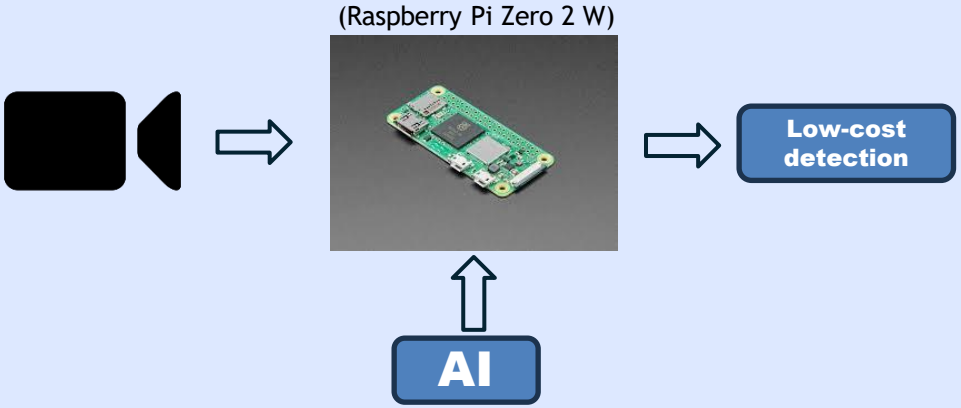
INO-YOLO: A Lightweight Real-Time Deep Learning Model for Drowning Detection on Low-Cost Edge Devices

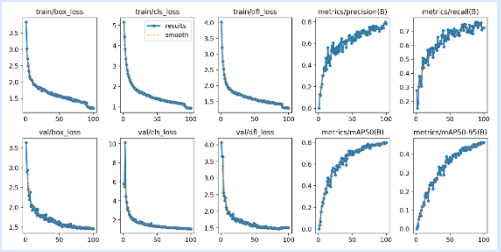
Research question

- How can real-time drowning detection systems become less expensive for ease of deployment in low-income regions?



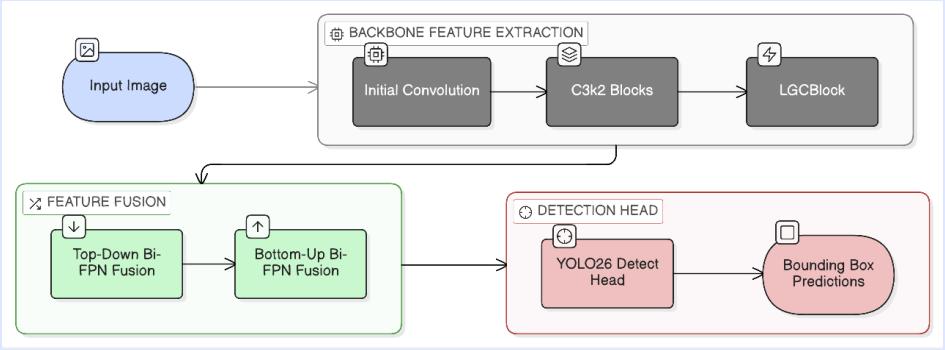
Data analysis and results

- Strong model results, all goals were met
  - **DmAP50 = 0.90, Drowning Recall = 0.86**
- Extremely lightweight, allowing for rapid real-time detection
  - **3.0** GFLOPS
  - **1,439,070** parameters
  - **2.95** MB
  - **256.4** FPS



Methodology

- **Diverse dataset**: 2,000 original images of swimming/drowning, including images from various environments, lighting conditions, perspectives etc.
- **Deep learning**: Ultralytics YOLO26 optimized with LGCBlock + Bidirectional Feature Pyramid Network (BiFPN)
  - Lightweight model without sacrificing accuracy



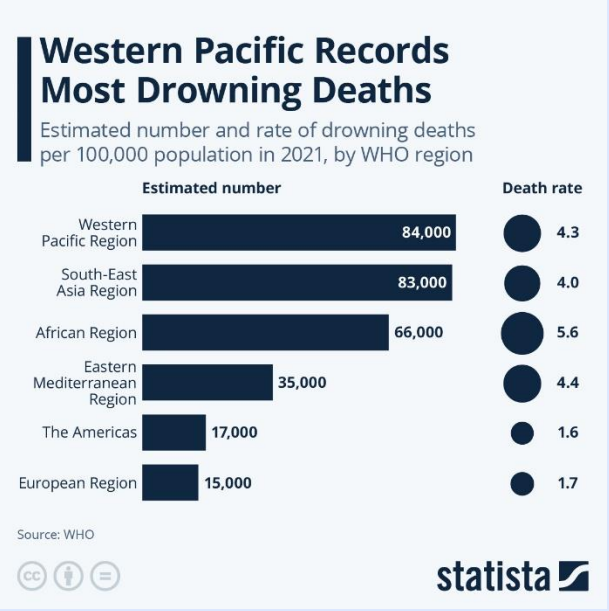
Interpretation and conclusions

- **Robust** and **lightweight** drowning detection model allowing for **low-cost** drowning detection
- The **high mAP50** and recall of our system enhances confidence in the detector, improving its effectiveness
- Capable in various environments
- Can be **directly used** on inexpensive microcontrollers for drowning detection
- Extremely **rapid** real-time detection, higher FPS than all other drowning detection models tested on the same dataset

Introduction

- Drowning is the 3rd leading cause of unintentional injury death, accounting for 7% of all injury-related deaths.
- The number of global drowning deaths reached 300,000 in 2021—equivalent to over 34 fatalities per hour
- Low and middle-income countries account for over 90% of unintentional drowning deaths, with most deaths occurring in the Western Pacific Region
- More children ages 1-4 die from drowning than any other cause of death

(WHO, 2024)



(WHO, 2025)

Drowning Detection Approaches

- **Wearable Sensing Systems**: collect physiological and motion signals using accelerometers, gyroscopes, and other sensors
- **Camera + Computer Vision Systems**: continuous monitoring, don't require equipment, non-intrusive, less expensive, can monitor all swimmers at once

(WAVE Drowning Detection System)

(SwimEye Drowning Detection System)

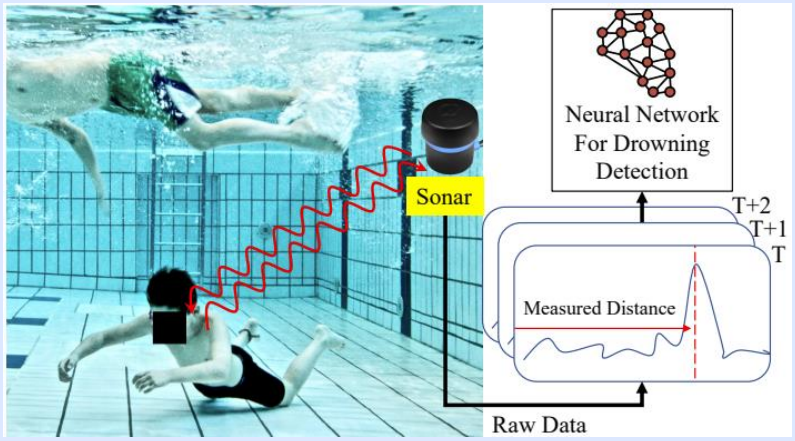
System	Comfort	Cost	Maintenance	Noise
Wearables	Medium	High	High	None
Vision-based	High	Depends*	Low	High*

*Can be optimized

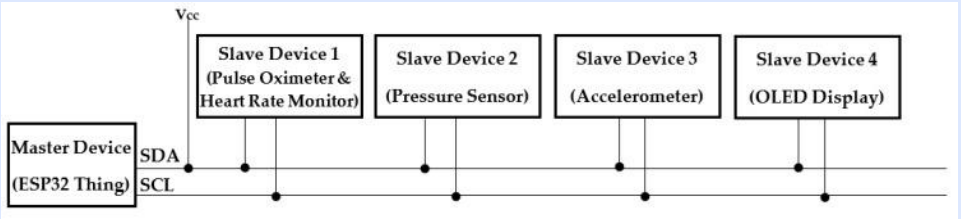
Drowning Detection Approaches In-Depth

Sensor Based and Other Approaches

- Wearable pressure sensors -> **can't distinguish underwater vs drowning**
- Wearable motion sensing using accelerometers & gyroscopes -> **high-energy play can be seen as drowning**
- Radio-frequency systems -> well implemented method, but **ineffective for low-cost detection**
- Sonar-based drowning detection -> **low resolution, multi-path interference, one-dimensional**
- Multi-sensor drowning detection **-> expensive**



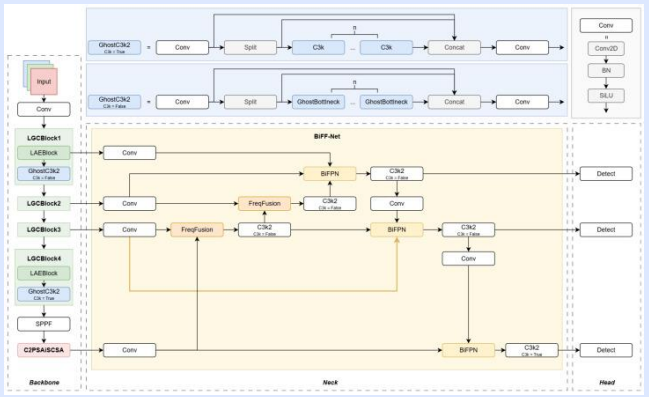
SwimSonar (He et al. 2022)



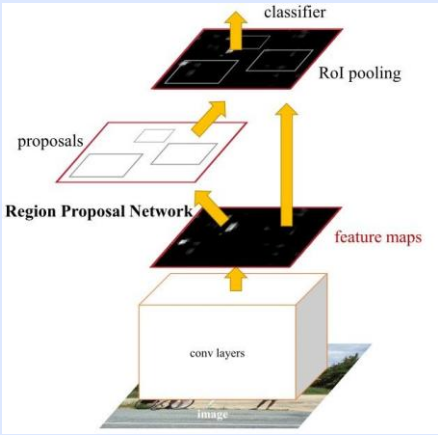
Multi-Sensor Drowning Detection (Jalalifar et al., 2022)

Current Deep Learning Models

- Zhang et al. (2025) created YOLO11-LiB, a lightweight model that effectively uses feature fusion to increase accuracy
  - Low FPS (~80)
  - FLOPS exceed limits for cheap edge devices
- He et al. (2023) used Faster R-CNN & YOLOv5
  - Low precision
  - Extremely low FPS (6)
  - Extreme parameters (190 x 10^6)



YOLO11-LiB (Zhang et al., 2025)



Faster R-CNN (Shaoqing et al., 2017)

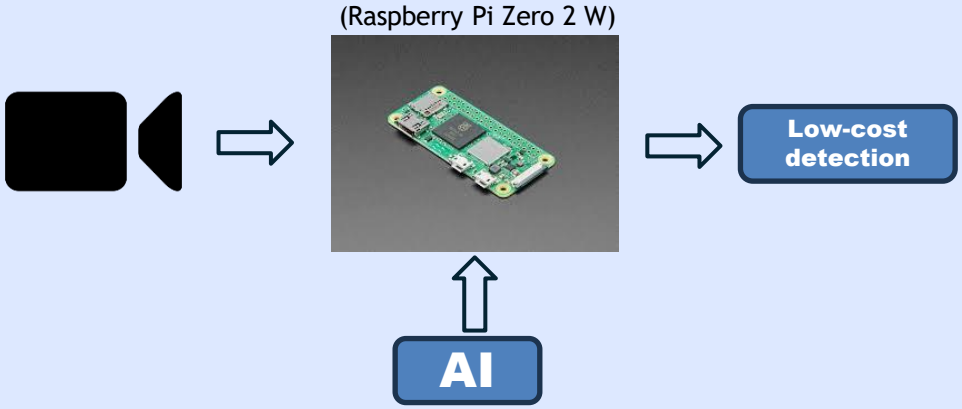
Goals, Challenges, and Solutions

Goals

1. Differentiate between drowning and swimming with **mAP50 >= 0.9, recall >= 0.85**
2. **Parameters <= 2,000,000** to reduce computation cost further than existing models
3. **GFLOPS <= 3** to be used effectively inside of edge devices (Raspberry Pi Zero 2 W)
4. **High FPS** for consistent, rapid, real-time detection

Challenges -> Solutions

1. Passive, low-cost drowning detection -> **Lightweight deep learning model using YOLO for computer vision**



2. Lightweight model -> **LGCBlock using Ghost Convolution**

3. Noise filtering & small object detection -> **Bi-Directional Feature Pyramid Network**

4. Robust model training -> **Extremely diverse dataset**

Data Collection

Overview

- Collected 2,000 original images from public dataset platforms and open internet resources (publicly available videos and images)

- Diverse dataset with images of:
  - indoor & outdoor pools
  - various lighting conditions
  - subjects of differing age groups
  - images captured by varying devices
  - differing perspectives

Splits and Labels

- Dataset split into training, validation, and testing images following a 7:2:1 ratio
- All images were labeled (drowning / swimming) in YOLO format through Roboflow annotation tools

<class_id> <x_center> <y_center> <width> <height>

Data augmentation

- Used to increase dataset diversity and reduce overfitting of model
- Including rotation, scaling, brightness adjustment, saturation variation, etc.

Example Images from dataset

Data processing

- Pytorch, Ultralytics YOLO
- Nvidia Tesla T4 GPU using Google Colab

Class distribution within training set

Drowning	Swimming
663	537

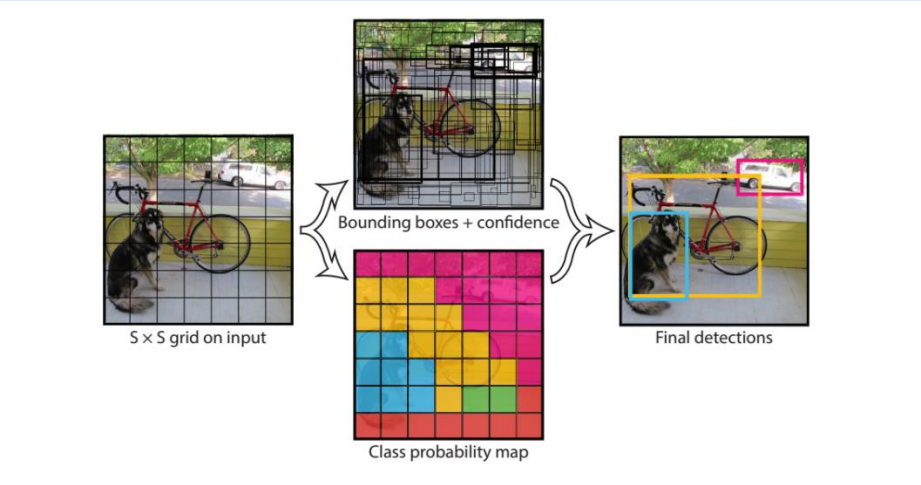
YOLO Overview

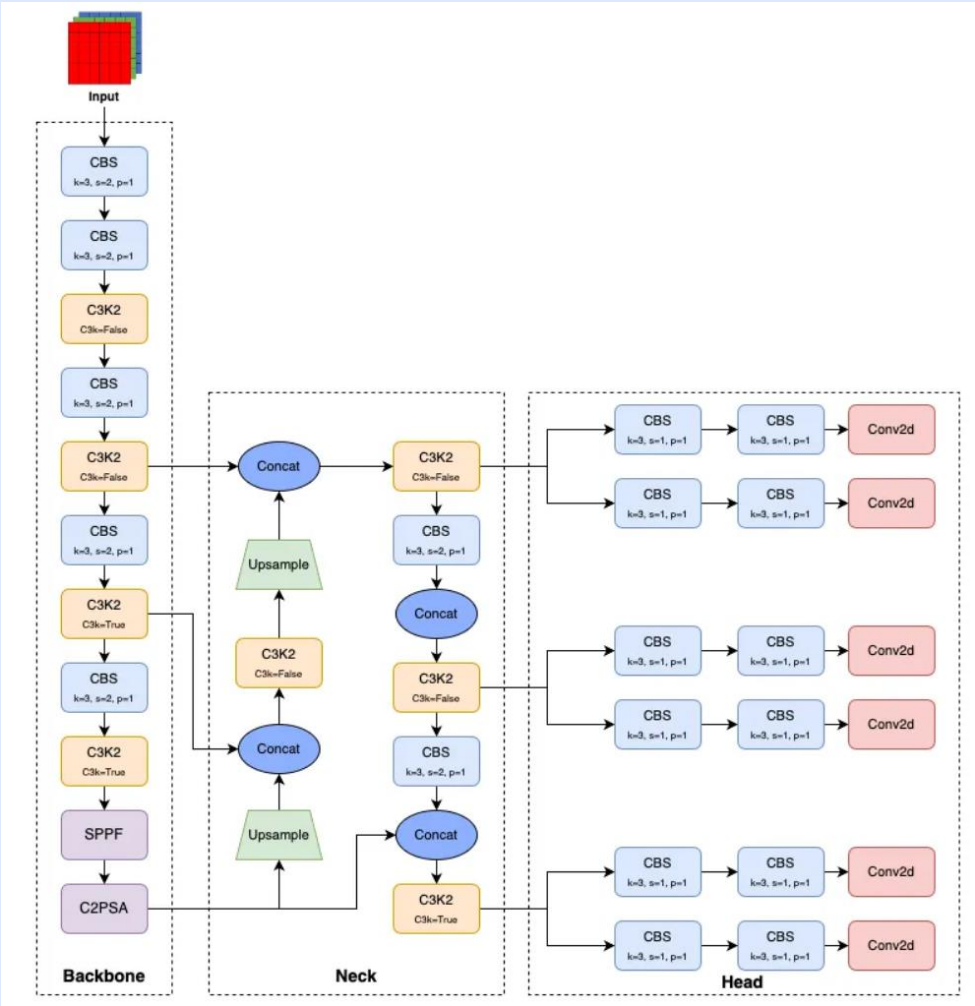
What is YOLO?

- Single-Stage Object Detection
  - Input Processing/Resizing
  - Grid Division
  - Feature Extraction using CNNs
  - Prediction: Bounding boxes & class probabilities
  - Post Processing: Combining bounding box with class probabilities -> single result

- **Consists of:**
  - Backbone (Feature Extraction)
  - Neck (Feature Fusion)
  - Head (Detection/Predictor)

- **Strengths:**
  - Much faster than Two-Stage models (such as RCNN)
  - Small models for embedded deployment





YOLO11 Architecture
(Ultralytics, 2024)

Custom Architecture In-Depth

Backbone (Feature Extraction)

- Optimized for speed
- Standard Convolution & C3k2 (Early stages): preserve high-resolution spatial details for small object detection
- **LGCBlock** (Later stages): Combines LAE and GhostC3k2 to balance lightweightedness with high-resolution feature retention (Zhang, 2025)

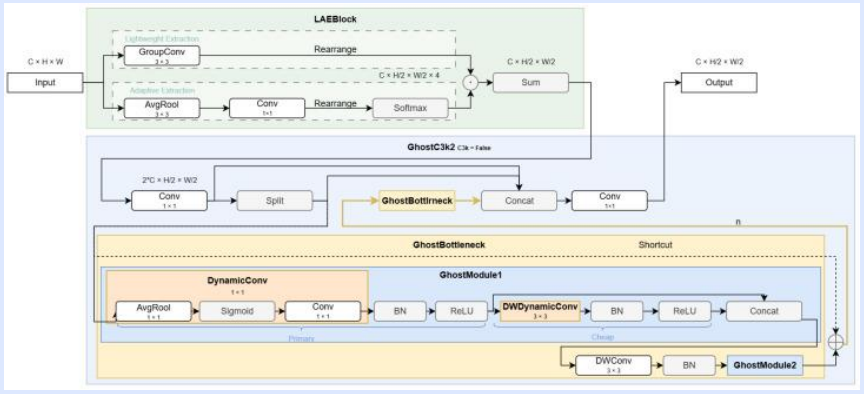
Neck (Feature Fusion)

- Optimized for accuracy
- **Bi-directional Feature Pyramid Network:** bidirectional cross-scale connections and weighted feature fusion (Tan et al., 2020)
- Robust small-object detection
- Modified for increased efficiency and lightweightedness

Head (Prediction)

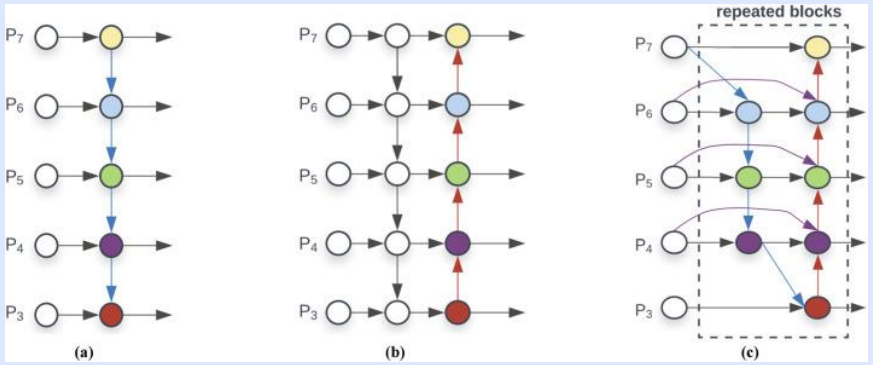
- Ultralytics YOLO26 detection head

(Ultralytics, 2026)

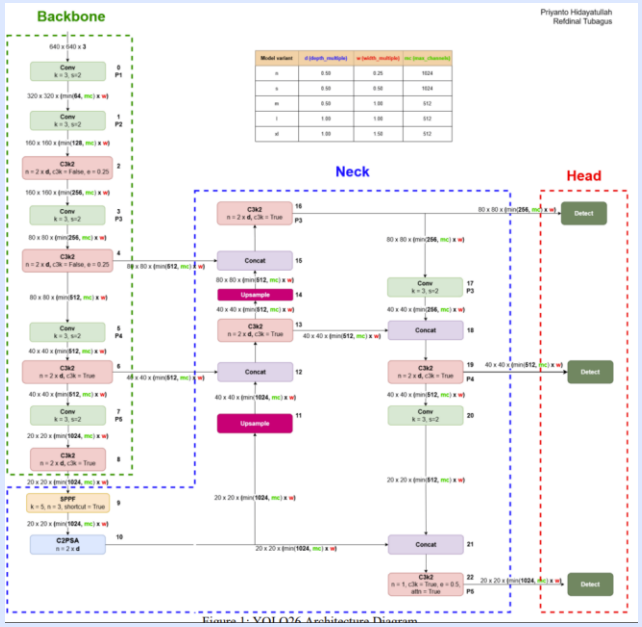


LGCBlock (Zhang, 2025)

(Tan et al., 2020)



(a) FPN, (b) PANet, (c) BiFPN



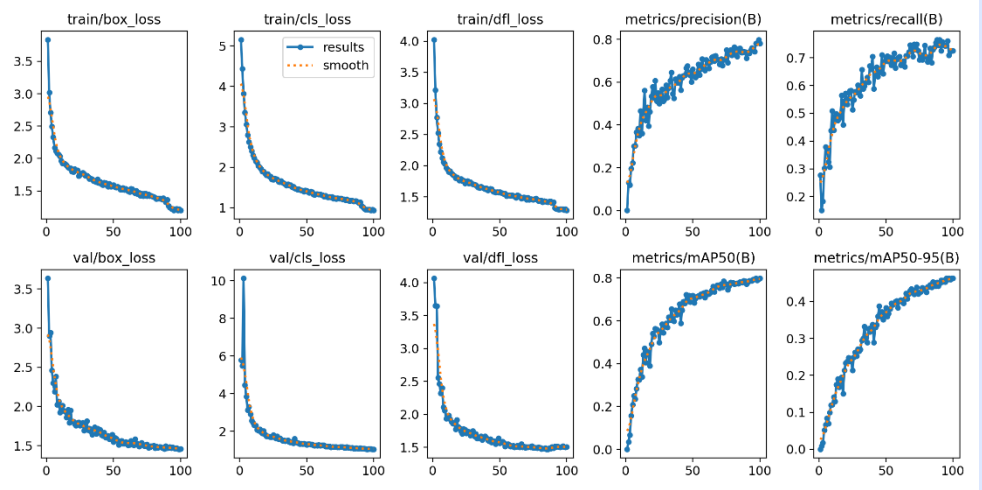
YOLO26 Architecture (Ultralytics, 2026)

Training and Evaluation

- Imperative to increase accuracy and reduce the number false negatives in drowning
  - False negatives (low drowning recall) → drowning labeled as swimming → potentially deadly
  - Low DmAP50 (drowning accuracy)→ false positives, low model confidence → inefficient system

- Model's performance is exceptional considering its reduced computational cost
  - Drowning Recall = **0.86**
  - Drowning mAP50 = **0.90**
  - **3.0** GFLOPS
  - **1,439,070** parameters
  - **2.95** MB
  - **256.4** FPS

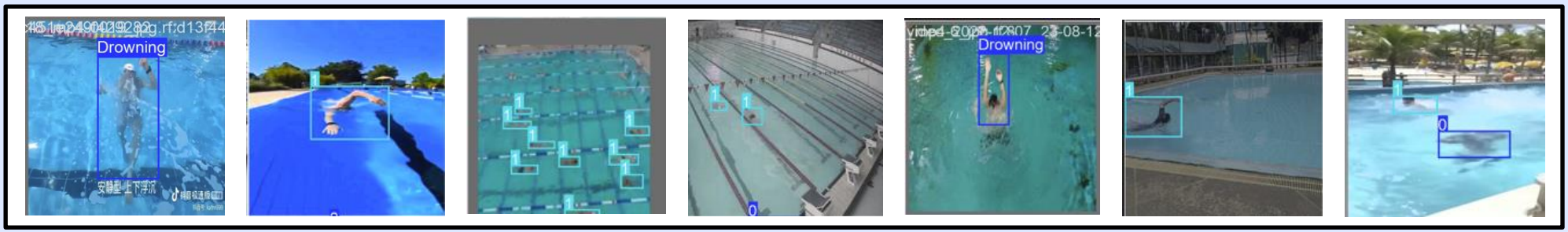
$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$



Training and Validation metrics (for all classes) over 100 epochs

Comparative experimental results

Model	Size (MB)	DR (%)	DmAP50 (%)	FLOPs (G)	FPS	Params (M)
YOLO11n	5.26	88.8	91.8	6.3	189.5	2.58
YOLO12n	5.31	83.3	90.0	6.3	126.2	2.56
Faster R-CNN	278	**91.5**	92.8	369.2	18.2	136.5
YOLO11-LiB	4.25	89.7	**94.1**	6.2	80.5	2.02
Ours	**2.95**	86.3	90.0	**3.0**	**256.4**	**1.44**

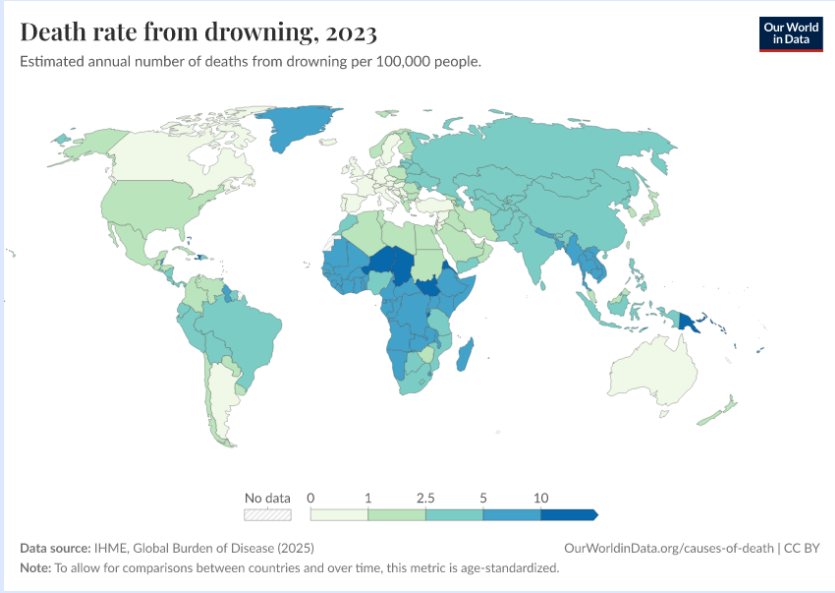


Example detection results
(0: Drowning, 1: Swimming)

Discussion and Limitations

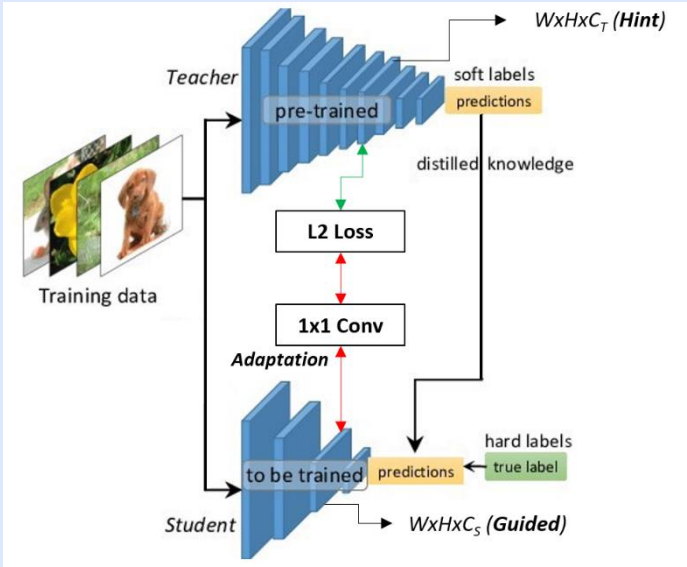
Discussion

- Lightweight model increases accessibility to low-income regions due to low deployment cost -> $30 drowning detection
- Effective even for small-object detection -> infant/toddler drowning prevention (vulnerable population)
- Faster detection than every other published deep-learning drowning detection model -> more time to prevent drowning
- High mAP50 and recall → **reliable** drowning detection system with high confidence in the system



Limitations

- Although dataset is diverse, all images are of pool environments -> over 70% of drowning deaths for teens aged 15-19 occur in open water
  - Solution: collect or simulate outdoor drowning/swimming dataset for training to increase model robustness
- Although recall and mAP50 are on par with modern detection models, can still be optimized further
  - Potential Solutions: By combining results of multiple heavy models, a compression technique can be used to distill knowledge to a single lightweight model (Hinton et al., 2015)



(Ganesh, 2019)

Implications and Future Research

Our Drowning Detection Model is **Directly Applicable**

- INO-YOLO is ready to be used inside edge devices for detection in/for:
  - Indoor / Outdoor Environments
  - Various lighting conditions
  - Various age groups
  - Multiple camera perspectives

- Cheaper drowning detection* for implementation in low-income regions to reduce drowning fatalities worldwide

*Price Breakdown

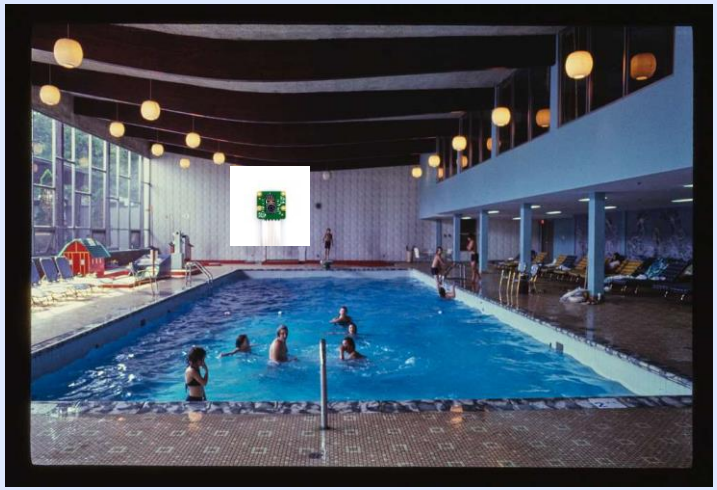
Item	Price
INO-YOLO	$0
Raspberry Pi Zero 2w	~$15.00
Raspberry Pi Camera Module V2	~$15.00
Total	**~$30.00**

Raspberry Pi Zero 2w

Raspberry Pi Camera Module V2

Future Research

- Lightweight pose estimation / temporal models for enhanced drowning detection
- Lightweight sensor fusion technology using sensors + computer vision
- Testing lightweight models using microcontroller edge devices, potentially even UAVs for beaches / oceans
- Multi-camera feature fusion for spatial data
- Better preprocessing methods / image enhancement for visual noise reduction in outdoor settings using simulated polarization, de-hazing, etc.



References

World Health Organization. “Drowning.” Who.int, World Health Organization: WHO, 13 Dec. 2024, www.who.int/news-room/fact-sheets/detail/drowning.

Zhang, Wenhui, et al. “A Pool Drowning Detection Model Based on Improved Yolo.” Sensors (Basel, Switzerland), U.S. National Library of Medicine, 5 Sept. 2025, pmc.ncbi.nlm.nih.gov/articles/PMC12431139/.

Hinton, Geoffrey, et al. Distilling the Knowledge in a Neural Network. 2015.

Tan, Mingxing, et al. “EfficientDet: Scalable and Efficient Object Detection.” ArXiv:1911.09070 [Cs, Eess], 27 July 2020, arxiv.org/abs/1911.09070.

Q. He, Z. Mei, H. Zhang and X. Xu, "Automatic Real-Time Detection of Infant Drowning Using YOLOv5 and Faster R-CNN Models Based on Video Surveillance," in Journal of Social Computing, vol. 4, no. 1, pp. 62-73, March 2023, doi: 10.23919/JSC.2023.0006.

Quang TN, Lee S, Song BC. Object Detection Using Improved Bi-Directional Feature Pyramid Network. Electronics. 2021; 10(6):746. https://doi.org/10.3390/electronics10060746

Wang, Kaixin, et al. “PANet: Few-Shot Image Semantic Segmentation with Prototype Alignment.” ArXiv:1908.06391 [Cs], 6 Feb. 2020, arxiv.org/abs/1908.06391.

Mittal, P. A comprehensive survey of deep learning-based lightweight object detection models for edge devices. Artif Intell Rev 57, 242 (2024). https://doi.org/10.1007/s10462-024-10877-1

“SwimEye.” MOBOTIX AG, 16 Mar. 2020, www.mobotix.com/en/swimeye. Accessed 22 Feb. 2026.

“Winner in Sport Equipment Design / Watersport, Aquatic, Paddle Sports WAVE Drowning Detection System.” FIT Sport Design Awards 2026, 12 Mar. 2023, fitdesignawards.com/winners/fit/3/1/3/. Accessed 22 Feb. 2026.

Ren, Shaoqing, et al. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, 1 June 2017, pp. 1137–1149, https://doi.org/10.1109/tpami.2016.2577031.

Ganesh, Prakhar. “Knowledge Distillation : Simplified | towards Data Science.” *Towards Data Science*, 12 Aug. 2019, towardsdatascience.com/knowledge-distillation-simplified-dd4973dbc764/.