Lecture 4 | YOLO

Secure Autonomous Systems

CSCI 6907/3907 86

Spring 2024

Prof. Sibin Mohan

https://bit.ly/secureauto-spring24

Cameras | Sliding Window Algorithm

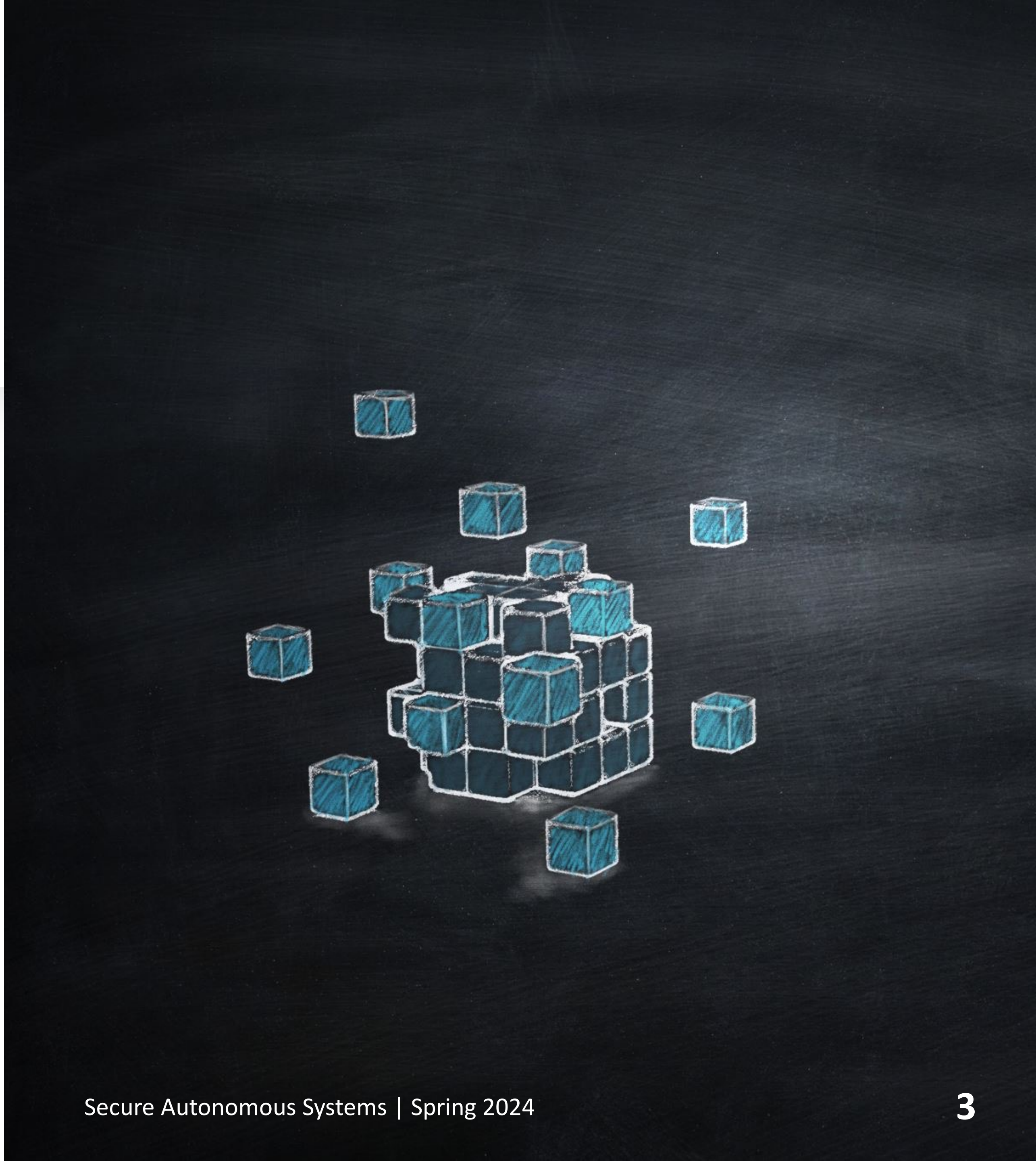
What about objects much larger or much smaller than window size?

YOLO!

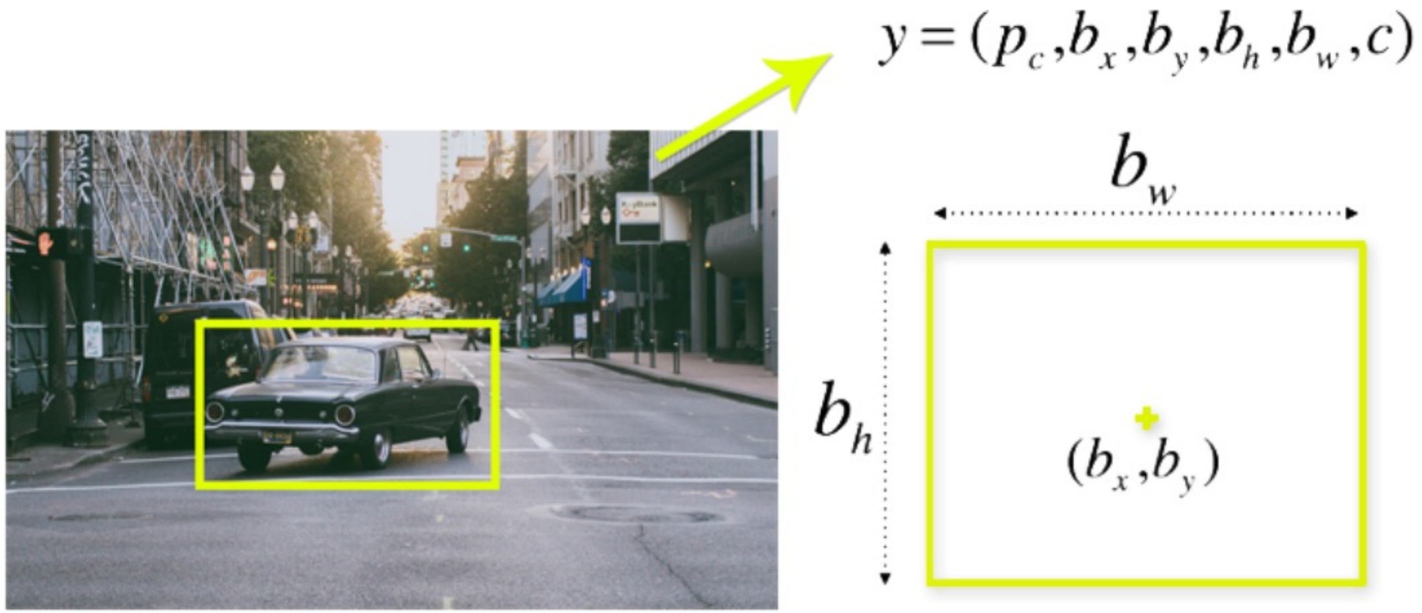
YOLO

- “**y**ou **o**nly **l**ook **o**nce”
- predict classes+bounding boxes
- in **one** run of the algorithm!

- but, what is being done, in practice?



Bounding Boxes



$$y = (p_c, b_x, b_y, b_h, b_w, c)$$

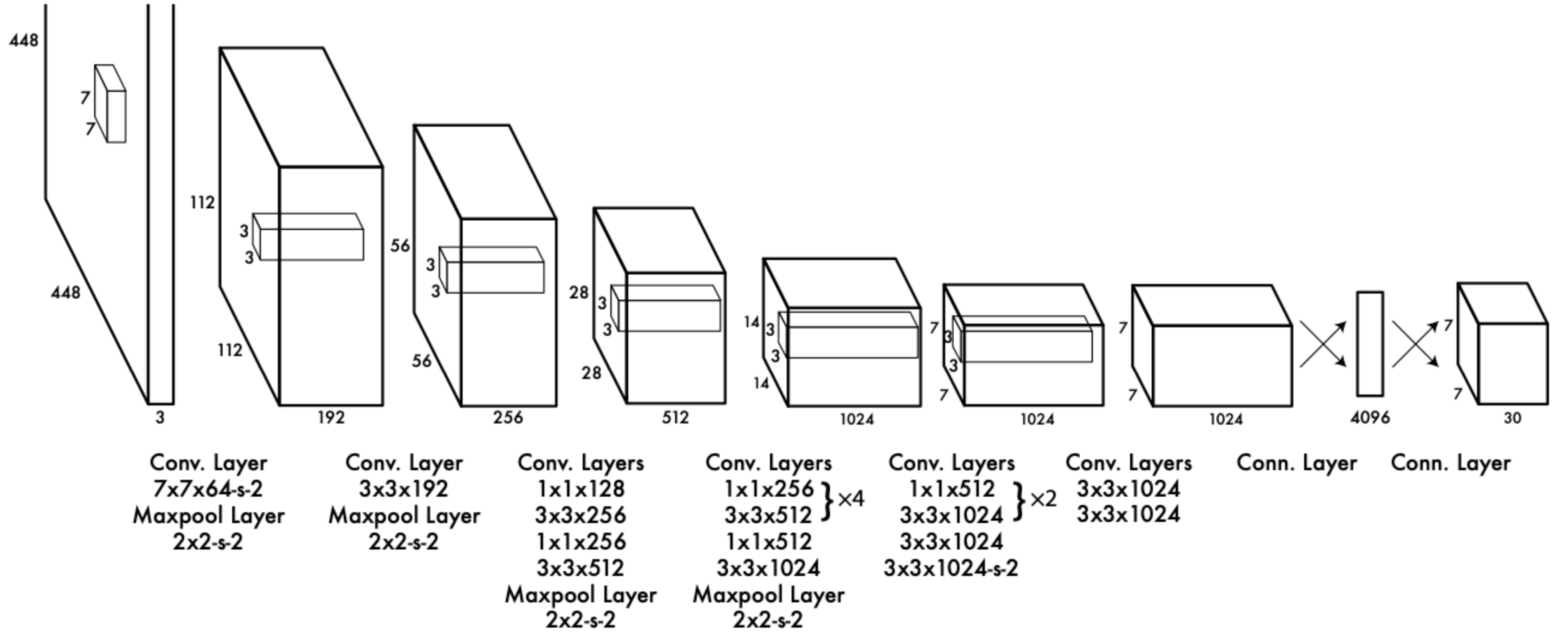
note: we are **not** trying to find “interesting regions” or “objects” in entire image
instead: we split our image into “**cells**”

Cameras | YOLO [contd.]

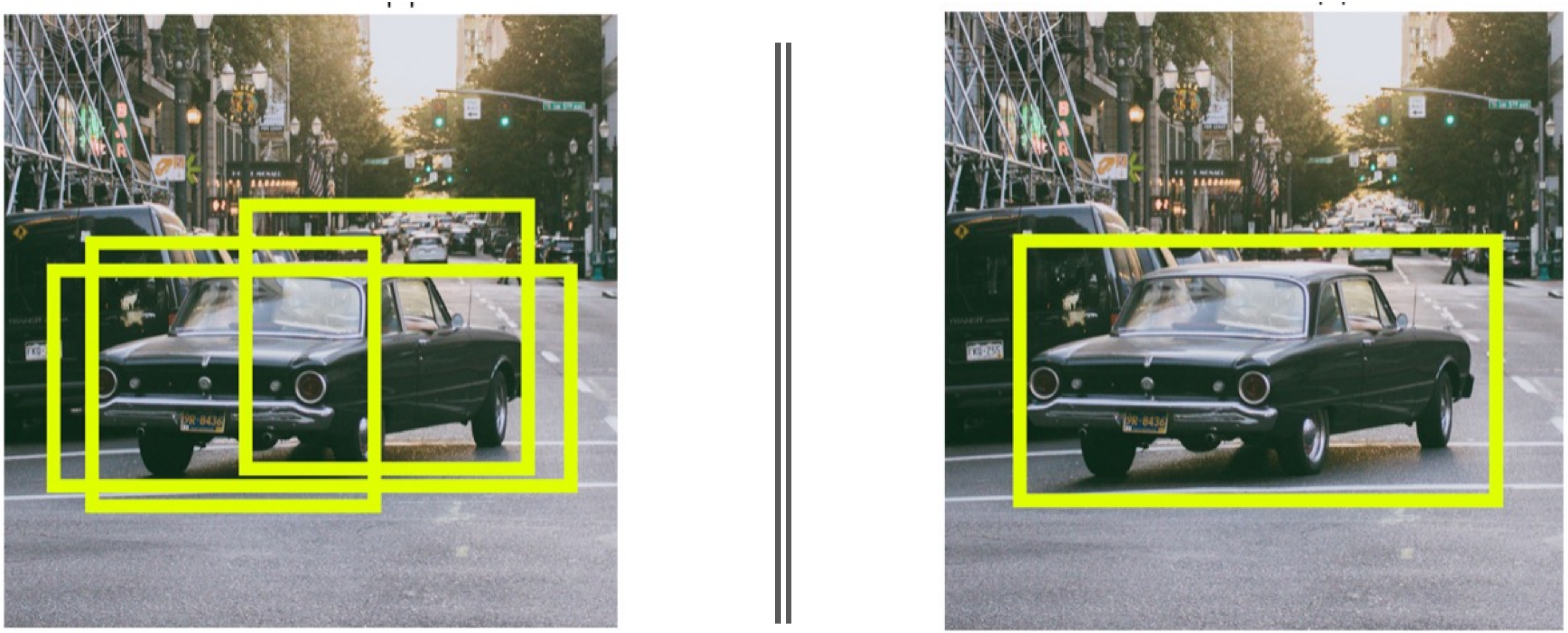
preprocessed image
(608, 608, 3)

each cell → responsible for predicting **5 bounding boxes**

YOLO Architecture



24 convolutional layers, **2** fully-connected layers

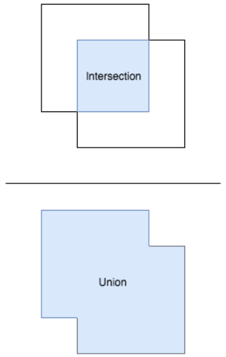


YOLO | Image Localization

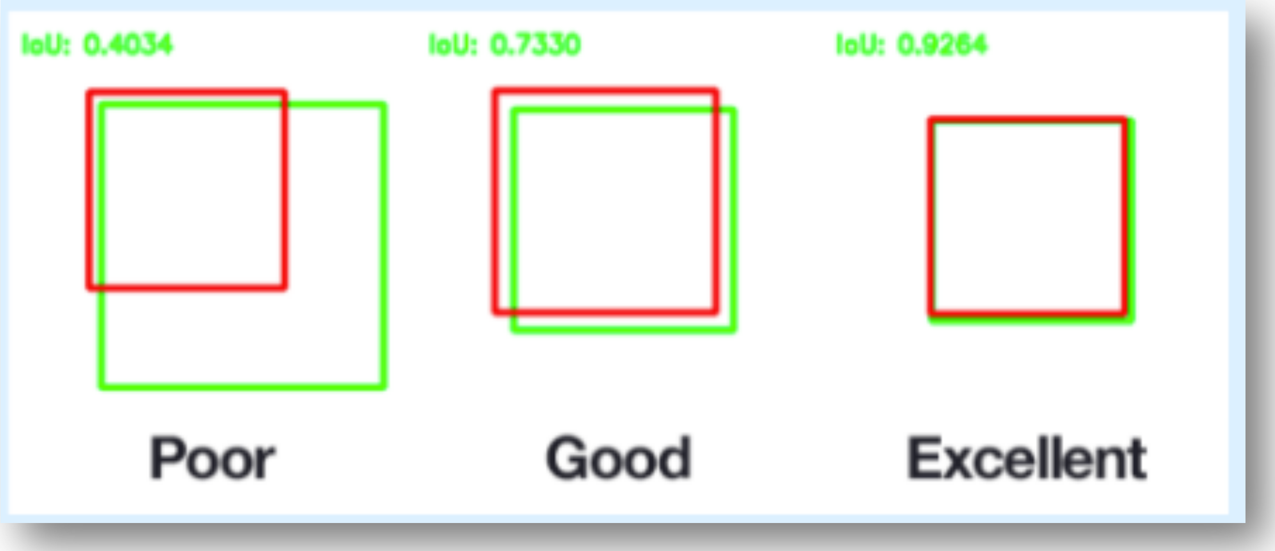
YOLO | IoUs

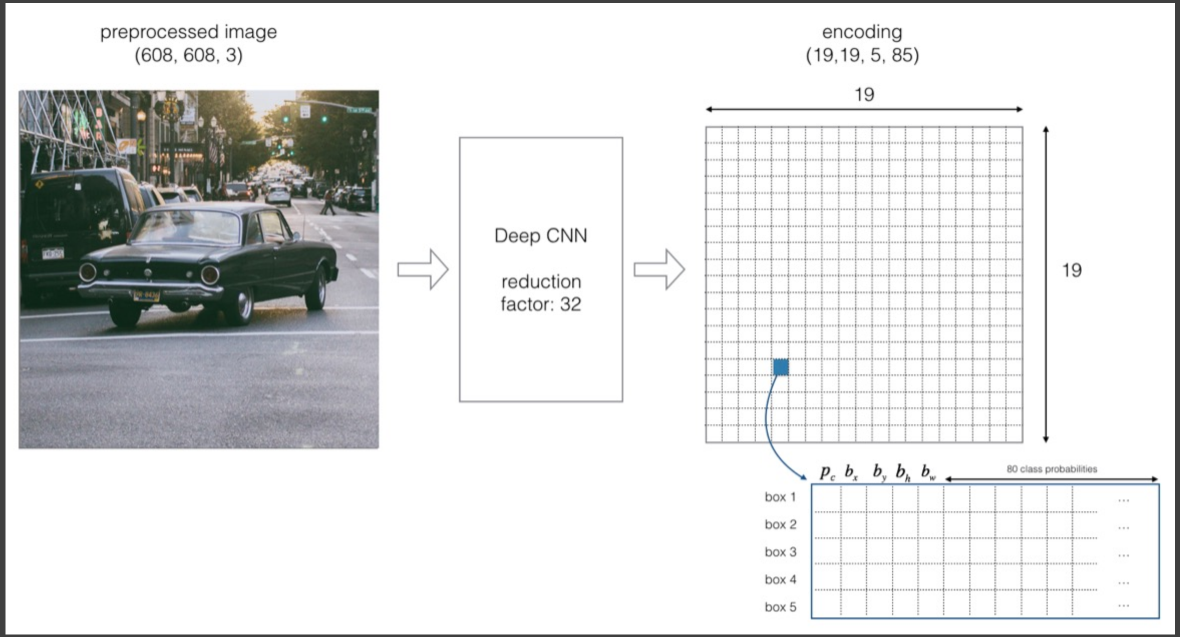
- During training ➔ compare CNN bounding box to **actual** ones
- Cost function, “**intersection over union**” (**IoU**)

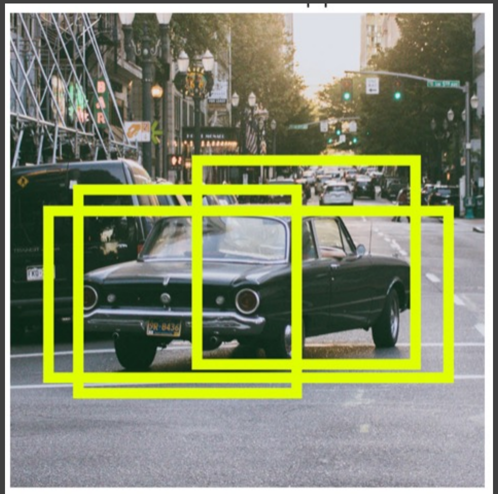
$$\boldsymbol{IoU} = \frac{area\ of\ \boldsymbol{intersection}\ of\ bounding\ boxes}{area\ of\ \boldsymbol{union}\ of\ bounding\ boxes}$$

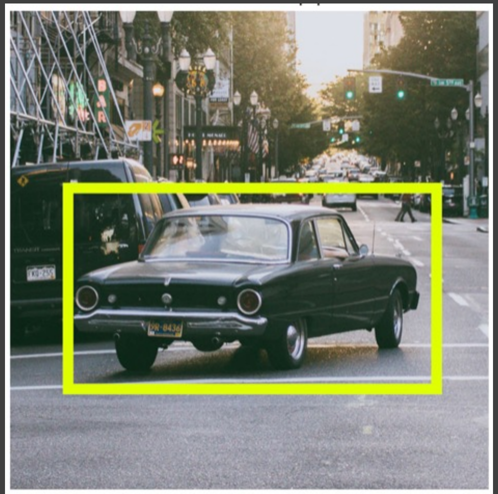


- If IoU is closer to **1** ➔ better the bounding box







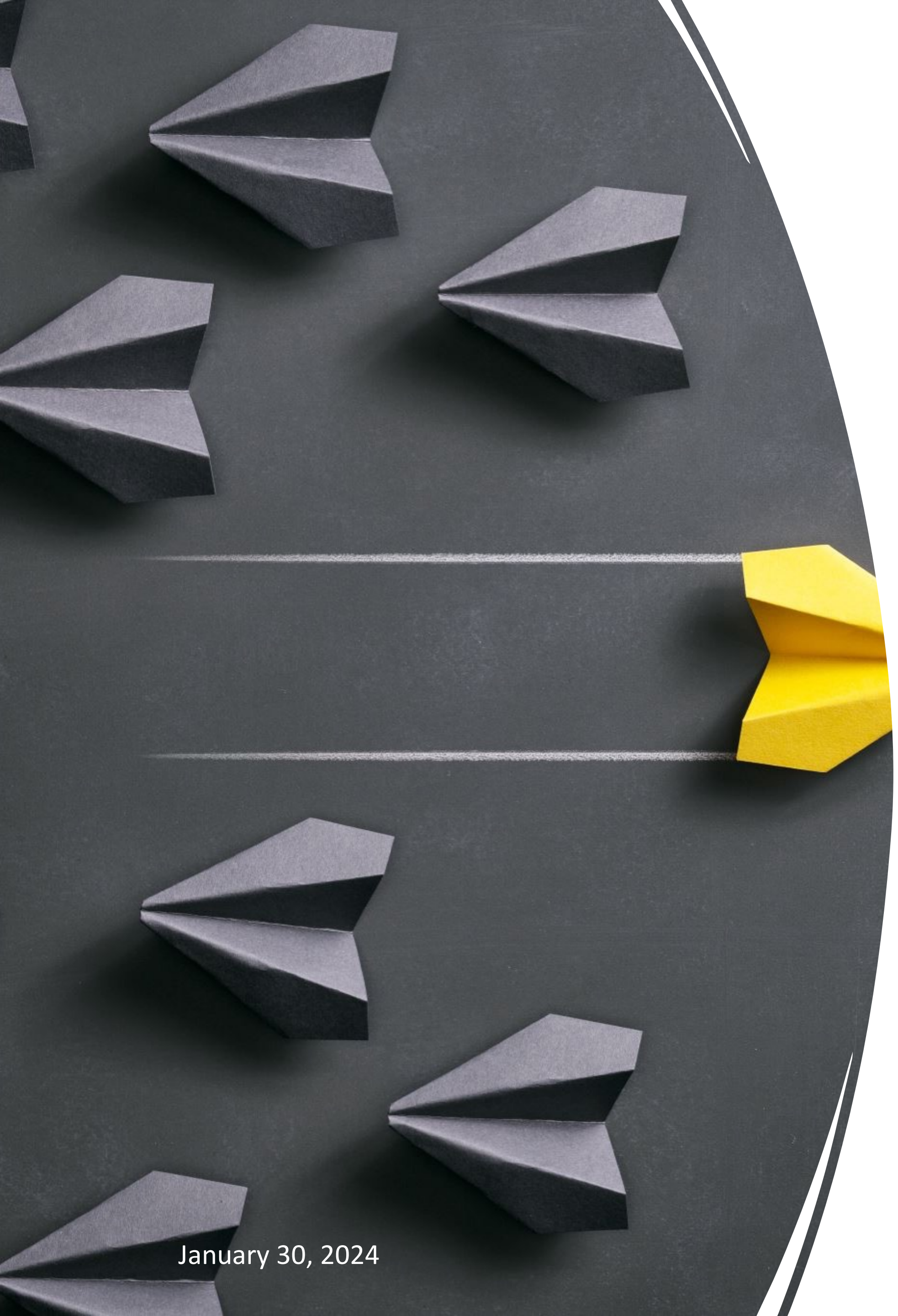


YOLO | Non-Max Suppression

- Majority of the cells won’t have bounding boxes
- Remove boxes with
  - low object probability
  - highest shared area
- **non-max suppression**
  - discard bounding boxes with probability less than threshold
    i.e. p < 0.5 or 0.6
  - take box with highest prediction value
  - discard/suppress boxes with IoU > threshold with that box
    i.e. 0.5 or 0.6
- suppress boxes that don’t have maximum probability

Attacking Object Detectors?

- training – misclassify objects
- attack the camera – show false image
- attack the objects/environment

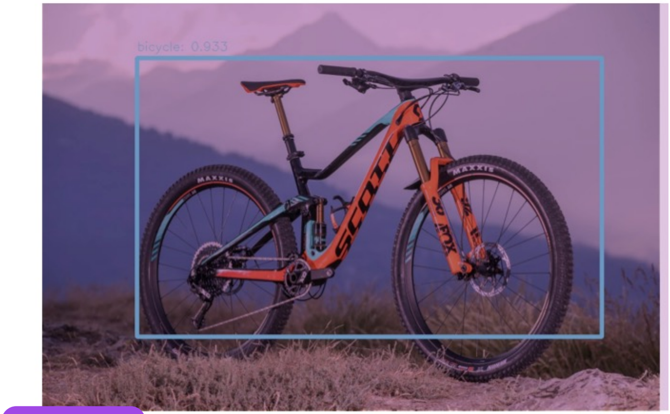


Attacking Object Detectors?

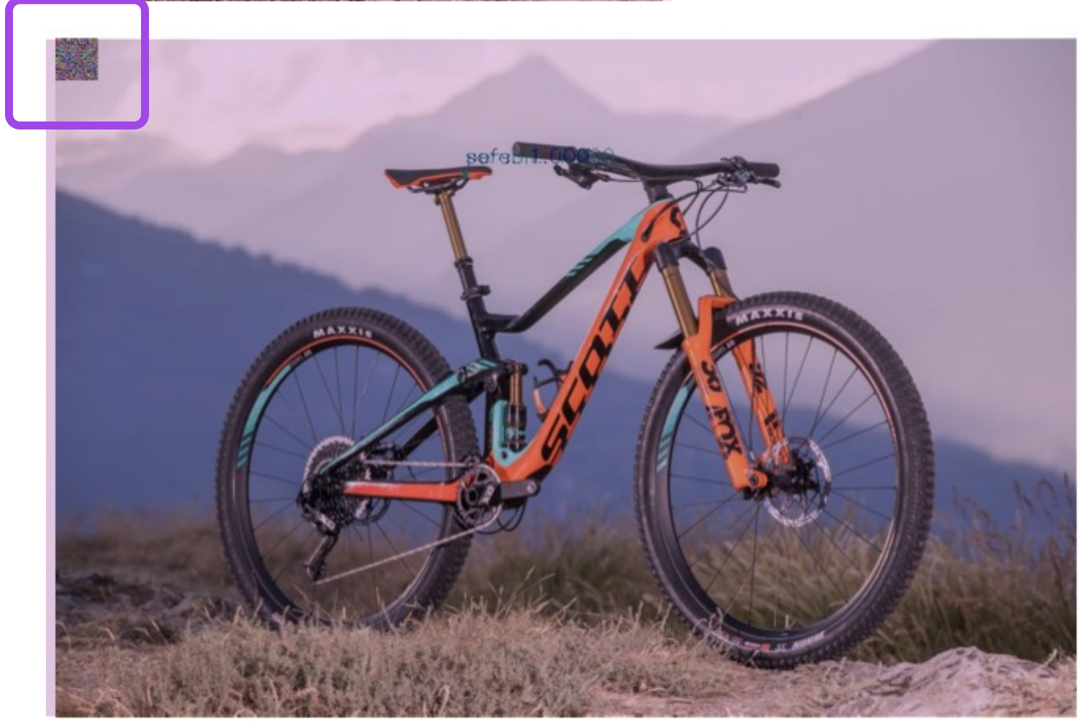
- Falsify the **training** set
  - Larger impact
  - Harder to do – less public access
- **Modify objects** being detected
  - Add paint/tape/appendages to cars to that it presents differently
- Attack the **inputs**
  - Add stickers to objects
  - Add extraneous pixels/data to the camera inputs

Attacking Object Detectors | Example

- Maximize loss of CNN classifier
- Maximize loss of object detector

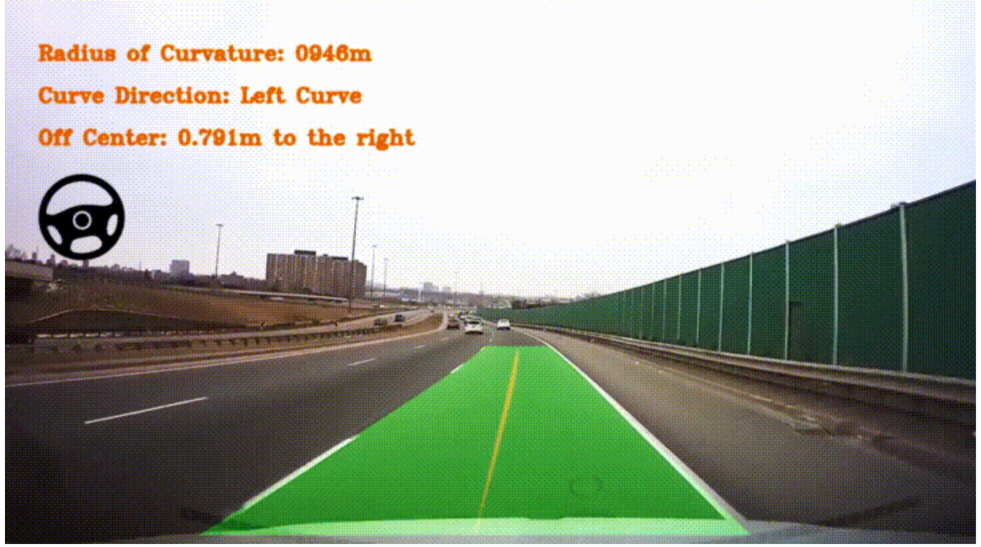


40x40 patch



Cameras

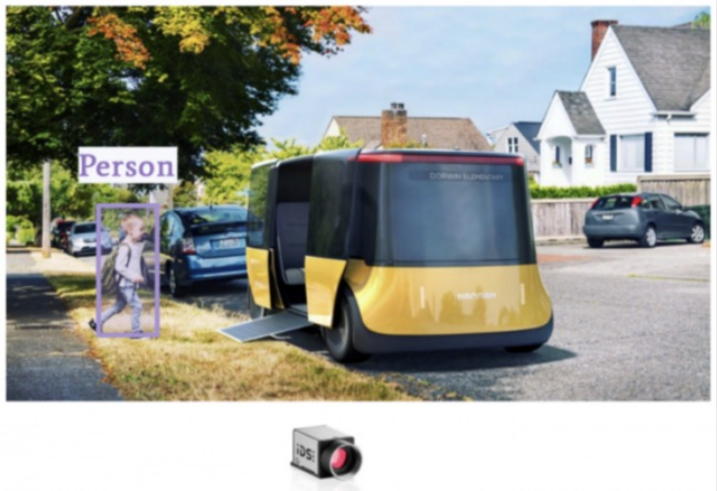
- Additional cameras
  - Lane following
  - Traffic signal monitoring

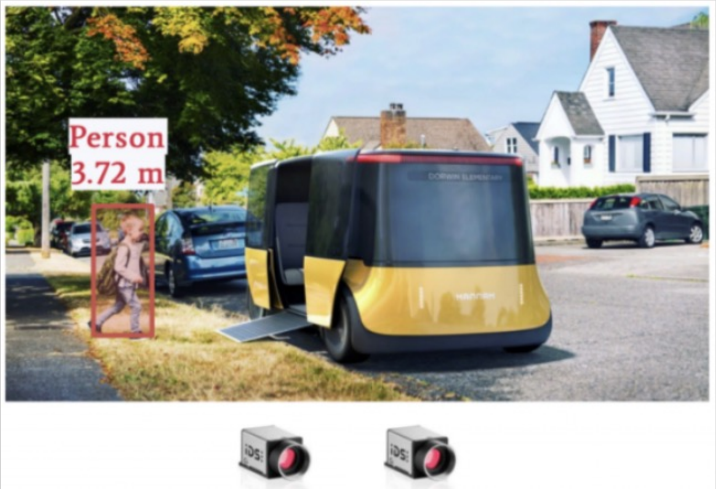

Radius of Curvature: 0946m
Curve Direction: Left Curve
Off Center: 0.791m to the right

Stereo Vision

- Problem with regular cameras+YOLO is **2D vision**
- “Fuse” camera data with LiDAR ➔ expensive

- Align **two cameras** and use **geometry**
- **Pseudo-LiDAR**





Stereo Vision

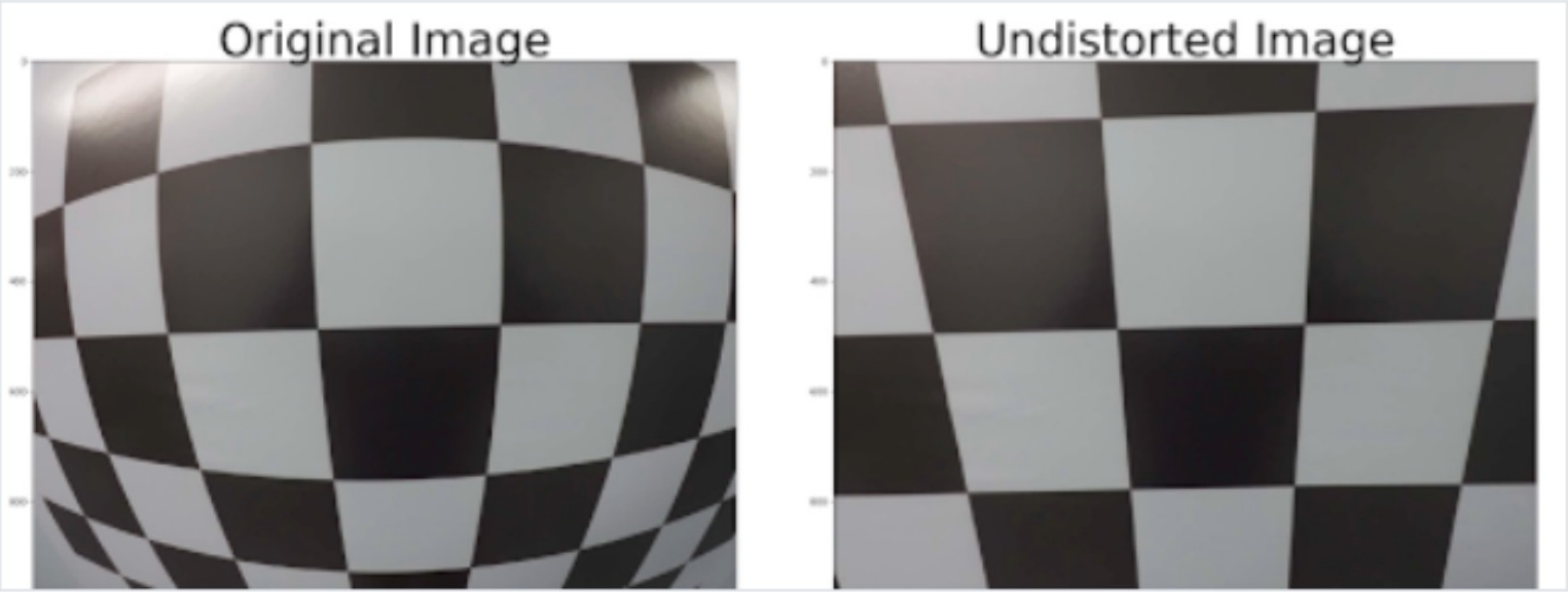
- Retrieve distance of an object using **two cameras** and **triangulation**

Steps

- Stereo calibration
- Epipolar geometry
- Disparity mapping
- Depth mapping
- Obstacle detection estimation

Stereo Vision | Calibration

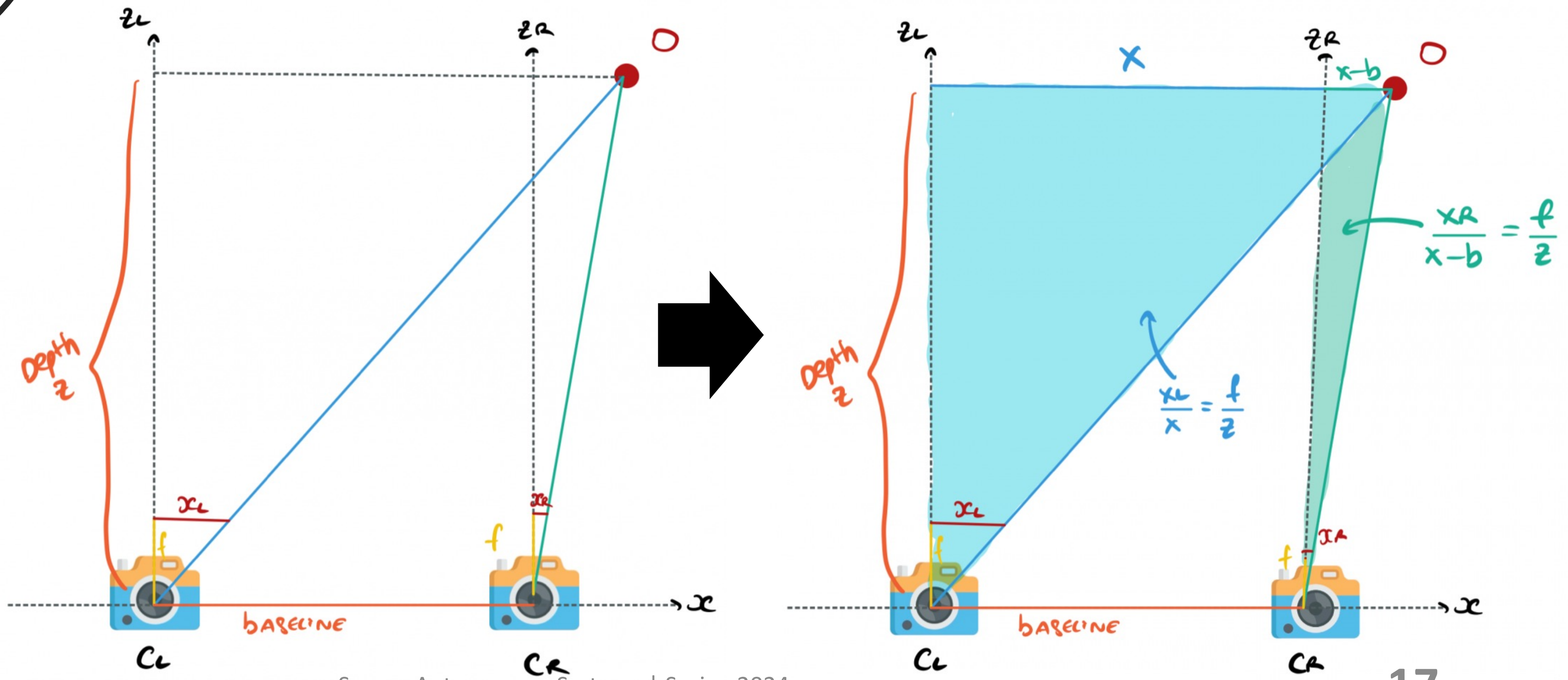
Create undistorted images from original camera ones



Stereo Vision | Calibration + Epipolar Geometry

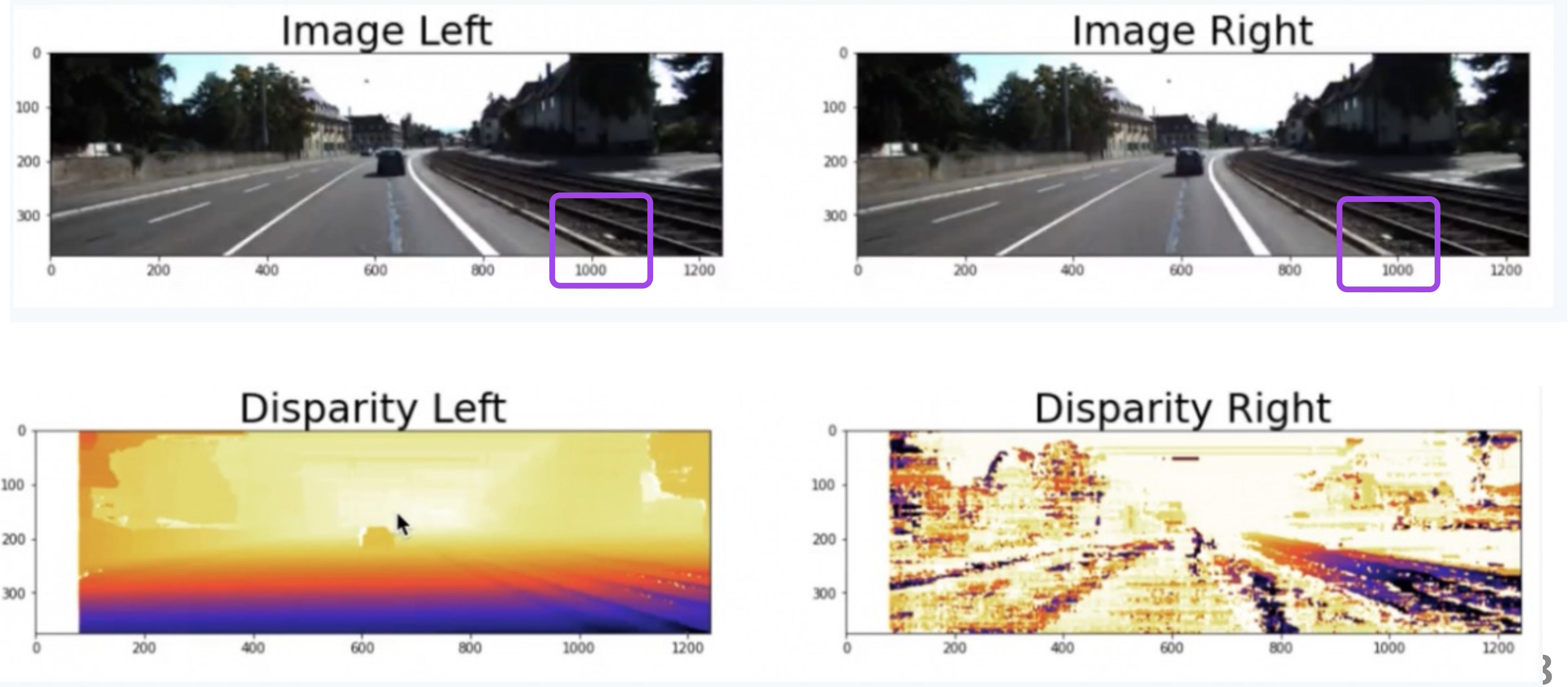
$$Z = \frac{f \cdot b}{x_L - x_R} = \frac{f \cdot b}{d}$$

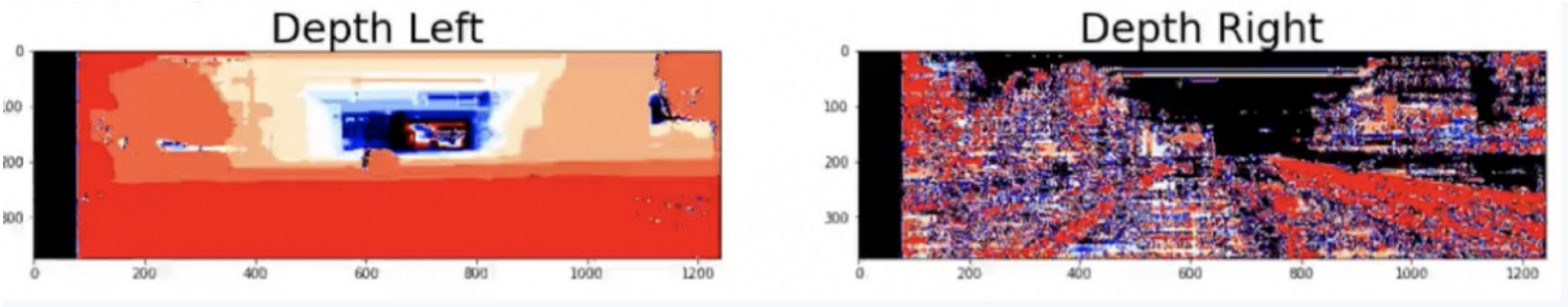
Geometry Calculations



Stereo Vision | Disparity Mapping

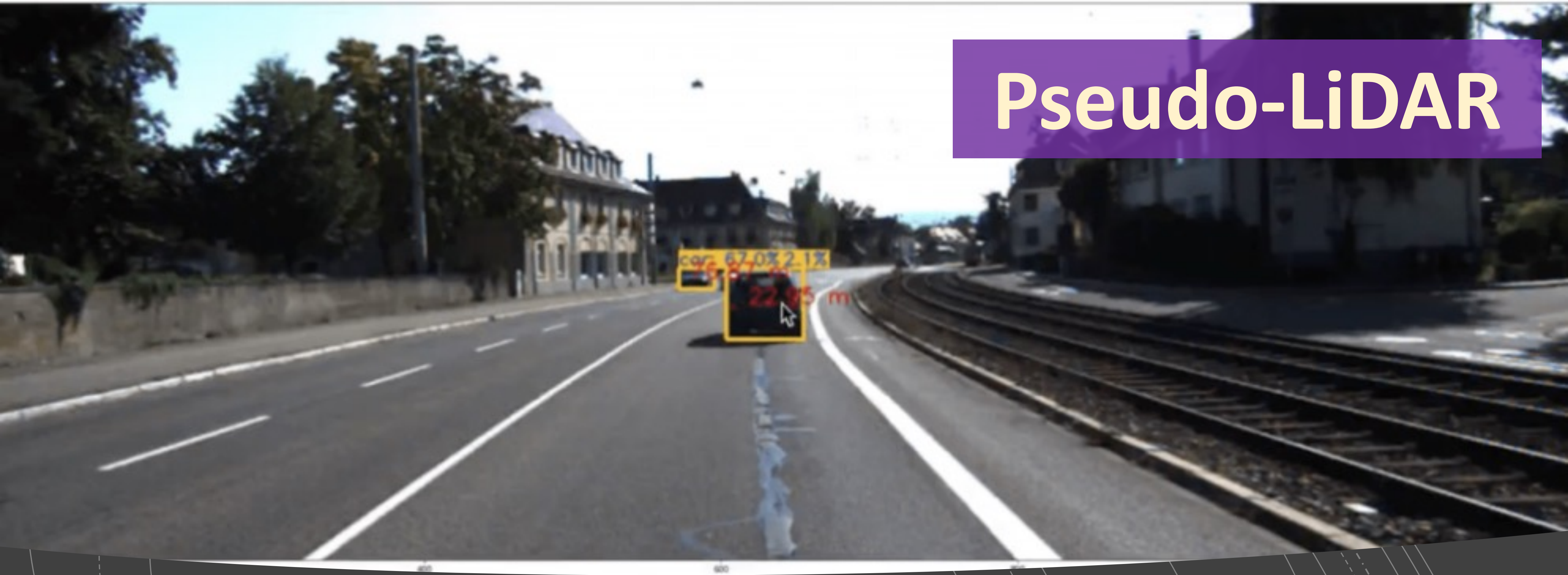
- Difference in image location of same 3D point from 2 camera angles





Stereo Vision | Depth Map

- **Distance of each pixel** in an image
  - Using other image+disparity map



Pseudo-LiDAR

Stereo Vision | Estimate Depth

- Using depth map, combine with YOLO
- E.g. run YOLO on left image and then use depth map
- In bounding box from YOLO, closest point can be taken

References

- Computer Vision/YOLO

https://medium.com/@albertlai631/how-do-self-driving-cars-see-13054aee2503

https://www.kdnuggets.com/2018/09/object-detection-image-classification-yolo.html

- Attack on YOLO paper

https://arxiv.org/pdf/1806.02299.pdf

- Stereo Vision/Pseudo LiDAR

https://www.thinkautonomous.ai/blog/?p=pseudo-lidar-stereo-vision-for-self-driving-cars