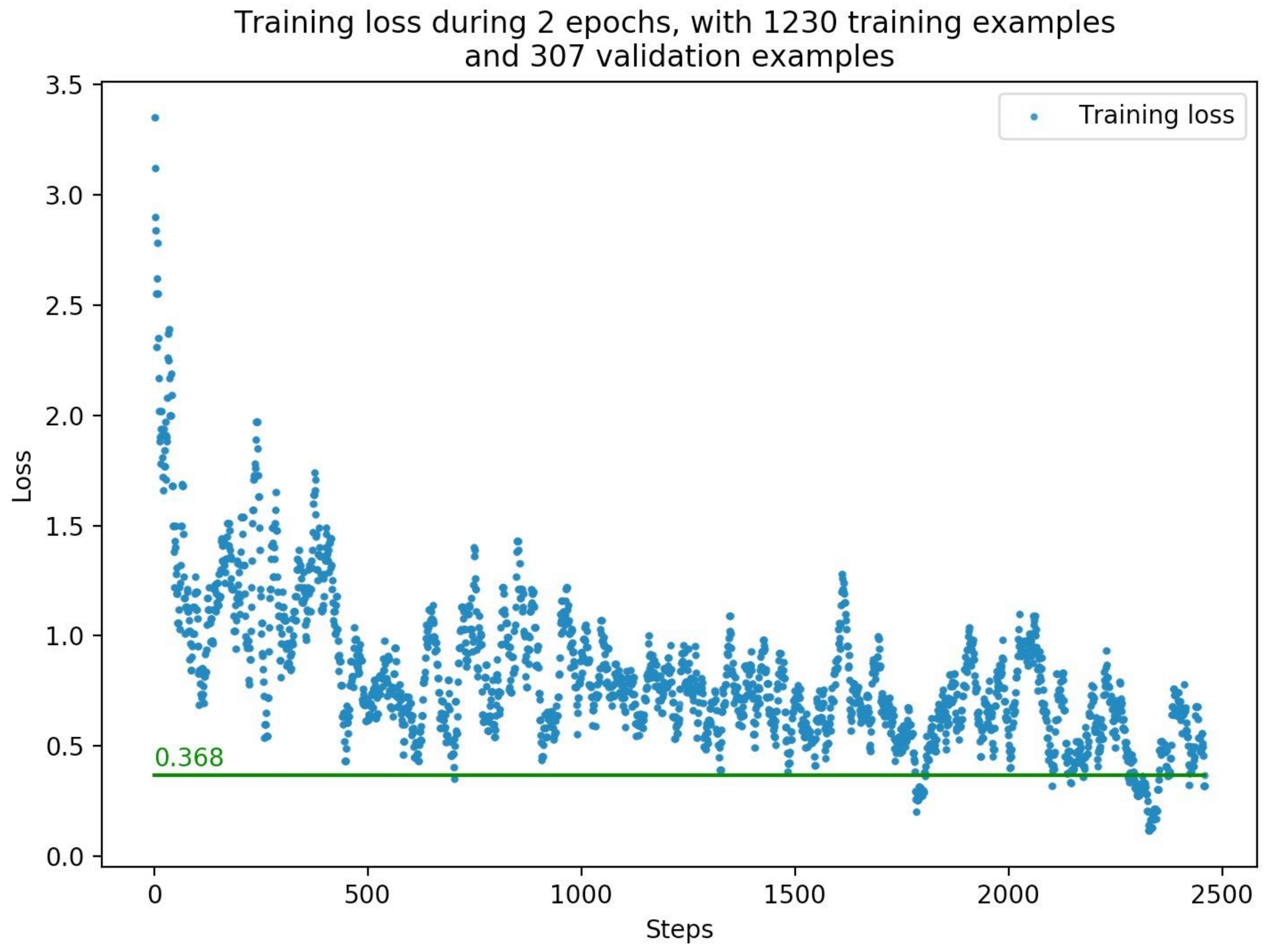
- ◐ 50%
- ◐ 40%
- ◐ 25%

Augmented images with several transformations :

- Solarization
- Rotation of the image
- Rotation only of the bboxes

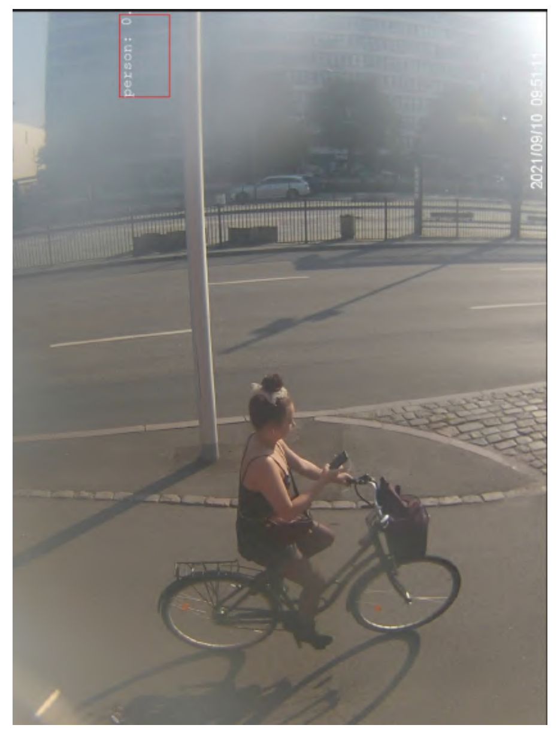
## Results

### Training Loss

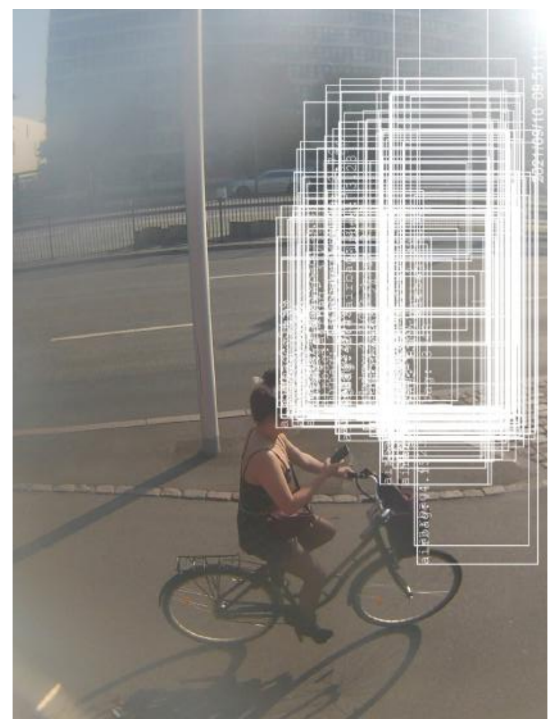


Plot of the training loss during 2 epochs of finetuning. The weights were initialized with those from a pre-trained model.

### Predictions



Before finetuning: 1 person detected in the top-left corner



After finetuning: many airbags detected

For now, the model overpredicts airbags. DETR is known for its poor performance on small objects.

### Evaluation

| | Average Precision (AP)<br>or Average Recall (AR) | |
|---|---|---|
| IoU | Initial | After 1 epoch |
| 0.50:0.95 | AP = 0.001 | AP = 0.004 |
| 0.50 | AP = 0.003 | AP = 0.012 |
| 0.75 | AP = 0.001 | AP = 0.002 |
| 0.50:0.95 | AR = 0.006 | AR = 0.022 |
| 0.50 | AR = 0.013 | AR = 0.042 |
| 0.75 | AR = 0.016 | AR = 0.042 |

COCO Evaluation metrics
After 1 epoch

There is some improvement after 1 epoch. Training more epochs and correcting the model will help obtaining better performance.

## Next steps

- Request access to GPU to train **more epochs**
- Train on **augmented dataset**
- **Compare** performances on datasets with varying proportions of empty frames
- **Crop frames** (remove top third of the image)
- Change **num_queries** (from 100 to 25)
- Try ignoring phone and airbag (too small)

## References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, Sergey Zagoruyko, “End-to-End Object Detection with Transformers”, 28 May 2020, https://arxiv.org/abs/2005.12872

[2] “DETR: End-to-End Object Detection with Transformers (Paper Explained)”, Yannic Kilcher, https://www.youtube.com/watch?v=T35ba_VXkMY

[3] “Recommendations for training Detr on custom dataset?”, Facebook Research’s GitHub, 28 May 2020 https://github.com/facebookresearch/detr/issues/9

[4] Pytorch Lightning documentation : https://pytorch-lightning.readthedocs.io/en/stable/common/lightning_module.html?highlight=freeze

[5] Barret Zoph∗ , Ekin D. Cubuk∗ , Golnaz Ghiasi, Tsung-Yi Lin, Jonathon Shlens, Quoc V. Le Google Research, Brain Team, "Learning Data Augmentation Strategies for Object Detection", 26 Jun 2019, https://arxiv.org/pdf/1906.11172v1