Football Analysis Project

Marijan Sorić
(Žan Stanonik)

NTNU – Computer Vision and Deep Learning – TDT4265

Final Project

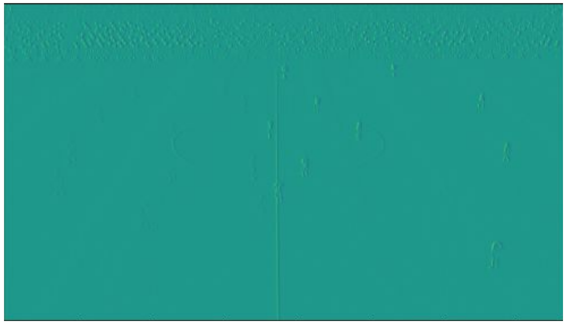
Presentation

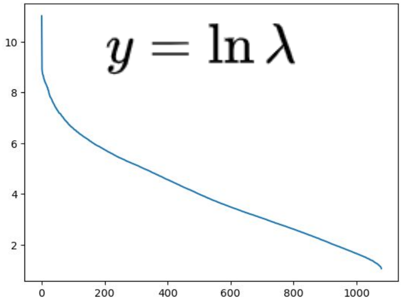
- Object Detection
- Tracking
- Keypoint Detection
- Data Analysis

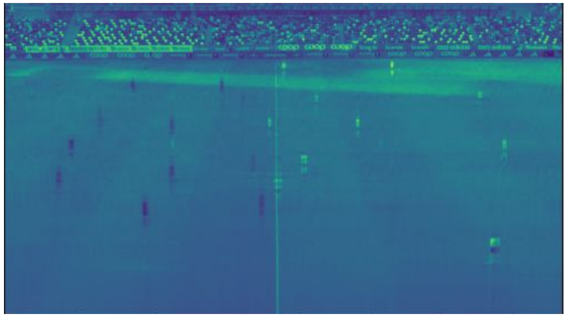
Image height: 1080

Image width: 1920

EDA *Images*



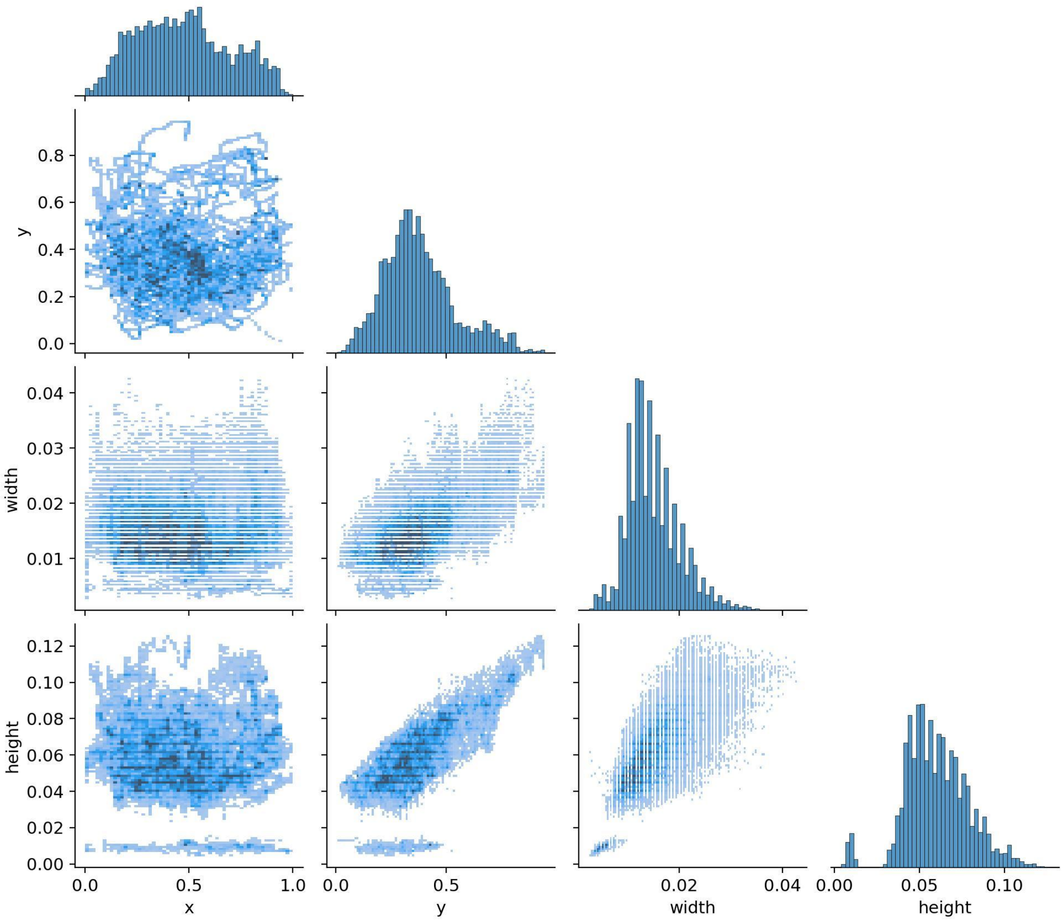




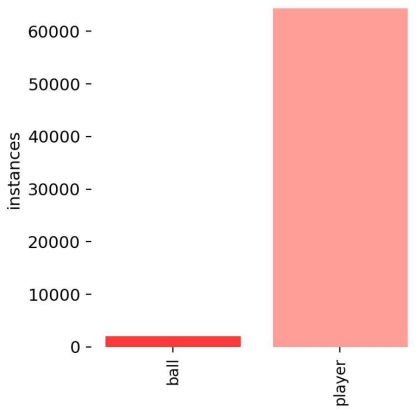
Training set: 3 604 images

- Mean Image
- Convolution with kernel (*left sobel, top sobel)*
- SVD
- Low Rank Decomposition
- Max Pooling Compression

EDA *Classification*



Labels correlation



You Only Look Once

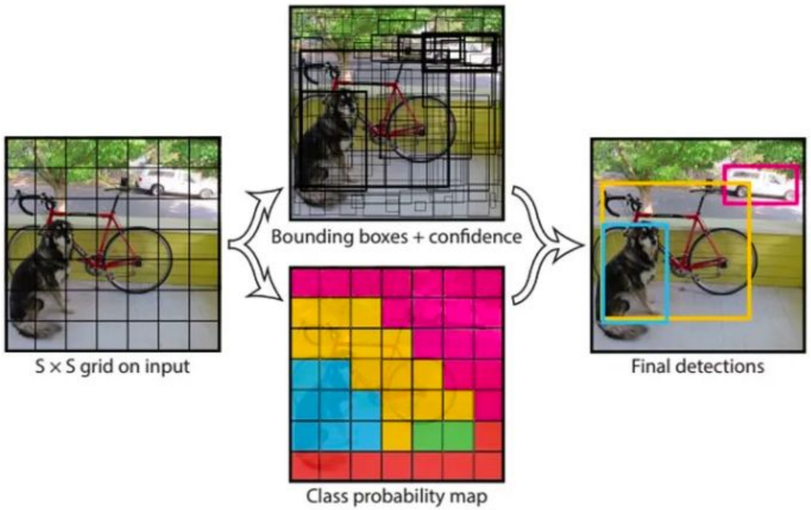


Figure 2: The Model. Our system models detection as a regression problem. It divides the image into an $S \times S$ grid and for each grid cell predicts B bounding boxes, confidence for those boxes, and C class probabilities. These predictions are encoded as an $S \times S \times (B * 5 + C)$ tensor.

Overlay grid cell produces

- Set of bounding boxes with confidence score

$$(x, y, \sqrt{w}, \sqrt{h}, C)$$

- Class probability map

$$(\mathbb{P}[\text{class}_i])_i$$

You Only Look Once

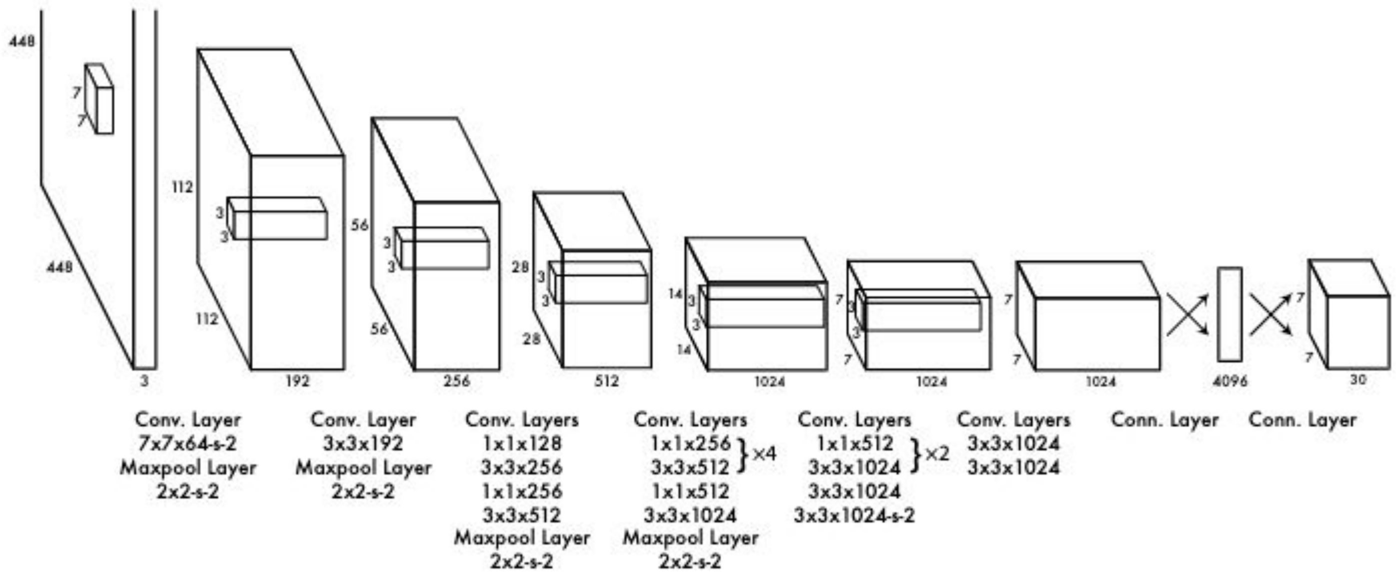


Figure 3: The Architecture. Our detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating 1×1 convolutional layers reduce the features space from preceding layers. We pretrain the convolutional layers on the ImageNet classification task at half the resolution (224×224 input image) and then double the resolution for detection.

You Only Look Once

Loss function

$$\lambda_{\textbf{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]$$

$$+ \lambda_{\textbf{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$$

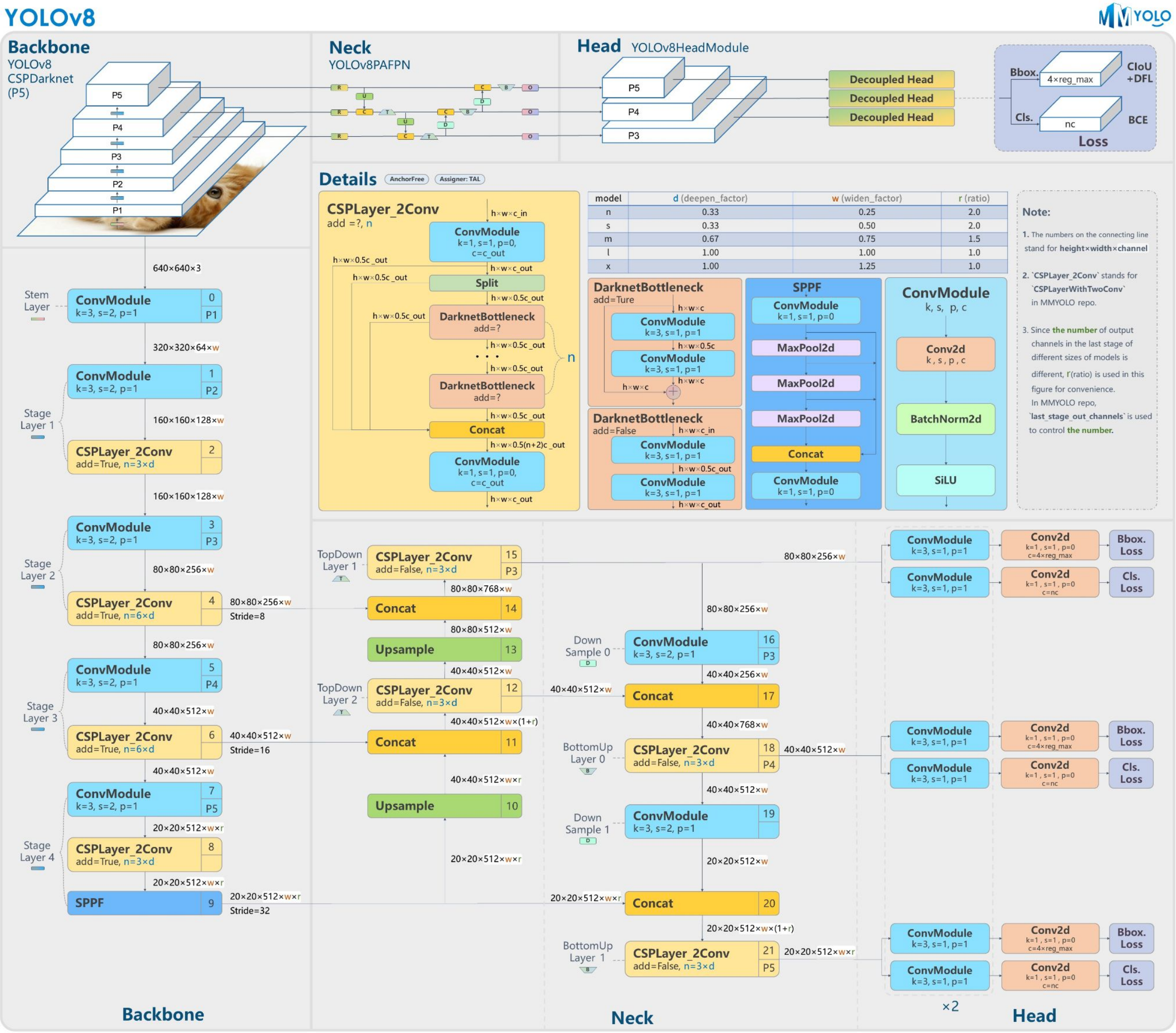
$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2$$

$$+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2$$

$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2$$

- Object class Loss
- Localization Loss

YOLOv8



- Multi-scaled objects detection of different sizes
- Anchor-free model

YOLOv8

Loss function

- **Classification** Loss: Cross-entropy

- **Objectness** Loss: Binary cross-entropy (presence or absence)

$$L_{BCE} = -\frac{1}{N}\sum_{i}^{N} \hat{y}_i \ln y_i + (1-\hat{y}_i)\ln(1-y_i)$$

- **Location** Loss: Complete IoU (error in locating the object)

$$L_{CIoU} = 1 - IoU + \frac{d^2}{C^2} + \alpha v$$

Training

Model: **YOLOv8x**

Model	params (M)	FLOPs (B)
YOLOv8n	3.2	8.7
YOLOv8s	11.2	28.6
YOLOv8m	25.9	78.9
YOLOv8l	43.7	165.2
YOLOv8x	68.2	257.8

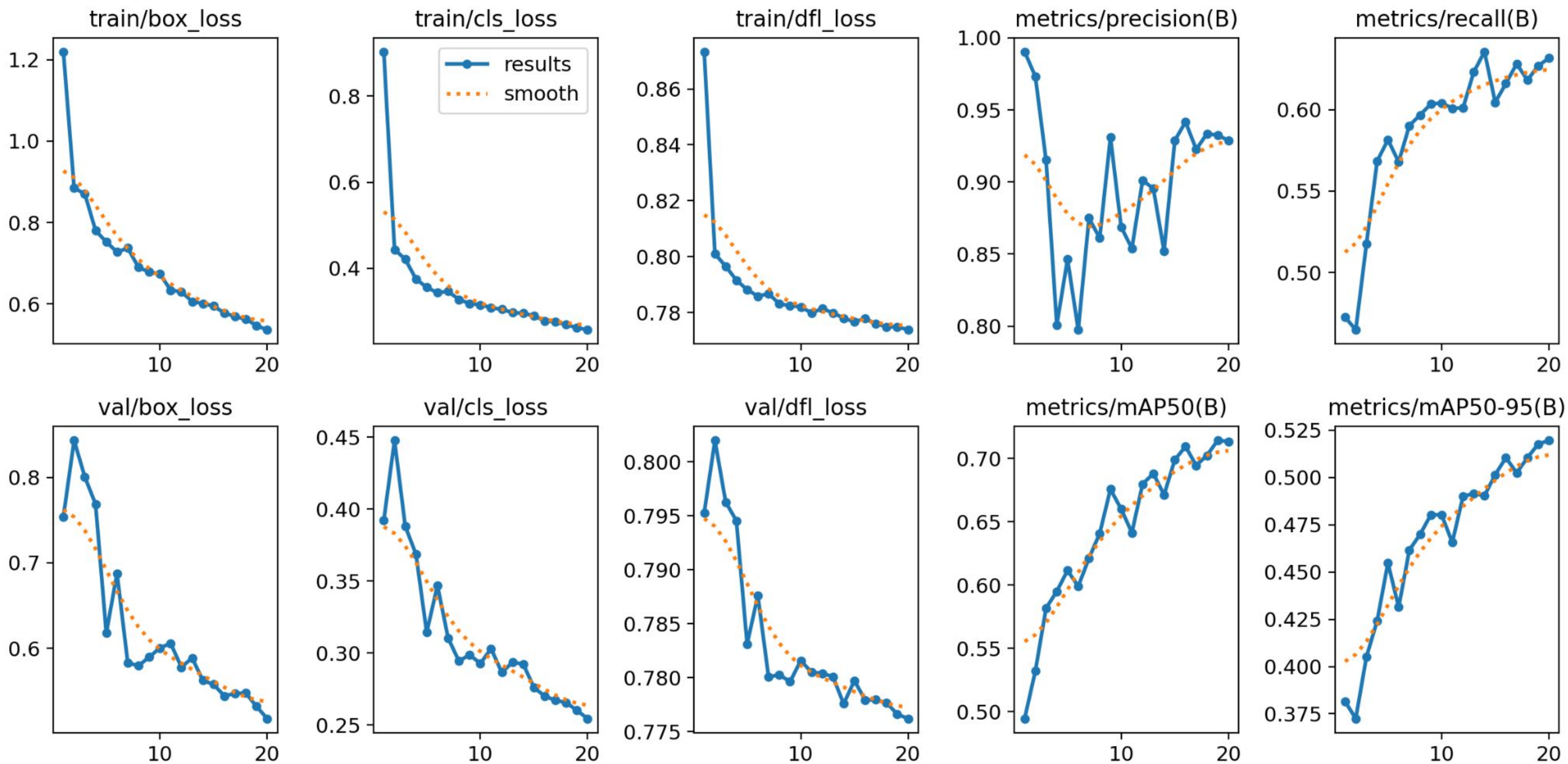
- Transform MOT into YOLO format
- Folder architecture
- Train – Val Split 80/20
- Image size, Batch size, optimizer, threshold

Final model

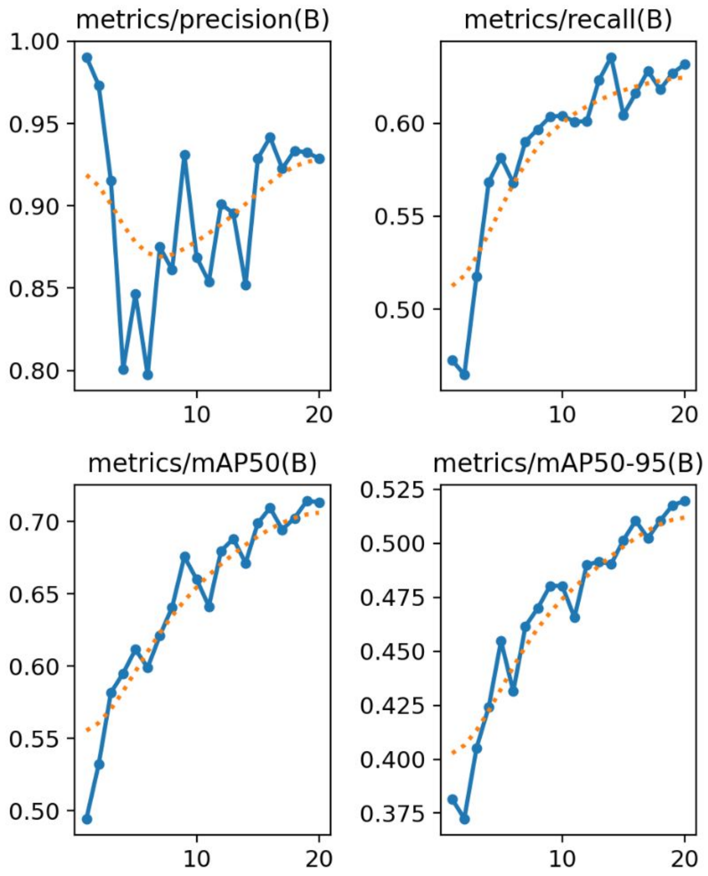
Overview

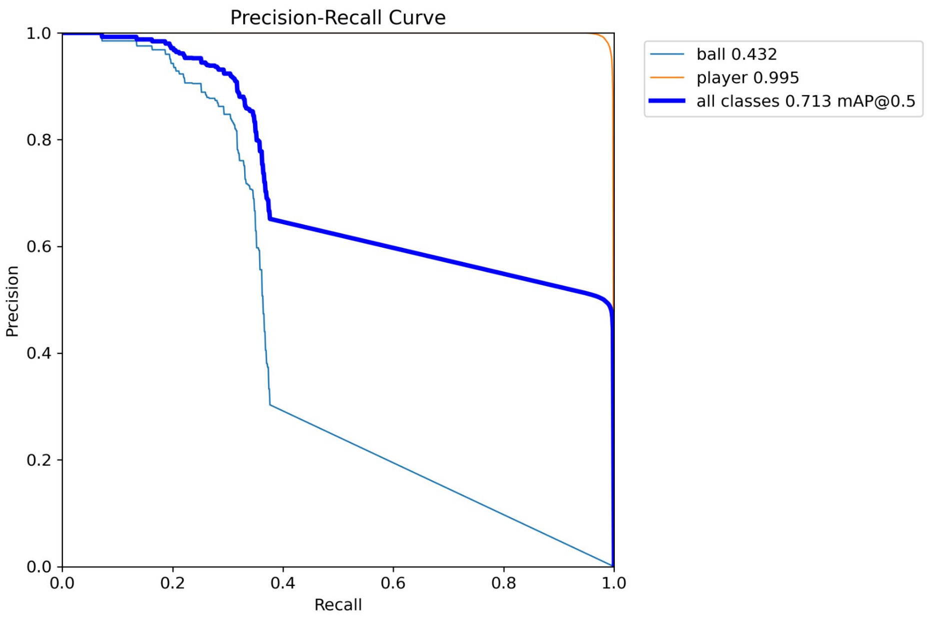
- Fine-tuned YOLOv8x
- Loss function
  - Class+Objectness+Loc
- Optimizer
  - AdamW
- Data augmentation
  - Mosaic augmentation
- Time required
  - 1.5 hours IDUN
- Runtime Analysis
  - 27.2ms preprocess
  - 267.8ms inference
  - 5578.6ms postprocess per image

Results *Quantitative*

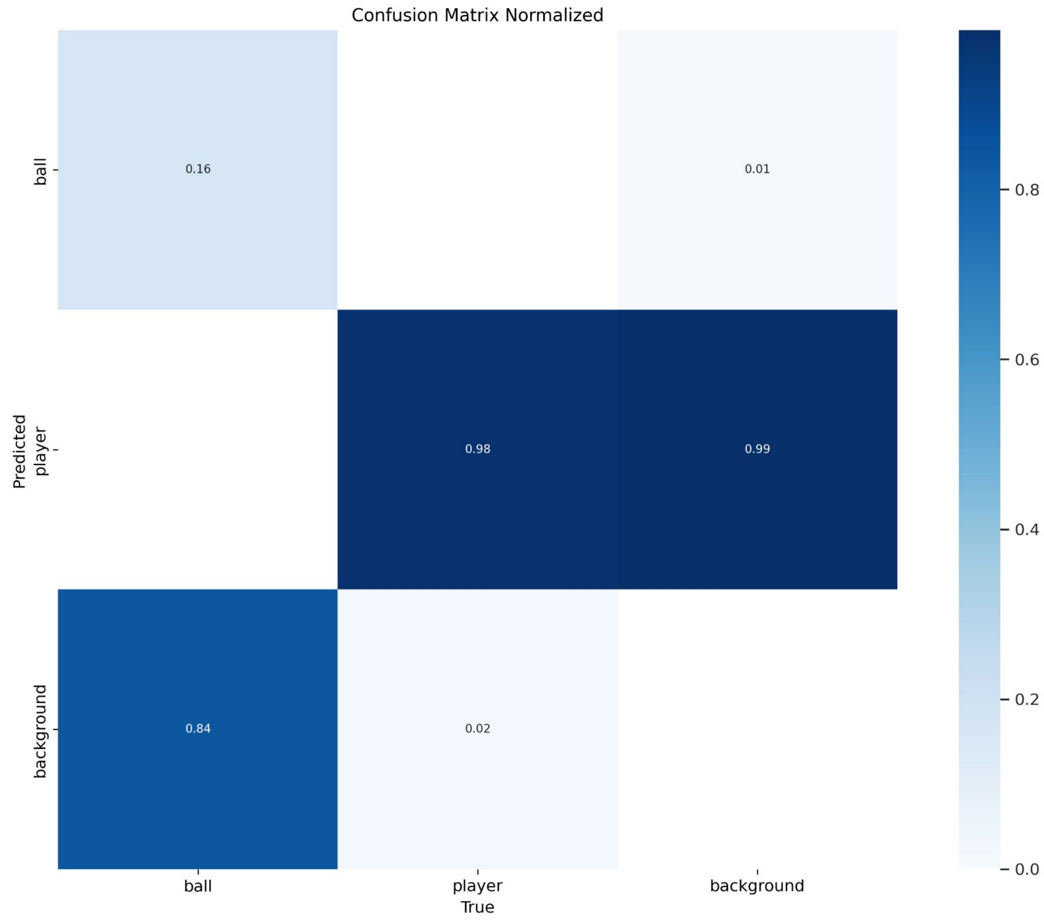


Results *Quantitative*





Results *Quantitative*



Normalized confusion matrix

- Ball detection
- Background

Results *Qualitative*

Carbon Footprint

Approximation Up

```
+---------------------------------------
| NVIDIA-SMI 545.23.08             Driver
|---------------------------------------
| GPU  Name                 Persistence-M
| Fan  Temp   Perf          Pwr:Usage/Cap
|
|=======================================
|   0  Tesla P100-PCIE-16GB           On
| N/A   33C    P0            32W / 250W
|
+---------------------------------------
|   1  Tesla P100-PCIE-16GB           On
| N/A   33C    P0            30W / 250W
|
+---------------------------------------
```

$$W = Pt \implies W = 250W \times 1.5h = 375Wh$$

- Electric car energy economy measurement = *x*

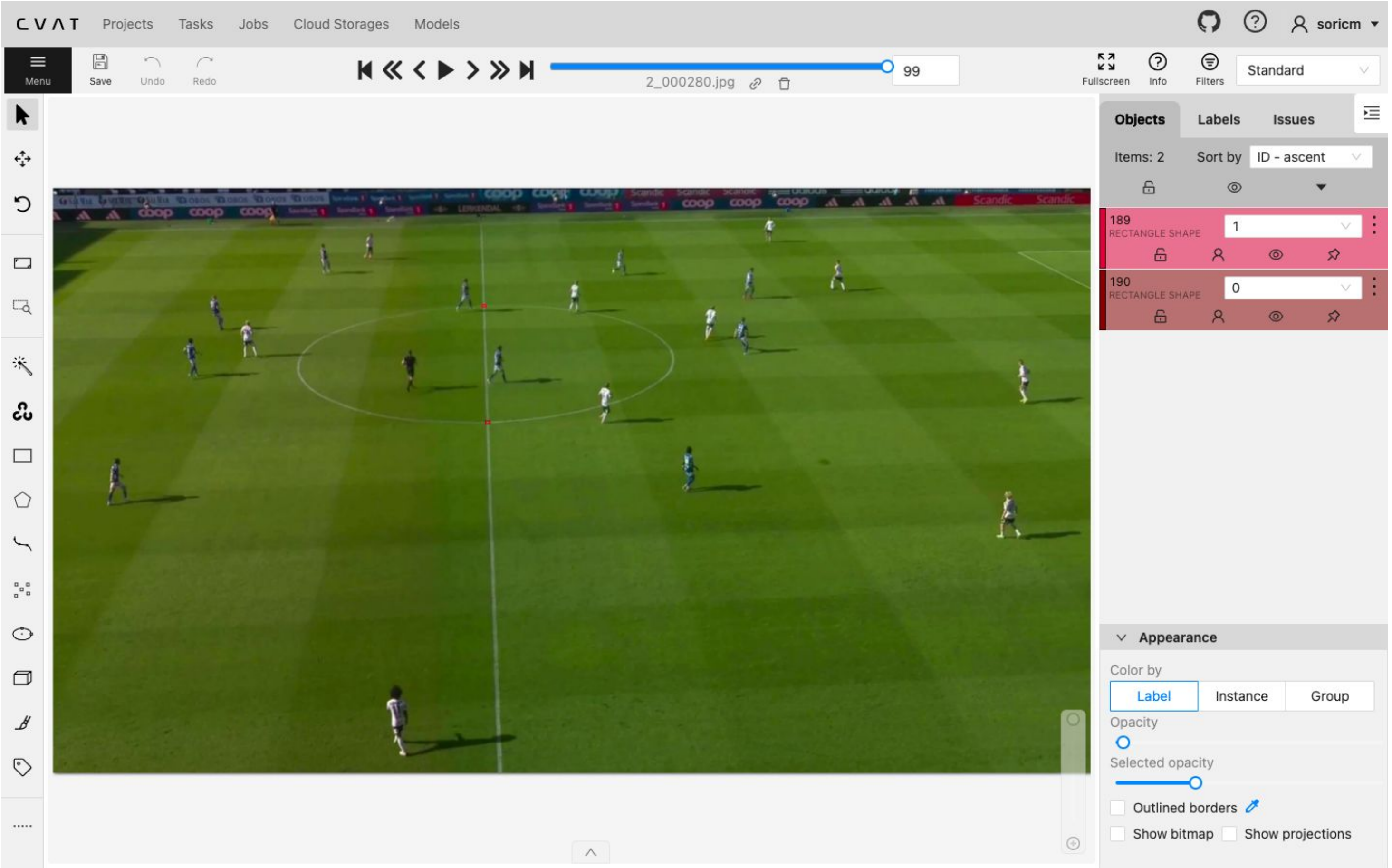
$$x \in [97, 198]\, Wh/km$$

$$\implies \frac{W}{x} \in [1.9, 3.9]\, km$$

Discussion

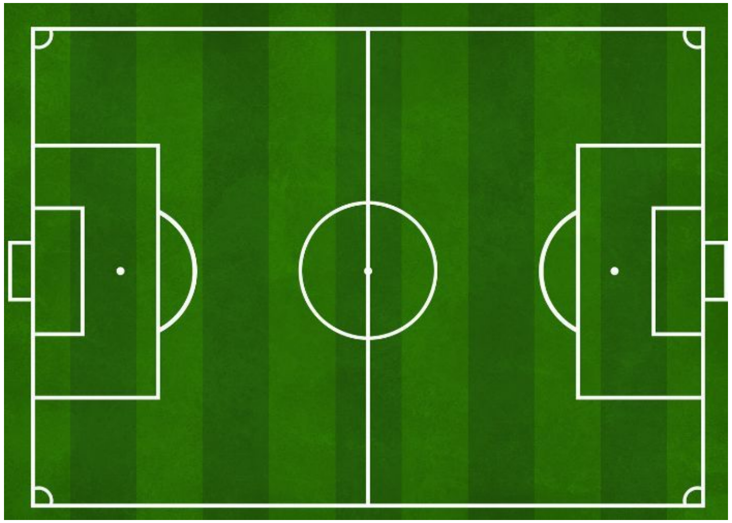
- Unbalanced data (1/22)
- Shape size
- Tracking
- Try another model Faster R-CNN
- Contribution: Marijan Sorić

Keypoint and Analysis



Manual annotation 100 images with CVAT

Keypoint and Analysis

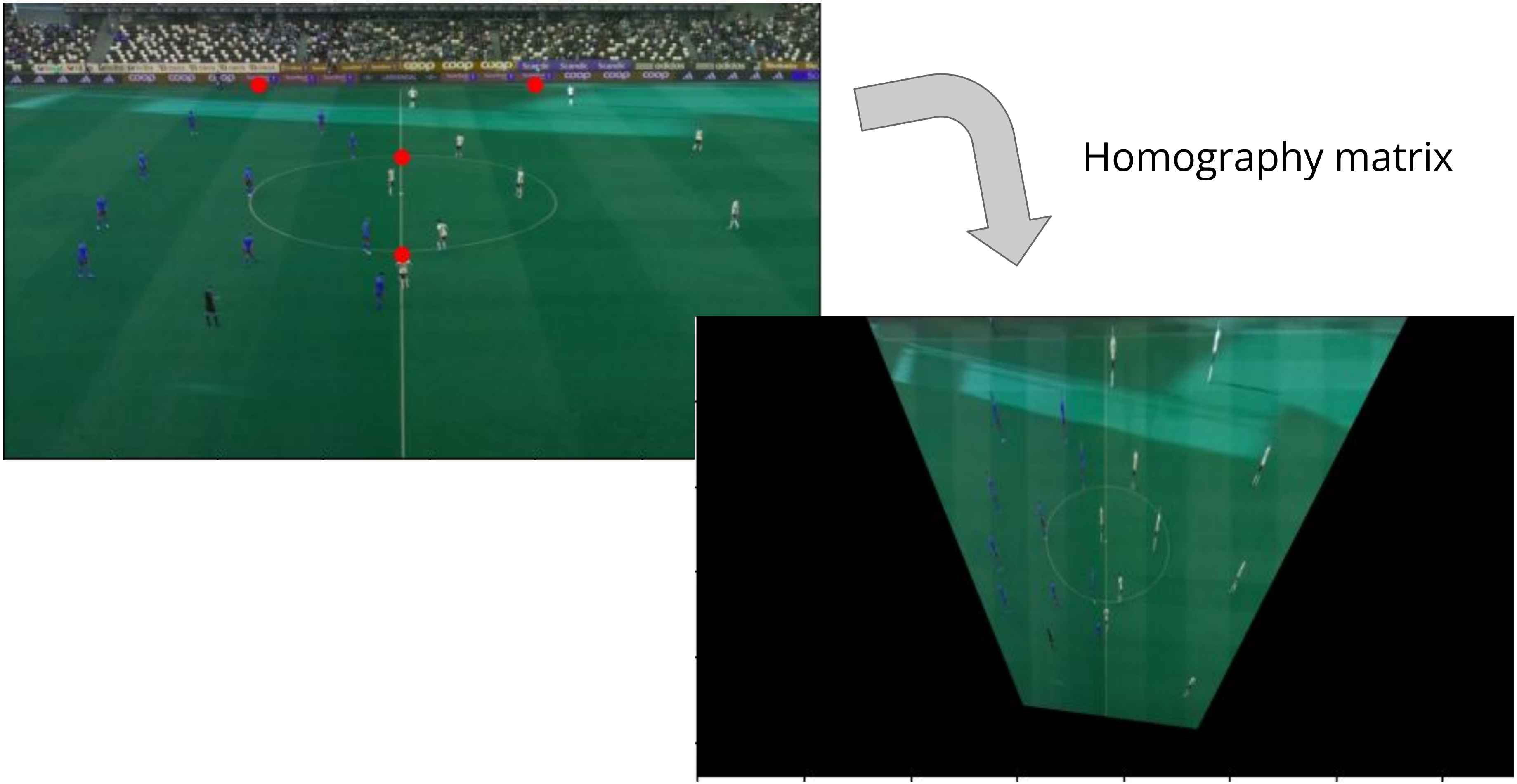


Length : 105m, Width : 68m

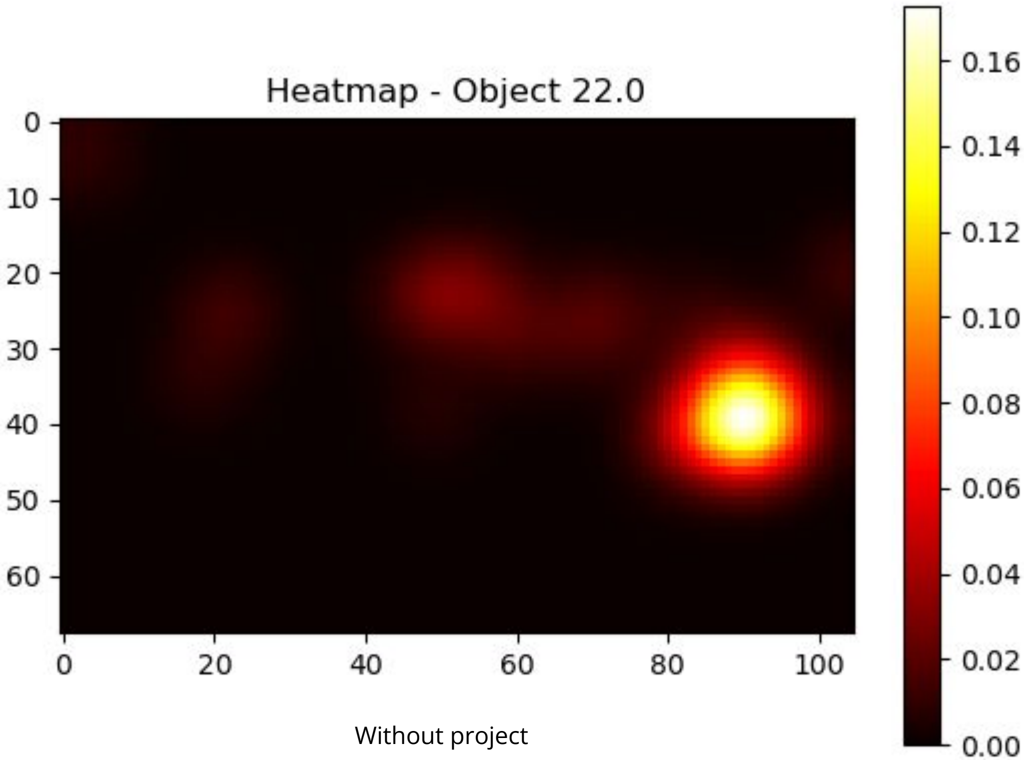
- Need to measure real distance for data analysis
- Projective transformation
- Find homography matrix

$$\begin{pmatrix} x^* \\ y^* \\ 1 \end{pmatrix} = \underbrace{\begin{pmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{pmatrix}}_{H_P} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Keypoint and Analysis



Keypoint and Analysis



Without project

Features of interest

- Distance run
- Speed distribution
- Heatmap

Sources

- https://learnopencv.com/iou-loss-functions-object-detection/#ciou-complete-iou-loss
- https://docs.ultralytics.com/fr/yolov5/tutorials/architecture_description/#41-compute-losses
- https://blog.roboflow.com/whats-new-in-yolov8/
- https://medium.com/@VK_Venkatkumar/yolov8-architecture-cow-counter-with-region-based-dragging-using-yolov8-e75b3ac71ed8
- https://docs.ultralytics.com/modes/train/