Recurrent CNNs for Bounding Box Stability in Object Detection

Prerna Dhareshwar, Donovan Fung, Sanjeev Suresh
Stanford University

Abstract

Most modern object detection algorithms (YOLO,SSD) are prone to bounding box jitter. Our project explores the feasibility of attaching a recurrent neural network at the end of a YOLO detector to de-noise/stabilize a jittery bounding box trajectory.

Bounding box jitter?

Causes

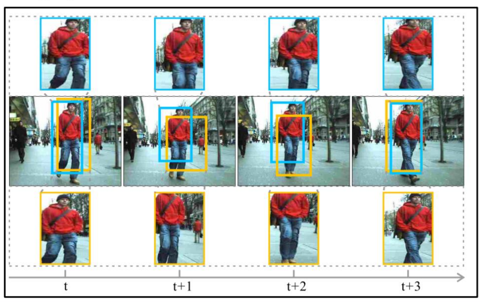
- Inherent pixel noise in camera sensors
- Improper aggregation of proposed bounding boxes

We care because

- Problematic in applications such as in surveillance where the behavior of an object depends on the bounding box movement
- Distracting!

Example

- Two bounding box trajectories are shown below, blue and orange. Notice how the center position of the orange box is not fixated on the person frame by frame

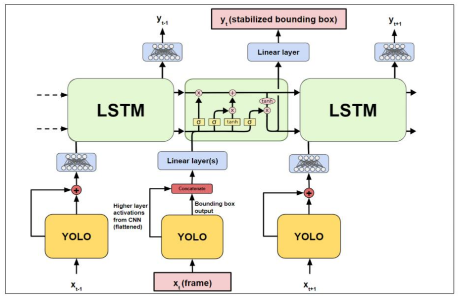


Methods

- Use Kalman Filters to "predict" bounding box trajectory and filter noisy bounding box predictions
- Fix improper aggregation in YOLO/SSD by using weighted Non-Max Suppression
- **Use an RNN as a filter and achieve improvement in performance!**

Approach

Architecture:



Training:

- Trained on MOT2015 bounding box trajectories, each ranging anywhere from 100-600 frames
- YOLO retrained for single-class detection
- Custom stability loss function, 20k epochs

LSTM inputs:

- Bounding boxes
- Flattened higher layer CNN feature maps

Evaluation Metrics:

- **Center position error**

$$e_x^f = \frac{x_p^f - x_g^f}{w_g^f}, \quad \sigma_x = std(e_x), \quad e_y^f = \frac{y_p^f - y_g^f}{h_g^f}, \quad \sigma_y = std(e_y)$$

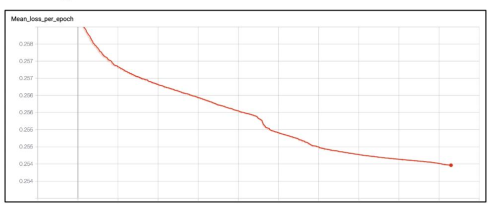
$$E_C = \sigma_x + \sigma_y$$

- **Scale and ratio error**

$$e_s^f = \sqrt{\frac{w_p^f h_p^f}{w_g^f h_g^f}}, \quad \sigma_s = std(e_s), \quad e_r^f = \left(\frac{w_p^f}{h_p^f}\right)/\left(\frac{w_g^f}{h_g^f}\right), \quad \sigma_r = std(e_r)$$

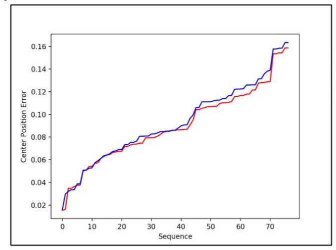
$$E_R = \sigma_s + \sigma_r$$

Training Loss:

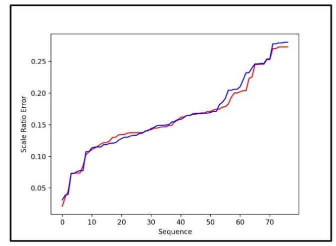


Results

Center position error:



Scale and ratio error:



	Pure YOLO	YOLO + SORT	YOLO + RNN
Center Position Error	93.01	71.11	89.89
Scale Ratio Error	163.27	323.979	161.44

- Results show YOLO+RNN improves performance by 5-8% with initial training
- With better training, confident in producing better performance
- Want to also try and implement different RNN architectures to figure out what is the best for bounding box stability

References

- Zhang, Hong, and Naiyan Wang. "On The Stability of Video Detection and Tracking "
- Leal-Taixé, Laura, et al. "MOT Challenge 2015: Towards a benchmark for multi-target tracking