# ICCV SSLAD Workshop Competition

# SSLAD workshop

SSLAD = Self-Supervised Learning for next generation industry level Autonomous Driving

- Mainly focused on autonomous driving.
- Continual learning as a major topic:
  - Self-supervised learning techniques
  - Life-long/incremental visual recognition methods
  - Weakly supervised learning algorithms
  - One/few/zero shot learning for perception tasks in self-driving
  - Domain adaptation

# SSLAD Challenge

3 tracks:

- Track 1: 2D object detection
- Track 2: 3D object detection
- Track 3: Continual learning
  - Track 3a: continual classification of 2D objects
  - Track 3b: continual object detection

# SSLAD Challenge track 3a

- Data from the Soda10M dataset
  - Data acquired from cars travelling in China, from the perspective of the driver
- Six classes: pedestrian, cyclist, car, truck, tram, tricycle
  - Highly class-imbalance data: many more cars than tricycles.
- Data presented to the model as a stream:
  - No shuffling of the data allowed
  - Train data from 3 days (3 days and 3 night), presented sequentially to the model.
  - A maximum of 10 images per minibatch
- Evaluation using Mean Class Accuracy (MCA) over 6 test points during training
  - Model must be evaluated at every moment without any further computation (no GDumb approaches)
  - Training set divided into 6 experiences, evaluation after every experience

# Soda10M dataset

- Large-scale 2D object recognition dataset for autonomous driving
- Contains 10M unlabeled images and 20k fully annotated images
  - In this track only the fully annotated set is used
- Images acquired by taxi drives with high resolution (1080p+) every 10 seconds.
- Horizon at the center of the image, occlusion from the interior of the car no more than 15% of the image
- Images annotated using the COCO format.

In the competition the division between train, validation and test set is “custom” and doesn’t follow the Soda10M division.

# Soda10M dataset

# Details of the Track 3a dataset

- Images of objects are cut out of the scene (suppose a perfect bounding box)
- 22,249 training images, 6,305 validation images.
- Rescaled to 64x64 pixels.
- Training images divided into 6 experiences in the following way:
  - exp1 -> 5,157 images, all the classes
  - exp2 -> 1,154 images, only classes 3, 4, and 5 (car, truck, tram)
  - exp3 -> 6,742 images, all the classes
  - exp4 -> 2,560 images, only classes 3, 4, and 5
  - exp5 -> 4,517 images, all classes
  - exp6 -> 2,119 images, all classes

# Details of the Track 3a dataset

Classes are highly imbalanced!

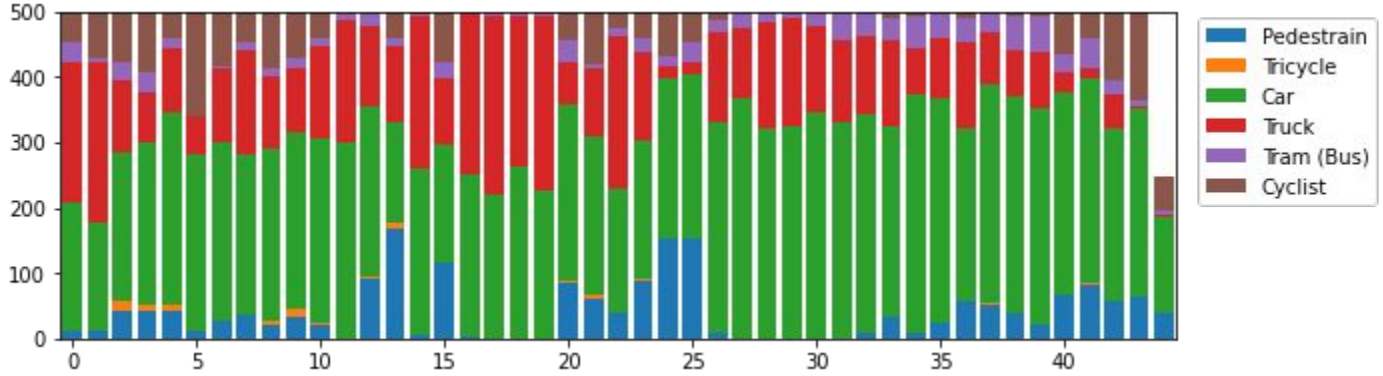
- Details of the classes in the **training** experiences
  - exp1 -> {pedestrian: 300, cyclist: 821, car: 2,498, truck: 1,324, tram: 157, tricycle: 57}
  - exp2 -> {car: 721, truck: 400, tram: 33}
  - exp3 -> {pedestrian: 972, cyclist:442, car: 3,066, truck: 2,051, tram: 190, tricycle: 21}
  - exp4 -> {car: 1,722, truck: 744, tram: 94}
  - exp5 -> {pedestrian: 259, cyclist: 55, car: 2,939, truck: 895, tram: 367, tricycle: 2}
  - exp6 -> {pedestrian: 303, cyclist: 397, car: 1,235, truck: 77, tram: 103, tricycle: 1}
- Total: {pedestrian: 1,834 - cyclist: 1,715 - car: 12,181 - truck: 5,491 - tram: 944  - tricycle: 81}

# Details of the Track 3a dataset

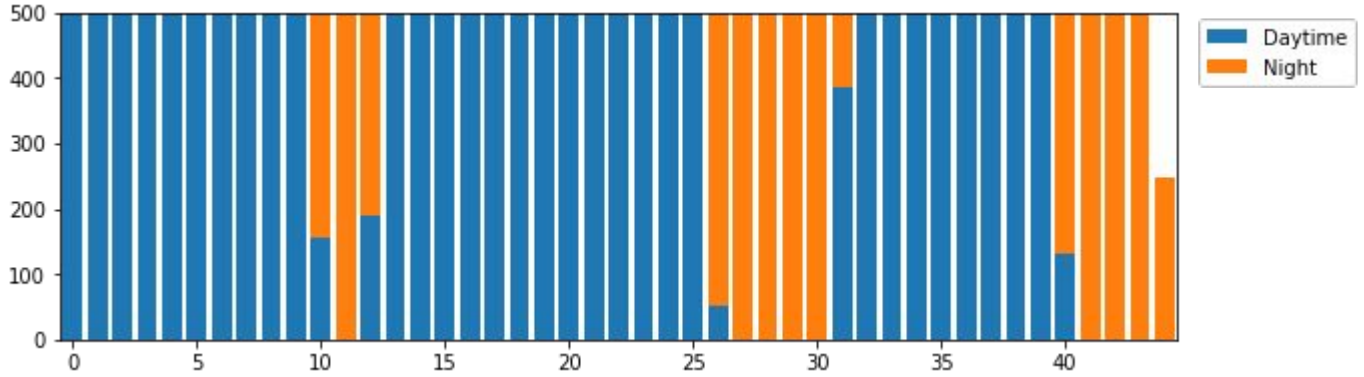
Classes are highly imbalanced!

- Details of the classes in the **validation** experiences
  - exp1 -> {pedestrian: 364, cyclist: 180, car: 3,429, truck: 1,677, tram: 251, tricycle: 5}
  - exp2 -> {cyclist: 188, tricycle: 201}
- Total: {pedestrian: 364 - cyclist: 368 - car: 3,429 - truck: 1,677 - tram: 251  - tricycle: 206}

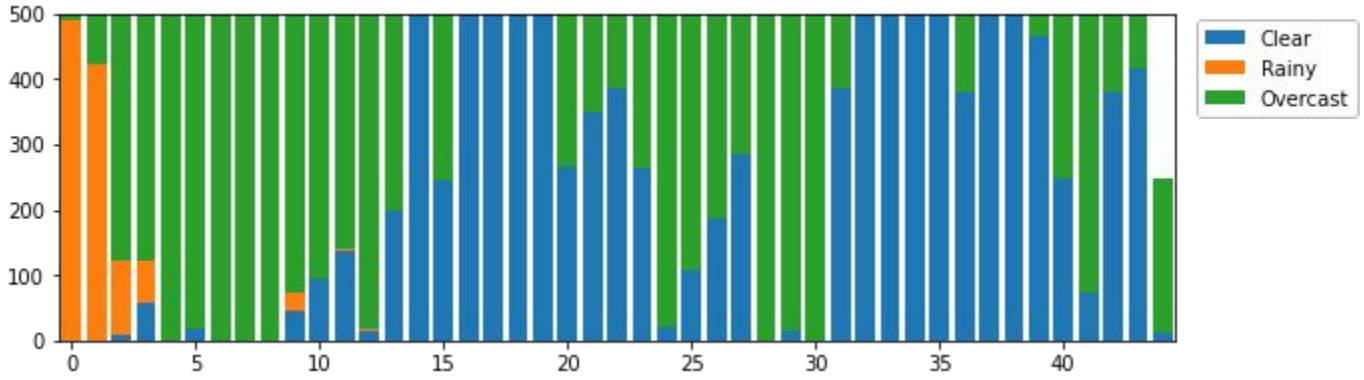
# Data Imbalances



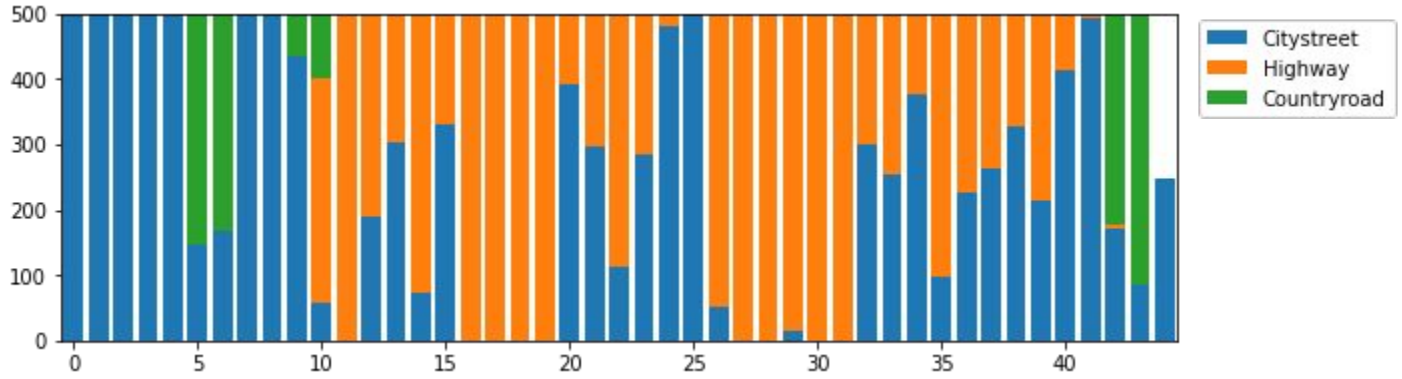
# Data Imbalances



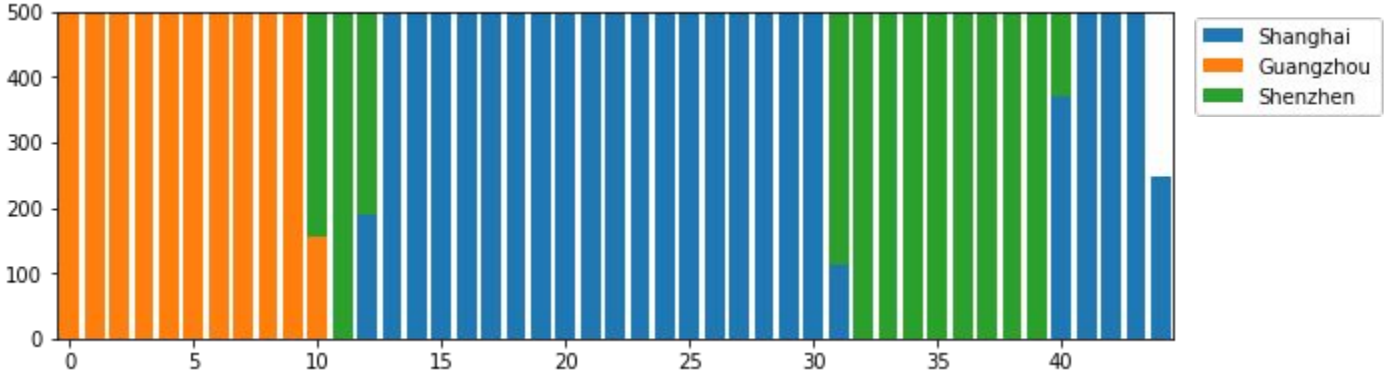
# Data Imbalances



# Data Imbalances



# Data Imbalances



# Example: the first 40 images

- Images are not shuffled, so the order is predetermined.
- Each “session” is composed of a “video” of images acquired by a certain driver, at 0.1fps.
- All the images of a session are used successively.
  - High probability of having the same subjects (same cars, same trucks…)
  - Data is highly non-i.i.d.
- Not defined the ordering of the subjects of the same image.

# Example: the first 40 images

Classes of the first 40 images (in order):

4, 4, 4, 4, 4, 4, 4, 3, 4, 3, 3, 3, 3, 4, 4, 1, 1, 3, 3, 3, 4, 4, 1, 3, 4, 4, 3, 3, 3, 3, 4, 4, 4, 3, 3, 4, 3, 4, 4, 1

1 = pedestrian

3 = car

4 = truck

# Example: the first 6 images




# Example: other images






# Evaluation

- Mean Class Accuracy (MCA)

$$\mathrm{MCA}_c = \frac{1}{6}\sum_{e=1}^{6} \mathrm{acc}_c^e$$

- Average Mean Class Accuracy (AMCA)

$$\mathrm{AMCA} = \frac{1}{6}\sum_{c} \mathrm{MCA}_c, \quad c = \{\text{pedestrian, cyclist, car, truck, tram, tricycle}\}$$

# Code

- All the code is based on Avalanche (apart some image preprocessing stuff).
- The data loading, evaluation, and  training flow is already implemented and must not be touched.
- We have to implement an Avalanche's plugin that contains our CL strategy.
- Apart from that we could change:
  - The model
  - The optimizer (and optimizer scheduler)
  - The loss
  - The batch size ( < 10 )
  - Any other Avalanche (or custom) plugin (the base strategy is the Naive strategy).

# General rules

- Maximum number of parameters of the model: 25M (105% of ResNet-50)
- Capacity of any sort of memory: 1,000 objects
- Only 1 epochs per experience (images can be viewed only one time)
- Pretraining on ImageNet allowed, but no other data can be used during training
    - Pretraining on other datasets?
- Model must be able to switch between training and eval mode seamlessly (no extra computation or extra step required).

# Submission

- A submission is composed of 2 files (a .txt file and a .log file) produced by the provided code.
  - Submission consists of a .zip archive containing those two files.
- The source code is not required.
- The top-3 teams must send a technical report about their method and their algorithm.
- Organizers can, at any time, disqualify a team for any reason or any violation of the rules.

# Preliminary tests

- A naive strategy already obtains an AMCA of 43.83%.
- Mean Final accuracy (last evaluation)
  - Pedestrian: 78.29%
  - Cyclist: 70.92%
  - Car: 95.95%
  - Truck: 47.23%
  - Tram: 63.98%
  - Tricycle: 0.00%
- Best method so far: mrifkikurniawan AMCA of 57.88%