

COMP-SCI 5588

Data Science Capstone

Professor: Dr Yugyung Lee

Term name: Bug Killers

Apr 24, 2025

Github Link: <https://github.com/nanxuanhui/DSCapstone.git>

Content

<i>Description of the Integrated Component(s)</i>	<i>2</i>
<i>Improved.....</i>	<i>2</i>
<i>Challenges Encountered During Integration.....</i>	<i>2</i>
<i>Contribution</i>	<i>3</i>
<i>Conclusion</i>	<i>4</i>

Description of the Integrated Component(s)

We integrated the Top-1 box selection mechanism from the NeurIPS 2024 paper YOLOv10: Real-Time End-to-End Object Detection into our fall detection system. This technique replaces traditional Non-Maximum Suppression (NMS) with a more efficient and deployable method where only the most confident box is kept per class. To implement this within our YOLOv5 pipeline:

- We introduced `decode_yolo_output()` to transform model outputs into usable detection format;
- We applied `one_to_one_top1_filter()` to perform YOLOv10-style per-class Top-1 filtering;
- The standard `non_max_suppression()` function was removed, completing a transition to an NMS-free inference framework.

Improved

Our fall detection system improved significantly in both inference efficiency and deployment capability. Specifically:

- The post-processing stage is now deterministic and faster to compute;
- Exporting to edge-optimized formats such as CoreML became easier and more reliable;
- Results are clearer, with one consistent output per class per frame;
- The performance remained stable, with comparable mAP and better interpretability.

Use NMS Free Top 1:

```
video 1/1 (1669/1676) /Users/nanxuan/Desktop/2025Spring/5588/Week14/yolov5/test/test2.mp4: 384x640 1 fallls, 120.5ms
video 1/1 (1670/1676) /Users/nanxuan/Desktop/2025Spring/5588/Week14/yolov5/test/test2.mp4: 384x640 1 fallls, 121.1ms
video 1/1 (1671/1676) /Users/nanxuan/Desktop/2025Spring/5588/Week14/yolov5/test/test2.mp4: 384x640 1 fallls, 119.2ms
video 1/1 (1672/1676) /Users/nanxuan/Desktop/2025Spring/5588/Week14/yolov5/test/test2.mp4: 384x640 (no detections), 137.2ms
video 1/1 (1673/1676) /Users/nanxuan/Desktop/2025Spring/5588/Week14/yolov5/test/test2.mp4: 384x640 (no detections), 131.3ms
video 1/1 (1674/1676) /Users/nanxuan/Desktop/2025Spring/5588/Week14/yolov5/test/test2.mp4: 384x640 (no detections), 118.1ms
video 1/1 (1675/1676) /Users/nanxuan/Desktop/2025Spring/5588/Week14/yolov5/test/test2.mp4: 384x640 (no detections), 120.2ms
video 1/1 (1676/1676) /Users/nanxuan/Desktop/2025Spring/5588/Week14/yolov5/test/test2.mp4: 384x640 (no detections), 119.4ms
Speed: 0.1ms pre-process, 117.3ms inference, 0.2ms NMS per image at shape (1, 3, 640, 640)
Results saved to runs/detect/exp
```

Original:

```
video 1/1 (1669/1676) /Users/nanxuan/Desktop/2025Spring/5588/Week14/yolov5/test/test2.mp4: 384x640 1 fallls, 116.6ms
video 1/1 (1670/1676) /Users/nanxuan/Desktop/2025Spring/5588/Week14/yolov5/test/test2.mp4: 384x640 1 fallls, 120.2ms
video 1/1 (1671/1676) /Users/nanxuan/Desktop/2025Spring/5588/Week14/yolov5/test/test2.mp4: 384x640 1 fallls, 116.8ms
video 1/1 (1672/1676) /Users/nanxuan/Desktop/2025Spring/5588/Week14/yolov5/test/test2.mp4: 384x640 (no detections), 118.6ms
video 1/1 (1673/1676) /Users/nanxuan/Desktop/2025Spring/5588/Week14/yolov5/test/test2.mp4: 384x640 (no detections), 116.9ms
video 1/1 (1674/1676) /Users/nanxuan/Desktop/2025Spring/5588/Week14/yolov5/test/test2.mp4: 384x640 (no detections), 118.8ms
video 1/1 (1675/1676) /Users/nanxuan/Desktop/2025Spring/5588/Week14/yolov5/test/test2.mp4: 384x640 (no detections), 116.8ms
video 1/1 (1676/1676) /Users/nanxuan/Desktop/2025Spring/5588/Week14/yolov5/test/test2.mp4: 384x640 (no detections), 113.3ms
Speed: 0.1ms pre-process, 117.2ms inference, 0.3ms NMS per image at shape (1, 3, 640, 640)
Results saved to runs/detect/exp2
```

Challenges Encountered During Integration

We encountered various technical challenges, including:

- Tensor shape mismatches and dimensionality errors during decoding;
- Output inconsistencies between Torch and ONNX export formats;
- Misinterpretation of blank detections due to overly high confidence thresholds;
- Avoiding interference with YOLOv5's original pipeline while replacing NMS.

Contribution

Team Member	Contribution	Percentage
Hui Jin	Integrated the YOLOv10 Top-1 inference logic into detect.py by replacing the original NMS function. Designed and implemented decode_yolo_output() and one_to_one_top1_filter() to enable NMS-free decoding and filtering. Debugged multiple shape-related runtime errors to ensure stable output.	40%
Jayadithya Nalajala	Conducted performance benchmarking and comparison between the original YOLOv5+NMS and the Top-1 filtered approach. Measured inference speed, average detection accuracy, and analyzed false positives and false negatives using test videos.	20%
Saniya Pandita	Implemented detection visualization and output saving functionality, including writing results to image overlays and exporting .txt/.csv files. Ensured directory structure and output formatting met project standards. Verified prediction stability across varied video frames.	20%
Sai Jahnavi Devabhakthuni	Compiled and edited the final bilingual report; ensured technical descriptions were clear and consistent. Coordinated model conversion to CoreML and tested ARKit compatibility. Helped maintain alignment between the integrated	20%

	component and the broader project objectives.	
--	-----------------------------------------------	--

Conclusion

This session enabled our team to successfully integrate a state-of-the-art concept from NeurIPS 2024 into a practical fall detection system. The Top-1 selection mechanism improved clarity, speed, and deployability, which are critical for real-time edge inference. More importantly, the task strengthened our collaborative development skills and deepened our understanding of modern detection pipelines.