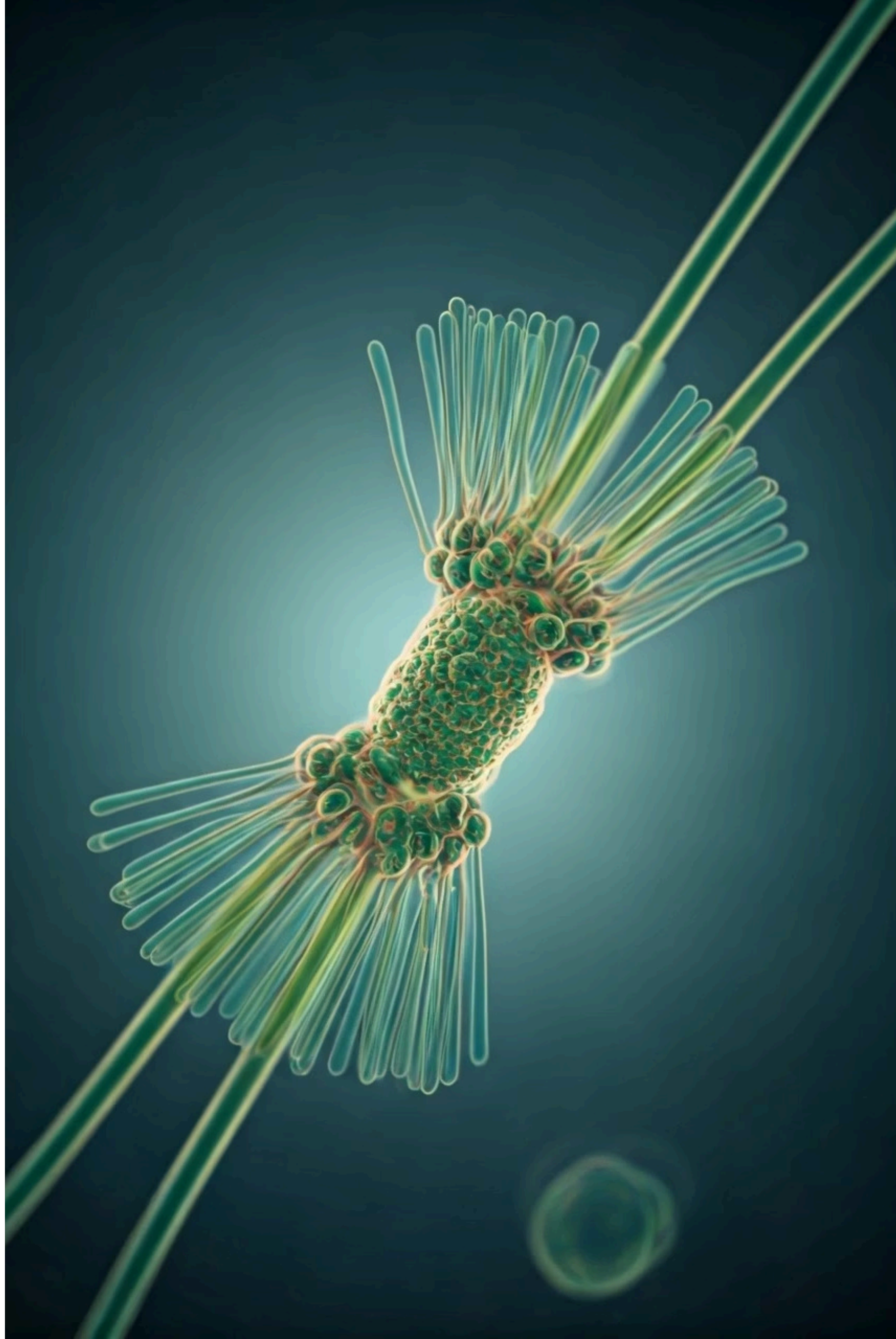
Comparative Analysis of Deep Learning Approaches for Bacterial Flagellar Motor Detection

Abhinaysai Kamineni
Pranav Dhawan
Aakash Singh

Wednesday April 30, 2025



Introduction

Bacterial Flagellar Motors

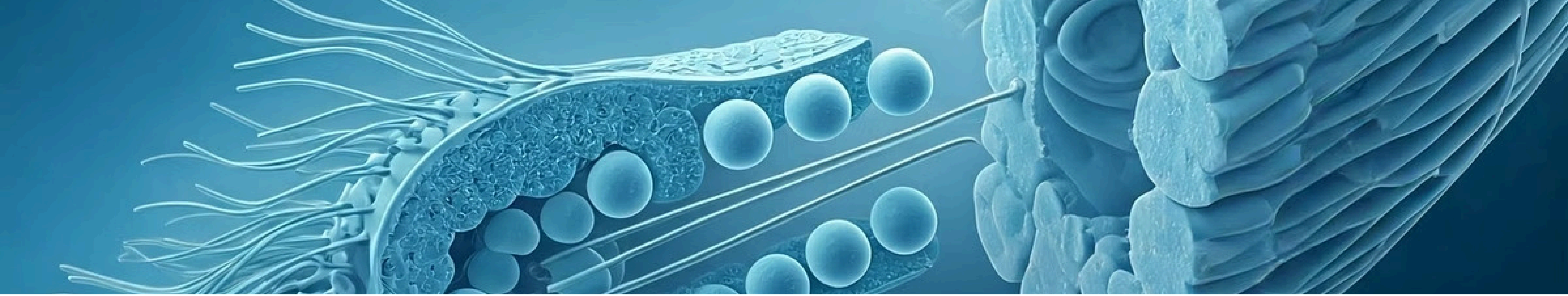
These nanoscale rotary machines are essential for bacterial motility. They play a critical role in infection processes.

Detection Challenges

Small size and low contrast make identification difficult. Noisy microscopy data further complicates automated detection.

Research Goal

Develop and compare detection methods. Our work aims to improve accuracy for these vital cellular components.



The Bacterial Flagellar Motor: Structure

Bidirectional Rotary Nanomachine

Spans bacterial membranes and functions in multiple directions.

Key Components

Consists ofbasal body, hook, and filament working together.

Motility Mechanism

Powers swimming through counterclockwise and clockwise rotation sequences.

Deep Learning Detection Methods Overview

YOLOv8/v10

Single-stage detector optimized for real-time performance. Uses unified architecture for speed and efficiency.

Faster R-CNN

Two-stage detection using region proposal network. Balances accuracy and computational demands.

CenterNet

Keypoint-based, anchor-free detection. Simplifies detection pipeline with center point emphasis.

Why CenterNet...?

Simpler Pipeline

Single-stage, model oriented approach

Efficient Detection

Predicts centers, size and offset in one pass

Ideal for Small Objects

Perfect for noisy tomogram slices

Data Preprocessing

Data Collection

~1,200 tomogram slices, grouped by tomo_id.

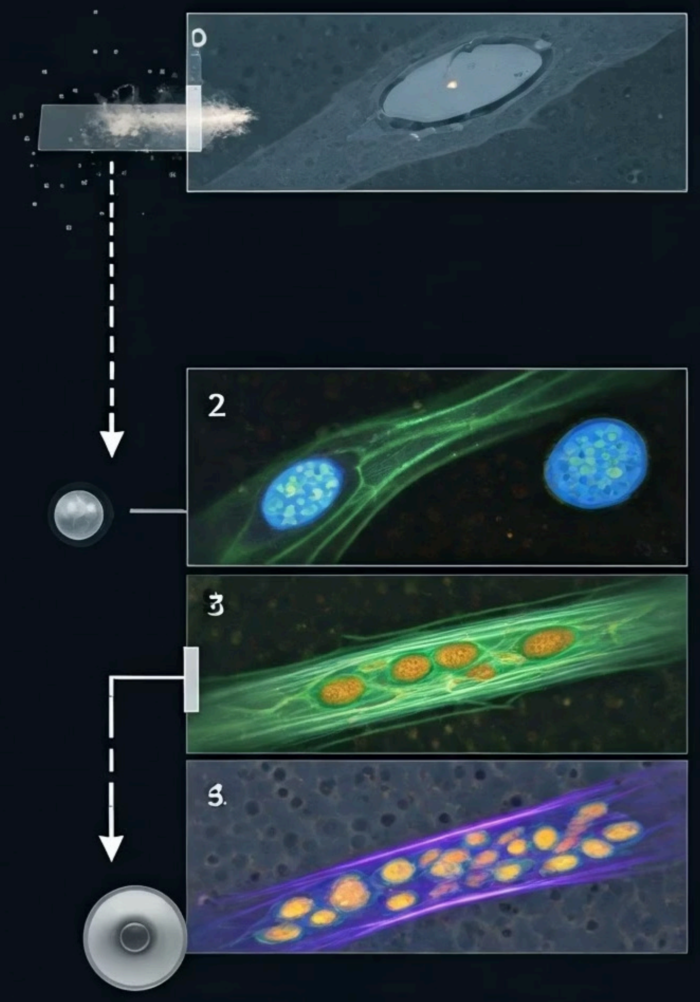
Ground-truth CSV (train_labels.csv) mapping each slice to one 100×100 px motor box.

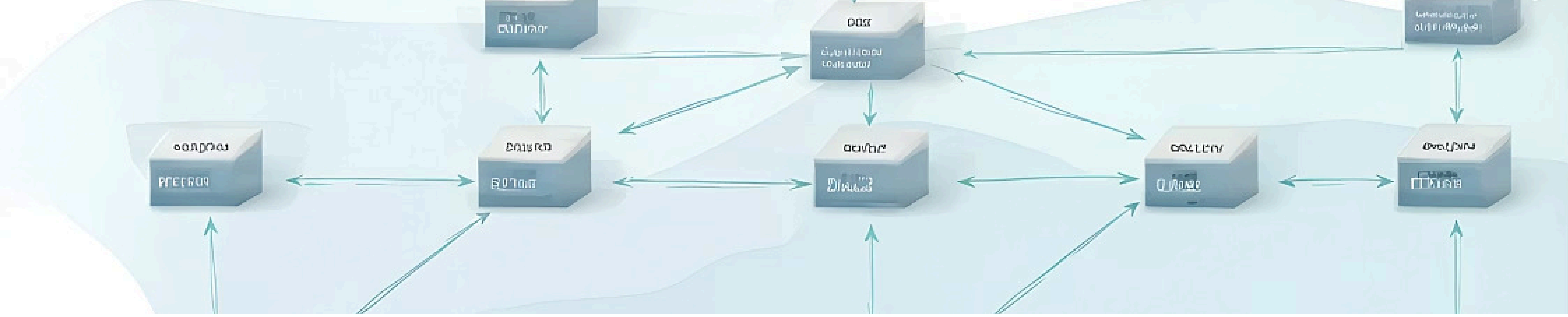
Image Preprocessing

- **Resize:** All slices resized (or padded/cropped) to 720×720 px.
  Z-score normalization: Subtract per-slice mean, divide by per-slice σ (no percentile clipping in code).
- **CLAHE** (clip limit = 2.0, tile grid = 8×8) applied on-the-fly.

Heatmap Generation

- **Downsample** to 180×180 heatmap grid.
  Heatmap: Place 2D Gaussian (σ = 2 px) at each motor center.
  Size & Offset: L1 regression maps generated for width/height and sub-pixel shifts.





CenterNet Architecture

Input Processing

Single-channel (1×720×720) grayscale slice into a ResNet-101 trunk (pretrained, cut after Layer4)

Produces C2 (256×180×180), C3 (512×90×90), C4 (1024×45×45) feature maps (roughly).

Feature Processing

- **Lateral 1×1 conv** on C2, C3, C4 → 256 ch
- **Top-down upsampling** (×2 bilinear) & fusion via 3×3 conv → unified P2 of 256×180×180

Decoding Stage

Top-K selection: pick 150 highest-scoring heatmap peaks.

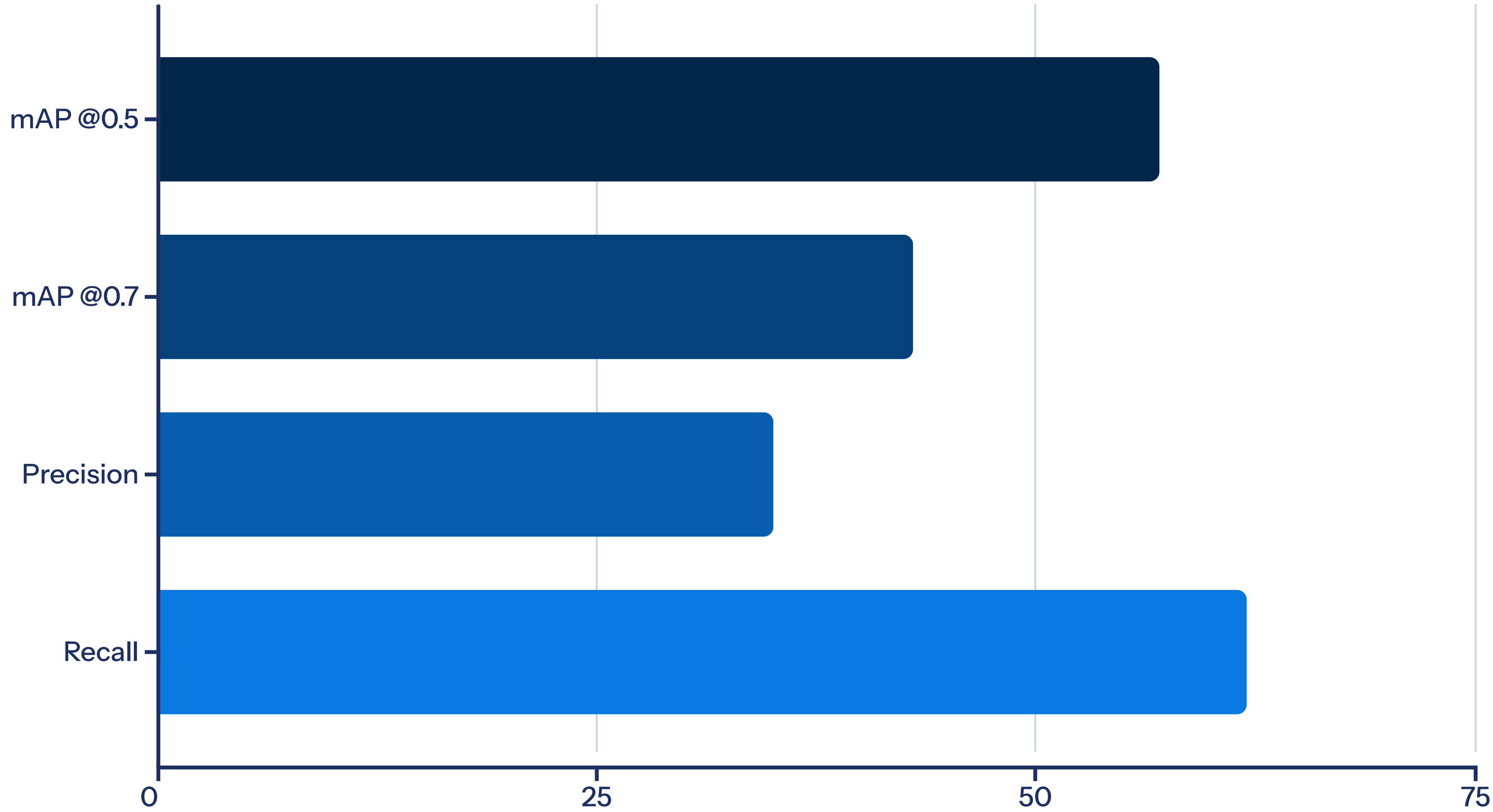
Box reconstruction: apply size & offset to each center.

Final Processing

Clip boxes to image boundaries

NMS with IoU threshold = 0.45

Current Results & Challenges



Decent Recall

Successfully finds most motors in images.

Fast Training

Simple model trains quickly at ~32 seconds per epoch.

Modest Precision

Still generating some false positives in detection.

YOLO Dataset & Preprocessing

Dataset Selection

BYU Flagellar Motor Dataset with tomographic slices. Designed for small, noisy motor detection.

Used TRUST value (4 or 6) to remove irrelevant images.

Preprocessing

Normalize slices using 2nd and 98th percentile.

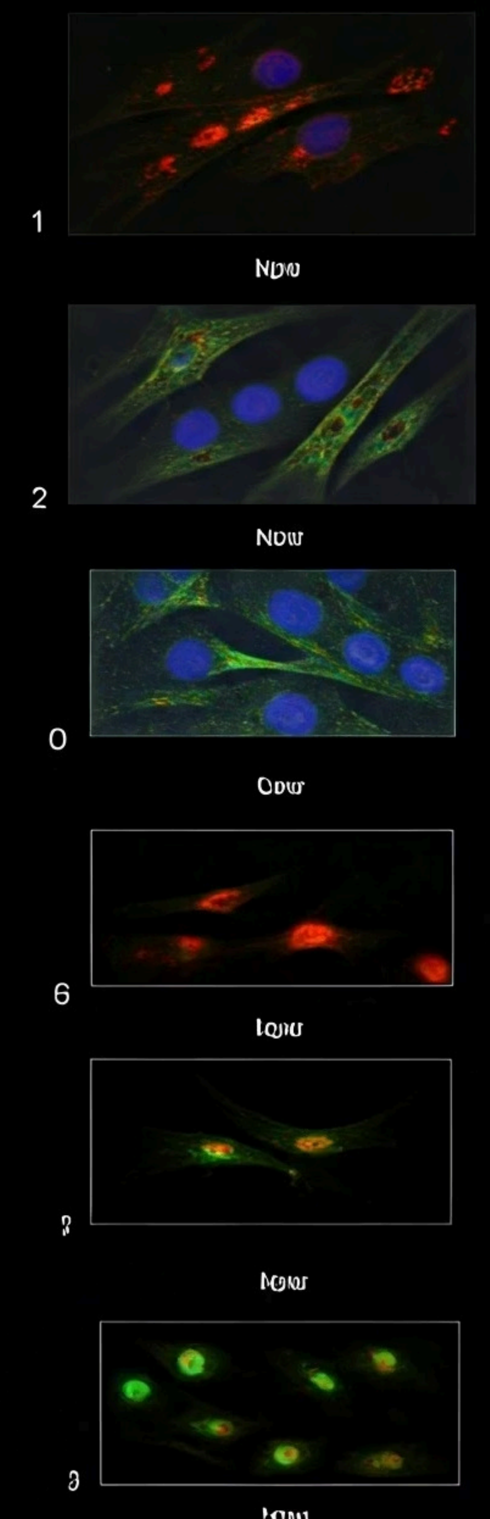
Create fixed 24×24 bounding boxes around motors.

Label Conversion

Transform coordinates into YOLO format to ensure compatibility with training architecture.

Train/Val Split

Divide by tomogram ID to prevent data leakage.



YOLOv8: Baseline Model

Model Architecture

YOLOv8-large with pretrained weights. Optimized for object detection tasks.

Training Configuration

100 epochs with 512 image size.

SGD optimizer with 0.001 learning rate.

Performance Metrics

mAP50: 0.800

mAP50-95: 0.422

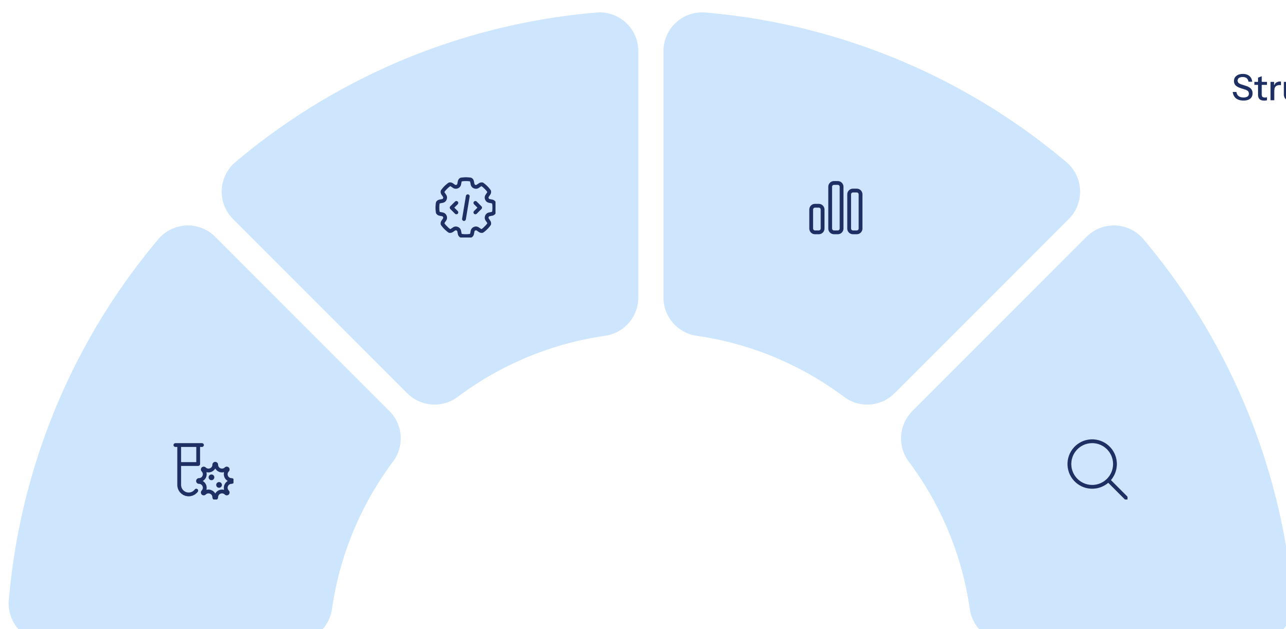
Precision: 0.676,

Recall: 0.779

Observations

Decent baseline performance.

Struggles with dense or noisy regions.



YOLOv10-X

Model Selection

YOLOv10x for improved small object detection with NMS (Non-Max Suppression) for post processing

Enhanced Training

300 epochs, 960×960 Image Size, 0.01 Learning Rate

Used Distributive Focal Loss

Advanced Augmentations

Mosaic, Mixup, Copy-Paste, and Flips

Superior Results

High MAP scores and perfect precision with 0.96 recall

YOLO Models Results Comparison

Metric	YOLOv8-l (Large)	YOLOv10-x (Extra Large)
mAP50	0.800	0.948
mAP50-95	0.422	0.630
Precision	0.676	1.000
Recall	0.779	0.960

Key Takeaways

- YOLOv10x outperforms in all metrics
- Perfect precision with high recall
- It performs optimal on confidence between 0.4 -0.5

Future Work

- Adaptive bounding boxes
- More augmentation techniques

Faster RCNN Approach

High Precision Detection

RPN and ROI pooling accurately localize small motors in noisy images.

Multi-Scale Feature Handling

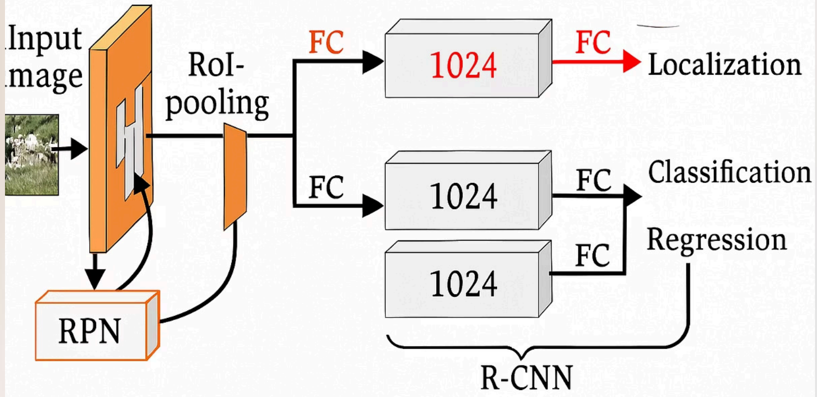
FPN backbone detects motors at various scales and growth stages.

Transfer Learning Benefits

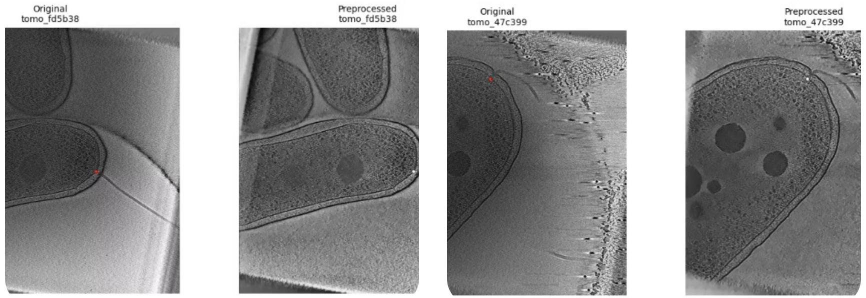
Pretrained ResNet-50 weights enable robust feature extraction with limited data.

Customizable Thresholds

Tunable parameters optimize sensitivity for faint motor signals.



Dataset Preparation & Image Preprocessing



437

Training Images

Including 10% background samples

80

Validation Images

With identical background ratio

0.5284

Initial mAP

Before preprocessing improvements

Data Augmentation and Hyperparameter Tuning

Horizontal & Vertical Flips

Improved directional robustness

Random Cropping

Enhanced positional invariance

Salt-and-Pepper Noise

Increased noise resilience

Scaling

Better size adaptability

0.6109

Final mAP

Overall performance score

0.8739

mAP50

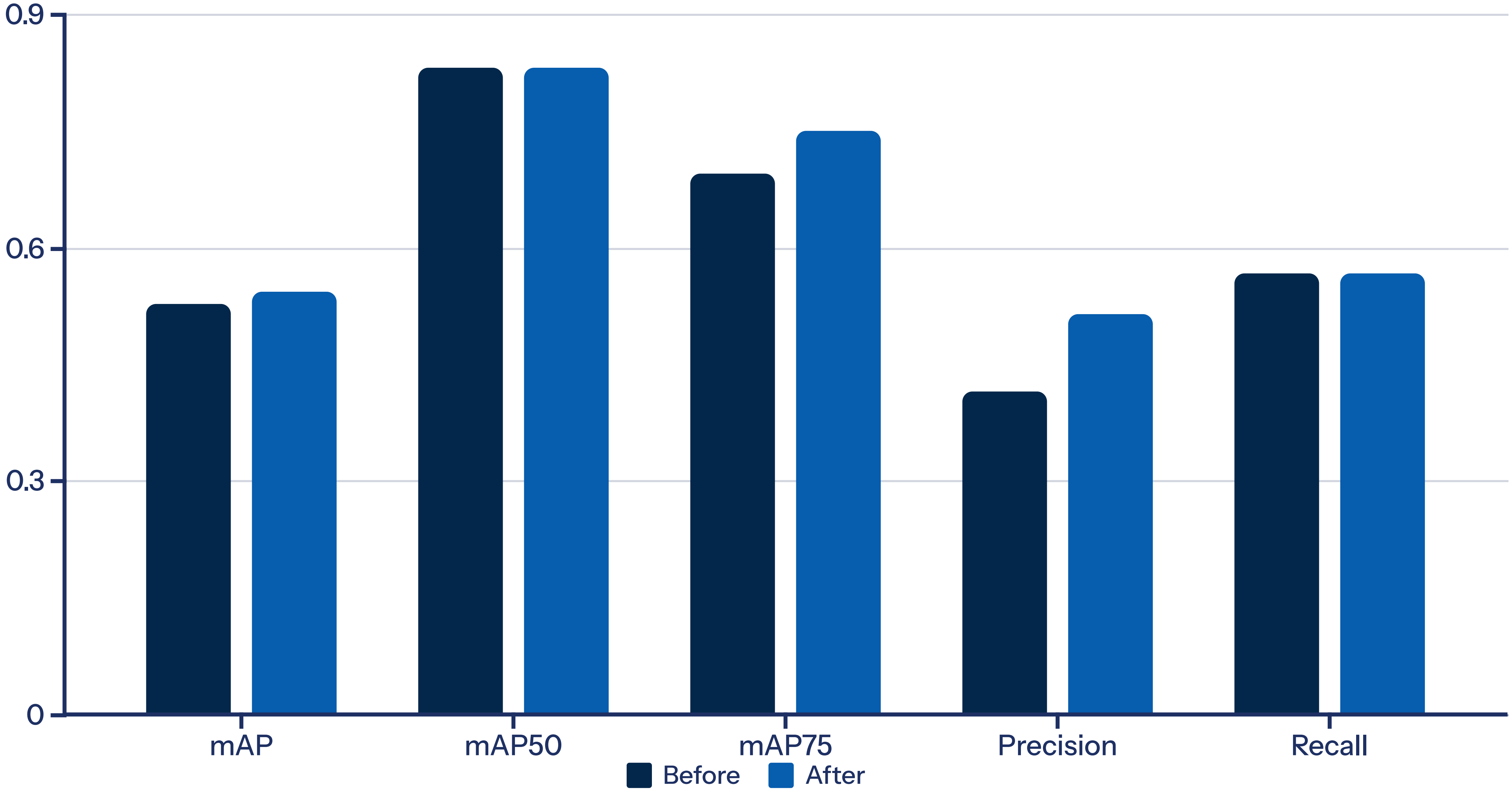
At 50% IOU threshold

0.7368

Recall

Proportion of motors found

Evaluation Metrics for Faster RCNN



Conclusion

Method Comparison

Each approach offers unique strengths with some challenges for different research needs.

Top Performers

Best Model: YOLOv10x with map50 = 0.948 and map95 = 0.63

Center Net Improvements

Along with the addition of ensembling a second detector like Faster RCNN will improve the MAP and the F1 Scores.