Helmet Detection with CBAM

Q and A Session

December 2025

This document contains all questions and answers from the debugging and analysis session for Helmet Detection using YOLOv8 with and without CBAM attention module.

Helmet Detection with CBAM - Q&A Session

Date: December 18, 2025

Table of Contents

1. Git Commands and Repository Setup

Q: What is the code to git pull?

A: To pull the latest changes from the remote repository:

Q: Error - "fatal: not a git repository"

A: This error means you're running git pull in a folder that is NOT a git repository. Navigate into the cloned folder first:

Q: How to connect to GitHub?

A: Common git commands:

Check existing connection

```
git remote -v
```

Add new remote

```
git remote add origin https://github.com/username/repository.git
```

Push changes

git push -u origin main

2. File Path Configuration Issues

Q: Why is "file not found" error occurring?

A: The paths in configuration files were pointing to wrong locations:

3. Model Testing and Accuracy Results

Q: Are test images and labels being used correctly?

A: Verification showed:

4. Why Results Were Bad - Domain Shift

Q: Why did the test results come so bad?

A: Initial Results:

5. Too Many Predictions Issue

Q: Why is the model making too many predictions (538 for 104 ground truth)?

A: The model was detecting "helmet" on:

6. CBAM Class Mismatch Issue

Q: Why are CBAM models making false detections while non-CBAM models work fine?

A: Critical mismatch between training and testing CBAM definitions:

7. Comparison of 4 Models

Q: What are the results of all 4 models?

Final Results (24 test images, 104 ground truth):

8. Why CBAM Decreased mAP

Q: Why did mAP drop after integrating CBAM when it's supposed to increase accuracy?

A: Multiple reasons:

1. Higher Precision, Lower Recall Trade-off

CBAM models are too conservative - they're more careful about what they detect (higher precision), but they miss more objects (lower recall).

2. CBAM Suppresses Weak but Valid Features

The attention mechanism might be suppressing features for:

3. CBAM Placement Issue

CBAM placed after every C2f block is too aggressive and disrupts the feature pyramid.

4. Not Enough Training

CBAM adds ~2-3% more parameters. With same epochs:

5. YOLOv8 is Already Optimized

YOLOv8 architecture is highly efficient. Adding attention can sometimes hurt rather than help.

9. Precision vs IoU Trade-off

Q: If precision increased, why did mAP decrease?

A: CBAM makes the model MORE SELECTIVE (stricter in what it considers a detection)

Visual Example:

Ground Truth: 10 helmets in image

The Formula:

$$\text{mAP} = \int_0^1 \text{Precision}(r) \, dr$$

10. Justification for Using CBAM

Q: Why use CBAM if it doesn't always increase accuracy?

A: CBAM's Purpose is NOT Just Accuracy:

Your Results Show CBAM Works Partially:

- Precision: 89%   93%   (+4%)   Fewer false positives

For Your Report:

"We integrated CBAM to study attention mechanism effects on helmet detection. While CBAM increased precision by 4% (reducing false positives), it decreased recall. This trade-off suggests CBAM makes the model more conservative. For applications where false alarms are costly (e.g., automated penalty systems), CBAM is beneficial. For applications where missing detections is critical (e.g., safety monitoring), standard YOLOv8 is preferred."

11. Why Recall Decreased with CBAM

Q: Why did recall decrease when CBAM is supposed to increase recall?

A: CBAM is NOT TRAINED PROPERLY in your experiment:

1. Training Duration Issue

| Issue | Impact |

2. Too Many CBAM Blocks

With 3+ CBAM blocks, small features get repeatedly suppressed at each stage:

3. Attention Collapse

The attention mechanism learned to SUPPRESS some features instead of ENHANCE them.

For Your Report:

"CBAM theoretically improves feature representation through attention. However, in our experiment with 100 epochs, CBAM showed conservative behavior - increasing precision but decreasing recall. This suggests the attention mechanism requires longer training to properly learn which features to enhance rather than suppress."

12. Fair Comparison - Same Epochs for All Models

Q: If CBAM needs more epochs, shouldn't we train all models for 200 epochs?

A: YES! For a valid experiment, all models must be trained under identical conditions.

Scientific Principle: Control Variables

| Variable | Must Be Same |

Your Current Comparison IS Valid:

| Approach | Valid? | Explanation |

Your Conclusion:

"Under identical training conditions (100 epochs), CBAM integration did not improve detection performance for helmet detection, with standard YOLOv8+COCO achieving the highest mAP of 87.4%."

13. CBAM Convergence Speed Explanation

Q: So why is CBAM integrated if it needs more epochs?

A: Different models converge at different rates:

Why This Happens:

| Model | Parameters | Learning Complexity | Convergence Speed |

The Real Comparison:

| Scenario | Normal | CBAM | Winner |

For Your Report:

"At 100 epochs, standard YOLOv8 outperformed CBAM variants. This is because simpler architectures converge faster. CBAM's additional attention parameters require longer training to learn effective feature weighting. In resource-constrained scenarios (limited training time), standard YOLOv8 is preferred. For extended training budgets, CBAM may eventually match or exceed standard performance."

Simple Analogy:

| Student | After 1 month | After 3 months |

Summary

Key Findings:

1. Best Model: COCO pretrained YOLOv8 (no CBAM) - 87.4% mAP@0.5

Recommendations:

1. For maximum detection rate: Use COCO pretrained YOLOv8

Document generated from Q&A session on December 18, 2025