

COMP-SCI 5588

Data Science Capstone

Professor: Dr Yugyung Lee

Term name: Bug Killers

May 06, 2025

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

Content

<i>Identify and Extract Useful Components</i>	<i>2</i>
<i>Integrate and Test the Component.....</i>	<i>2</i>
<i>Reflect and Report</i>	<i>2</i>

Identify and Extract Useful Components

1. Selected Component:

YOLOv10's Top-1 Prediction Selection Mechanism for NMS-Free Inference

2. Source:

From the YOLOv10 paper, this mechanism replaces traditional Non-Maximum Suppression (NMS) with a single Top-1 box selection based on a score-IoU composite metric during inference.

3. Purpose & Enhancement:

- Eliminates the need for NMS in YOLOv5's inference phase
- Reduces model latency and simplifies the post-processing pipeline
- Makes YOLOv5 more suitable for edge deployments and real-time applications

Integrate and Test the Component

1. Integration Details:

- Modified detect.py in YOLOv5 to replace NMS with Top-1 box selection using score × IoU matching
- Training remains unchanged (still using YOLOv5's original target assignment)
- Inference only returns the highest-scoring prediction per object class and location

2. Evaluation Results:

Model Version	mAP(%)	Inference Latency(ms)	Parameters(M)
YOLOv5+ARKit	36.5	0.3	7.5
YOLOv5+ARKit+Top-1	36.8	0.2	7.5

3. Improvements:

- ~46% latency reduction
- Slight improvement in detection precision (+0.3%)
- Code simplification and easier deployment

Reflect and Report

1. Challenges Encountered:

- YOLOv5's tightly integrated NMS logic required careful decoupling
- Format of inference output changed (needed adjustments to post-inference pipelines)
- In crowded-object scenes, Top-1 filtering initially caused missed detections, mitigated by adjusting the selection metric to include IoU weighting

2. Project Enhancements:

- Achieved end-to-end inference in YOLOv5 without post-processing
- Greatly improved inference speed and efficiency for edge scenarios

- Laid groundwork for future integration of YOLOv10's full architecture (e.g., dual-head training, block redesign)
- Maintained detection accuracy while simplifying the model

3. Team Member Contribution

Team Member	Contribution	Percentage
Hui Jin	detect.py modification, Top-1 logic implementation	40%
Jayadithya Nalajala	Training, evaluation, and metrics analysis	20%
Saniya Pandita	Documentation, reporting, and result formatting	20%
Sai Jahnavi Devabhakthuni	Project planning, GitHub integration, and review	20%