Underwater Segmentation Hybrid Approach

PRESENTED BY-

SHREYASH GAIKWAD
2203131

SUPERVISED BY-

Dr. SHEETALA PRASAD
Dr. ANUJ ABRAHAM

PROBLEM STATEMENT

Underwater object segmentation is challenged by low visibility, light distortion, and color degradation. We need to propose a deep learning-based approach using enhanced U-Net architecture for the same. This work supports robust marine monitoring and autonomous underwater exploration.

MODEL ARCHITECTURE

TimmHybridNet Architecture

1. Combines a powerful Swin Transformer encoder with a lightweight U-Net style decoder.
2. Extracts multi-scale hierarchical features that capture both fine textures and global context.
3. Employs skip connections to preserve spatial details and refine object boundaries.
4. High-level idea: Encoder (pretrained backbone) → decoder (upsampling + skip connections) → per-pixel class scores.

ENCODER: SWIN TRANSFORMER

1. Uses swin_base_patch4_window7_224.ms_in22k pretrained on ImageNet-22k.
2. Captures global context through transformer attention and local detail via hierarchical windowing.
3. Produces four levels of feature maps (C1–C4) with increasing semantic depth.
4. Strong encoder helps overcome underwater challenges like haze, color shift, and low contrast.

ENCODER SPECIFICATIONS

Encoder: Swin Transformer (Base) Code Name: swin_base_patch4_window7_224.ms_in22k

- "Swin Base": A powerful backbone with 88 Million parameters (much stronger than ResNet-50).
- "Patch4": We process the image in tiny 4x4 patches to keep small details sharp.
- "Window7": Uses a 7x7 window attention mechanism to focus on local textures (scales).
- "ms_in22k": The Secret Weapon.
  - Pre-trained on ImageNet-22K (14 million images, 22,000 classes).

HOW IT WORKS: THE "SWIN" MECHANISM

1. Patch Partitioning (The Starting Point)
   - Input image split into tiny, non-overlapping squares (e.g., 4×4 pixels).
   - Each patch becomes a token (like a word in NLP).
2. Window-Based Attention (Local Focus)
   - Standard Transformers: every pixel compares with every other → very slow.
   - Swin Transformer: groups patches into windows (e.g., 7×7 patches).
   - Attention computed only within each window → linear complexity.

3. Shifted Windowing (The “Magic” Step)

- Problem: local windows can’t see across boundaries.
- Solution: Shift the window grid in the next layer.
- Result: connects information across windows efficiently.

4. Hierarchical Feature Maps

- Patch Merging reduces resolution as network goes deeper.
- Creates a feature pyramid: C1 → C2 → C3 → C4.
- Captures fine details to global context for downstream tasks.

MULTI-SCALE FEATURE REPRESENTATION

Hierarchical Feature Pyramid (C1 → C4)

- C1: High-resolution spatial details (edges, outlines).
- C2: Mid-level textures and object parts.
- C3: Semantic structures of underwater objects (fish, reefs, wrecks).
- C4: Global scene understanding for large regions like sea-floor or background.
- Multi-scale representation enables accurate segmentation of both large and small underwater objects.

DECODER: U-NET STYLE UPSAMPLING PATH

Lightweight and Efficient Reconstruction

1. Three upsampling blocks, each containing:
   a. 3×3 Convolution
   b. Batch Normalization
   c. ReLU Activation
   d. Bilinear Upsampling (×2)
2. Skip connections inject encoder features at each level, preserving detail.
3. Feature addition (not concatenation) keeps parameters low while retaining spatial accuracy.
4. Decoder gradually rebuilds high-resolution segmentation maps.

FINAL SEGMENTATION HEAD

Refinement and Output Generation

1. ConvTranspose layer performs coarse resolution recovery.
2. Followed by BatchNorm and ReLU for stable refinement.
3. Dropout (0.2) protects against overfitting on small underwater datasets.
4. 1×1 Convolution produces final per-pixel class predictions (8 classes).
5. Output is resized to the target resolution for clean boundary alignment.

LOSS FUNCTIONS USED

- Dice Loss

  G = Ground truth mask

  M = Prdicted mask

$$L_{Dice}(M, G) = 1 - \frac{2|M \cap G|}{|M| + |G|}$$

- Focal Loss (for difficult pixels)

  M = Prdicted mask

  α = class-balancing factor

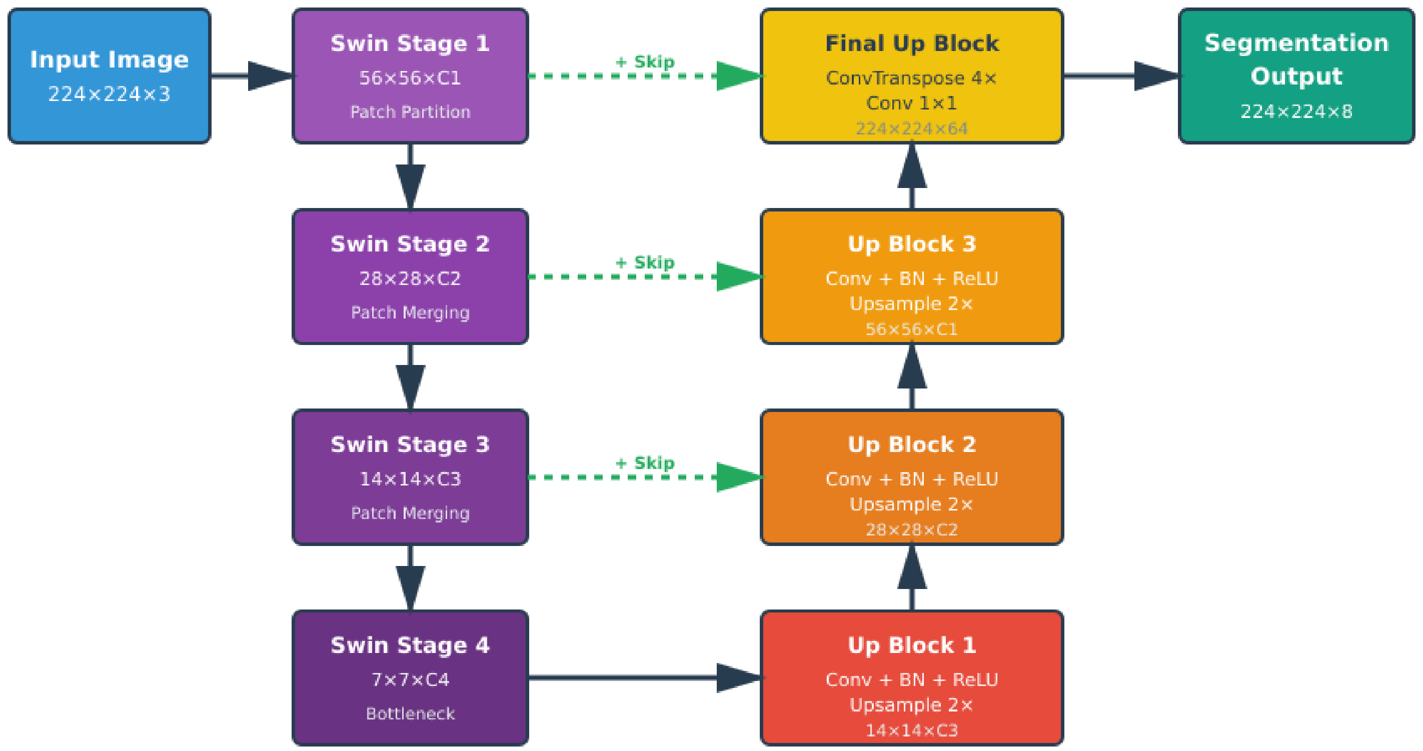
  γ = focusing parameter

$$L_{Focal} = -\alpha(1 - M)^{\gamma} \log(M)$$

- Combined Loss

  0.5*Focal loss + 1* Dice loss

MODEL ARCHITECTURE



Input Image
224×224×3

Swin Stage 1
56×56×C1
Patch Partition

Swin Stage 2
28×28×C2
Patch Merging

Swin Stage 3
14×14×C3
Patch Merging

Swin Stage 4
7×7×C4
Bottleneck

+ Skip

+ Skip

+ Skip

Up Block 1
Conv + BN + ReLU
Upsample 2×
14×14×C3

Up Block 2
Conv + BN + ReLU
Upsample 2×
28×28×C2

Up Block 3
Conv + BN + ReLU
Upsample 2×
56×56×C1

Final Up Block
ConvTranspose 4×
Conv 1×1
224×224×64

Segmentation Output
224×224×8

Model Details

Key Features:

- Backbone: Swin-Base Transformer
- Pretrained: ImageNet-22k (21,000 classes)
- Decoder: UNet-style with skip connections
- Input Size: 224×224×3
- Output Classes: 8 (underwater objects)

Loss Function:

- Combo Loss (Focal + Dice)
- Focal: α=1, γ=2
- Weight: 0.5×Focal + 1.0×Dice

Training:

- Optimizer: AdamW (lr=5e-5, wd=1e-2)
- Scheduler: CosineAnnealing
- Augmentation: ColorJitter, Flip H/V
- TTA: Horizontal flip averaging
- Dropout: 0.2 in final decoder

Performance:

✓ **mIoU: 72% on SUIM dataset**

Segmentation Classes

⬛	0: Background	🟥	4: Robots
🟦	1: Human Divers	🟪	5: Reefs
🟩	2: Plants	🟨	6: Fish
🟦	3: Wrecks	□	7: Sea-floor

Data Pipeline:

Image → Resize → ColorJitter → Flip
→ Normalize (ImageNet stats)
Mask → Resize (NEAREST) → RGB to Index
Validation: TTA with horizontal flip
Batch Size: 8 | Workers: 2

DATASET USED

- **SUIM Dataset (Semantic Underwater Imagery)**
- link of the dataset : https://www.kaggle.com/datasets/ashish2001/semantic-segmentation-of-underwater-imagery-suim

- **Fish Recognition Dataset (Fish4Knowledge)**
- link of the dataset : https://www.kaggle.com/datasets/madhushreesannigrahi/fish-recognition-ground-truth-data

SUIM DATASET

Properties :

- Contains ~1,525 images with pixel-wise annotations.
- Covers 8 underwater object classes:
- Background, Human Divers, Plants, Wrecks, Robots, Reefs, Fish, Sea-floor.
- Images captured under varying lighting, turbidity, and depth.
- Provides 110 dedicated test images for benchmarking.
- Serves as an official benchmark for underwater segmentation tasks.

Challenges in SUIM dataset :

- Significant color degradation (green/blue cast).
- Low visibility due to turbidity and suspended particles.
- High class imbalance, especially Plants, Robots, and Wrecks.
- Objects vary greatly in shape and size (tiny fish vs. large reefs).
- Complex backgrounds often blend with target objects.

FISH RECOGNITION DATASET

Properties :

- Contains 27,370 fish images across 23 species.
- Highly imbalanced: some species have 1000× more samples than others.
- Used to validate generalization of the model beyond SUIM.
- Training set (80%) → 21,896 images, Validation set (10%) → 2,737 images, Test set (10%) → 2,737 images
- Helps ensure the architecture works on large-scale underwater datasets.

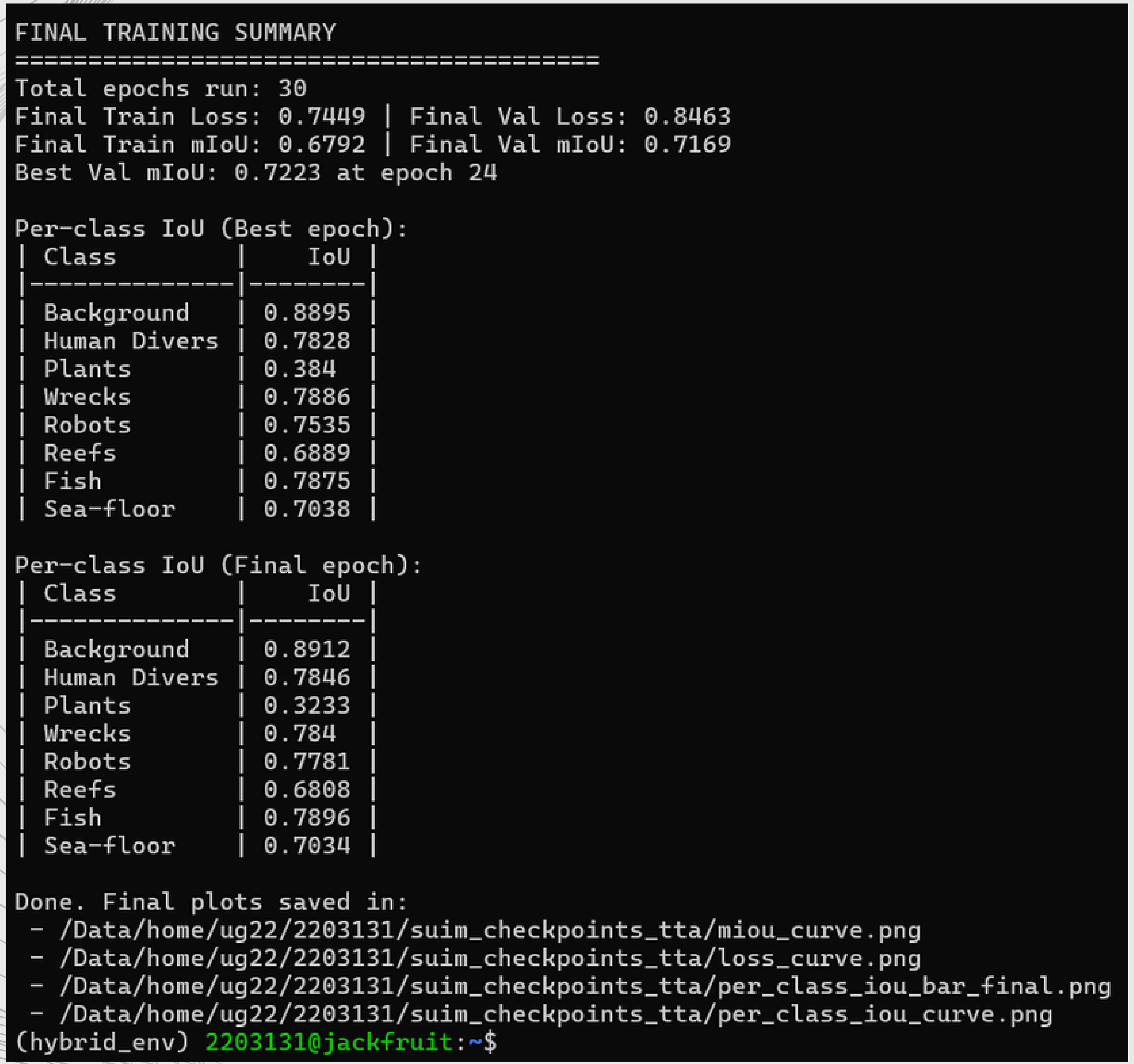
Challenges in this dataset :

- Severe class imbalance – some species have thousands of samples while others have very few.
- High visual similarity between species, making class boundaries hard to distinguish.
- Inconsistent lighting conditions underwater affecting color and texture patterns.
- Occlusions and motion blur due to fish movement and underwater currents.

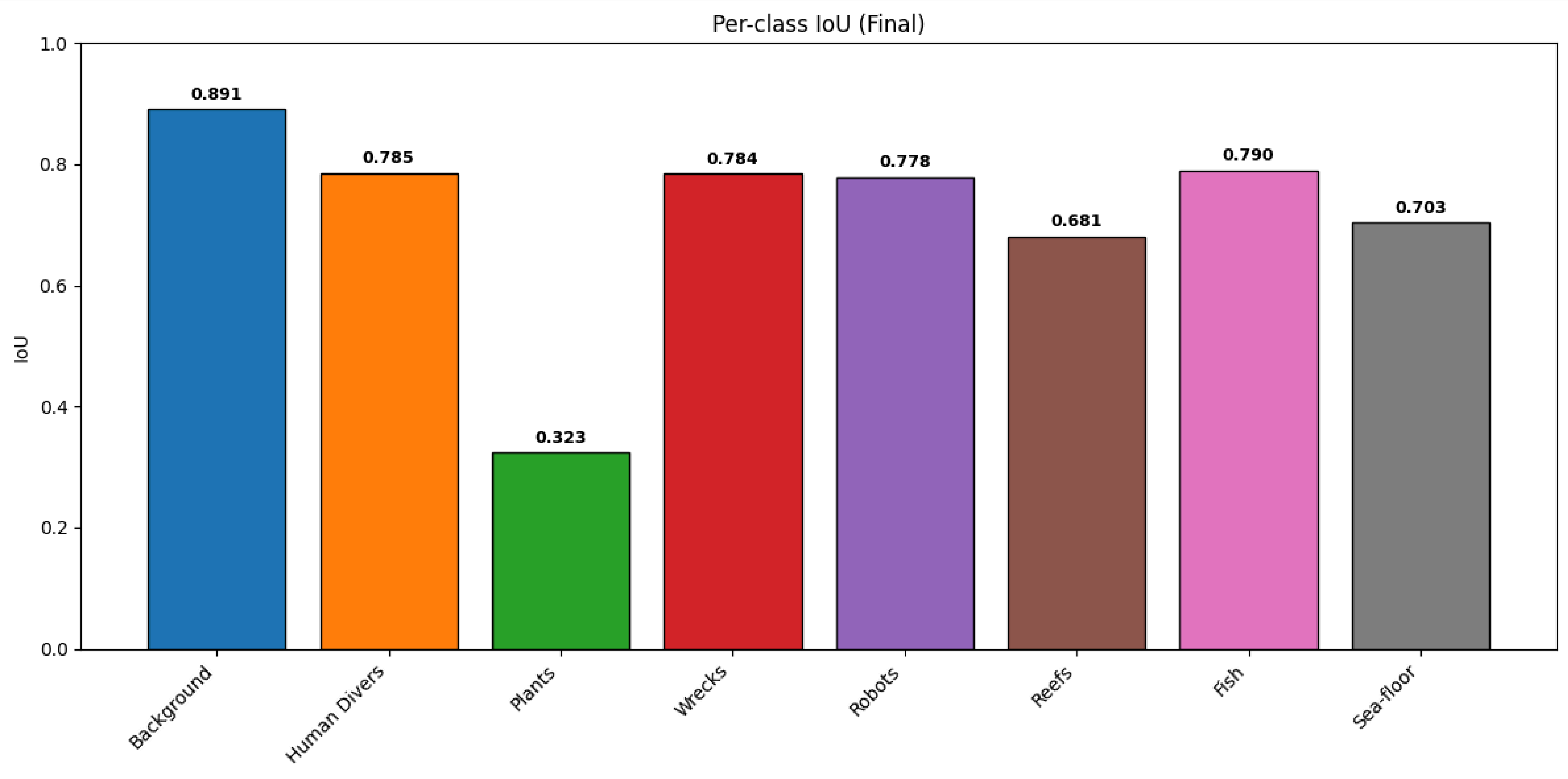
TRAINING PARAMETERS

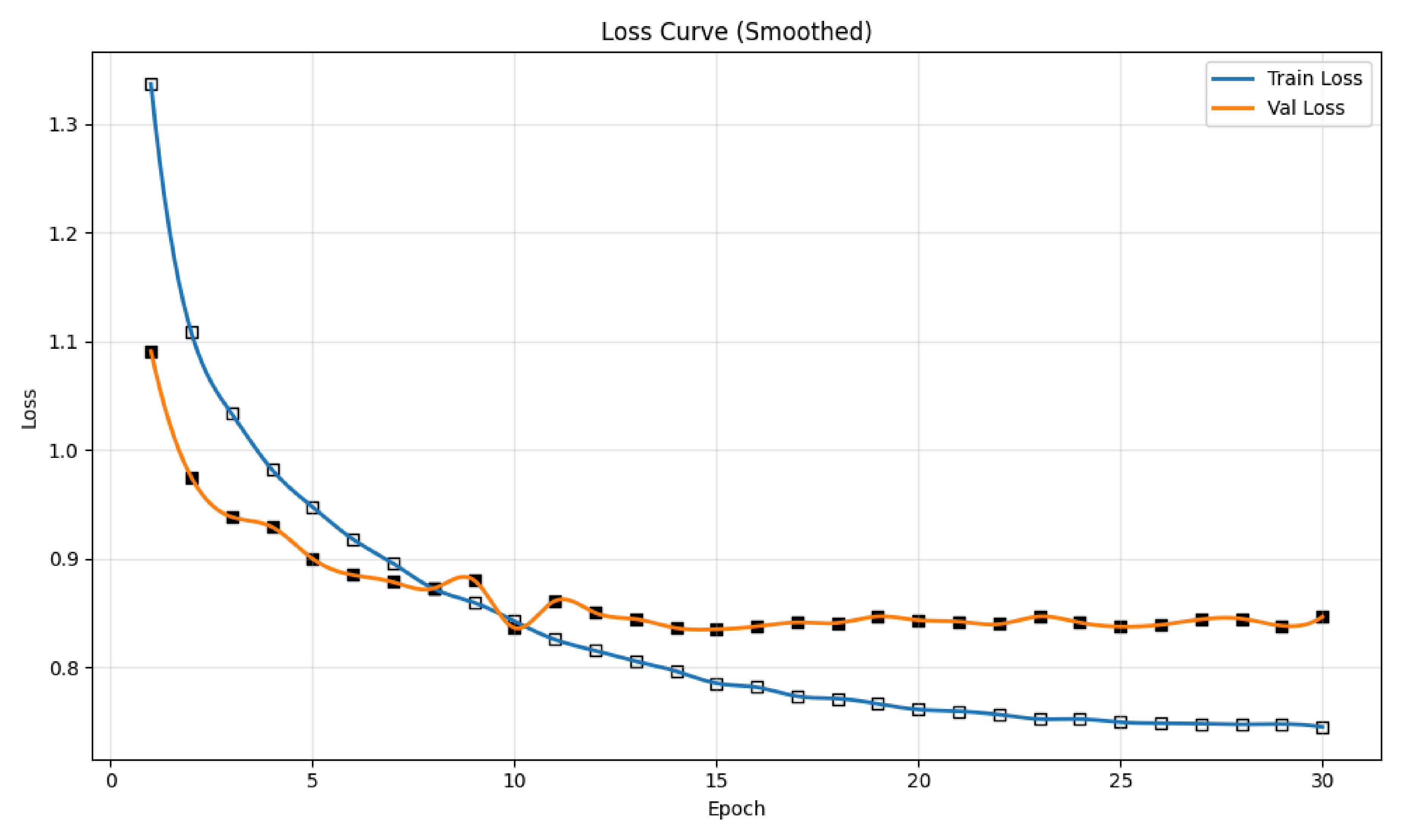
- Backbone: swin_base_patch4_window7_224.ms_in22k (timm, pretrained In22k)
- Input Size: 224 × 224
- Batch Size: 8
- Epochs: 30
- Optimizer: AdamW
- Learning Rate: 5×10^{-5}
- Weight Decay: 1×10^{-2}
- Scheduler: CosineAnnealingLR (T_max = epochs, eta_min = 1e−6)
- Regularization: Dropout (0.2), BatchNorm after all conv layers
- Loss Function: ComboLoss = 0.5×Focal(gamma=2, α=1) + DiceLoss
- Validation: Test-Time Augmentation (horizontal flip, averaged logits)
- Data Augmentation: ColorJitter + Random H/V flips
- DataLoader: num_workers = 2, pin_memory = True
- Reproducibility: seed = 42, deterministic cuDNN (when supported)
- Device: GPU for training; inference with .eval() and dropout disabled

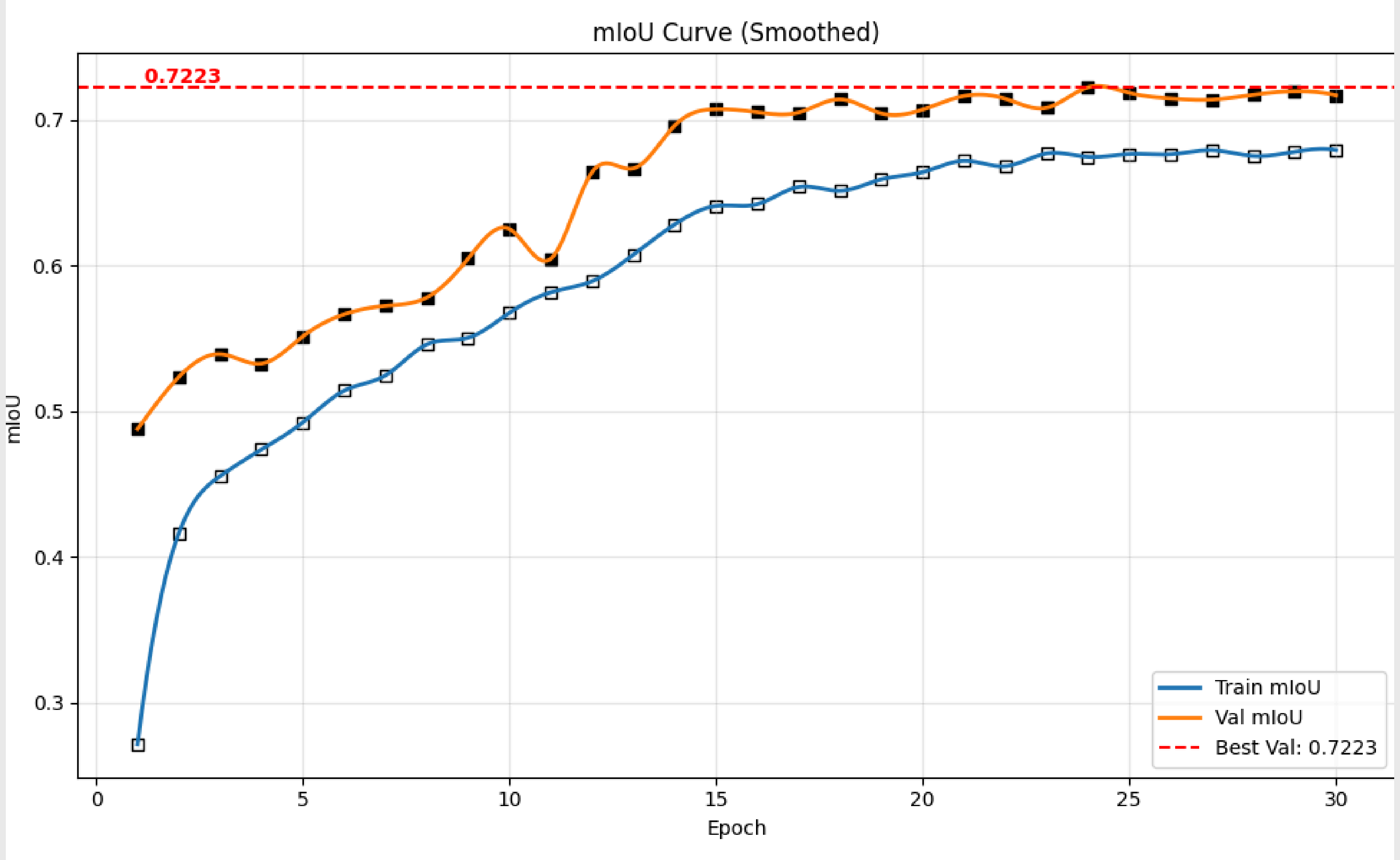
TRAINING RESULT ON SUIM

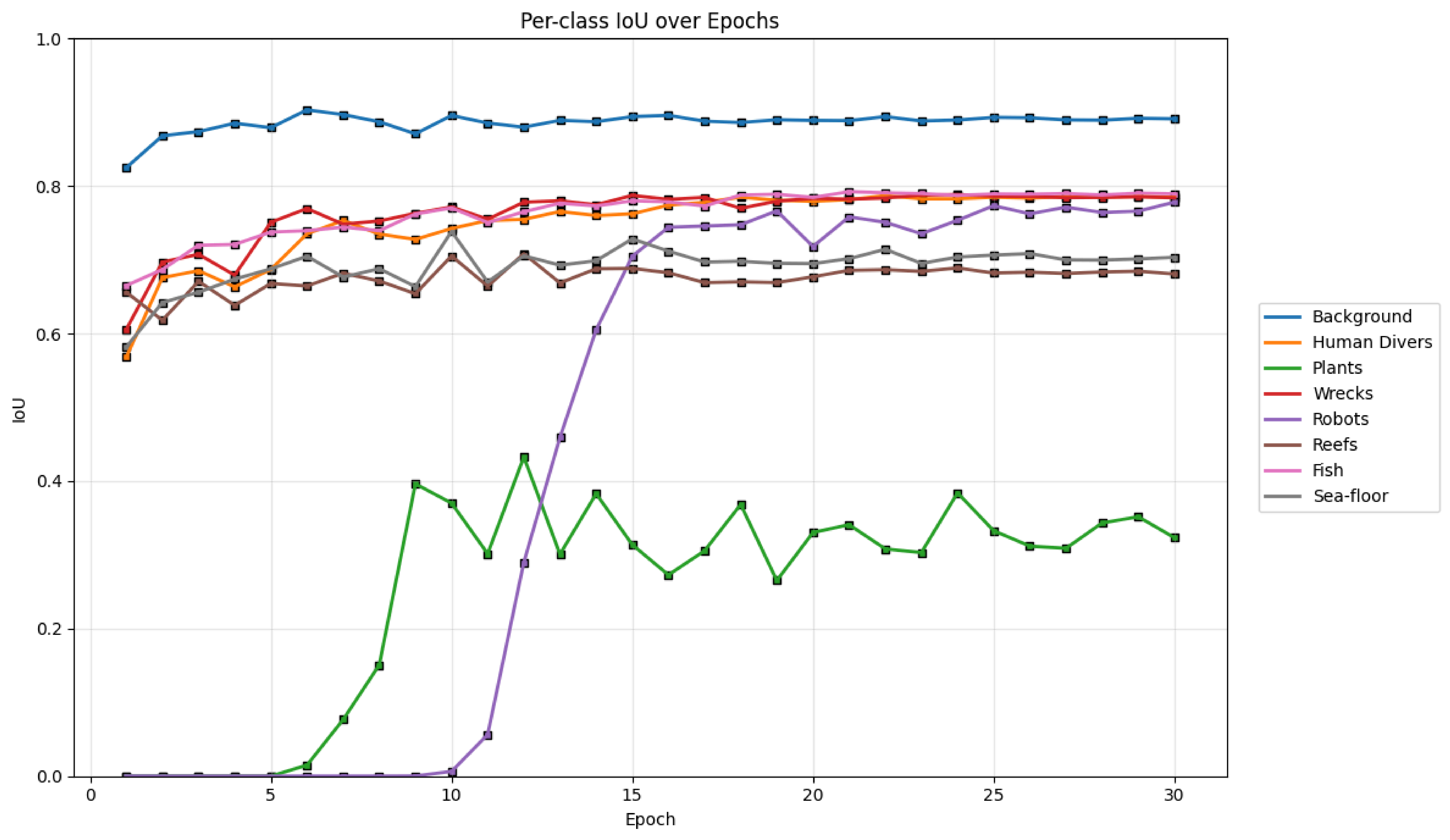


```
FINAL TRAINING SUMMARY
========================================
Total epochs run: 30
Final Train Loss: 0.7449 | Final Val Loss: 0.8463
Final Train mIoU: 0.6792 | Final Val mIoU: 0.7169
Best Val mIoU: 0.7223 at epoch 24

Per-class IoU (Best epoch):
| Class        |    IoU |
|--------------|--------|
| Background   | 0.8895 |
| Human Divers | 0.7828 |
| Plants       | 0.384  |
| Wrecks       | 0.7886 |
| Robots       | 0.7535 |
| Reefs        | 0.6889 |
| Fish         | 0.7875 |
| Sea-floor    | 0.7038 |

Per-class IoU (Final epoch):
| Class        |    IoU |
|--------------|--------|
| Background   | 0.8912 |
| Human Divers | 0.7846 |
| Plants       | 0.3233 |
| Wrecks       | 0.784  |
| Robots       | 0.7781 |
| Reefs        | 0.6808 |
| Fish         | 0.7896 |
| Sea-floor    | 0.7034 |

Done. Final plots saved in:
 - /Data/home/ug22/2203131/suim_checkpoints_tta/miou_curve.png
 - /Data/home/ug22/2203131/suim_checkpoints_tta/loss_curve.png
 - /Data/home/ug22/2203131/suim_checkpoints_tta/per_class_iou_bar_final.png
 - /Data/home/ug22/2203131/suim_checkpoints_tta/per_class_iou_curve.png
(hybrid_env) 2203131@jackfruit:~$
```

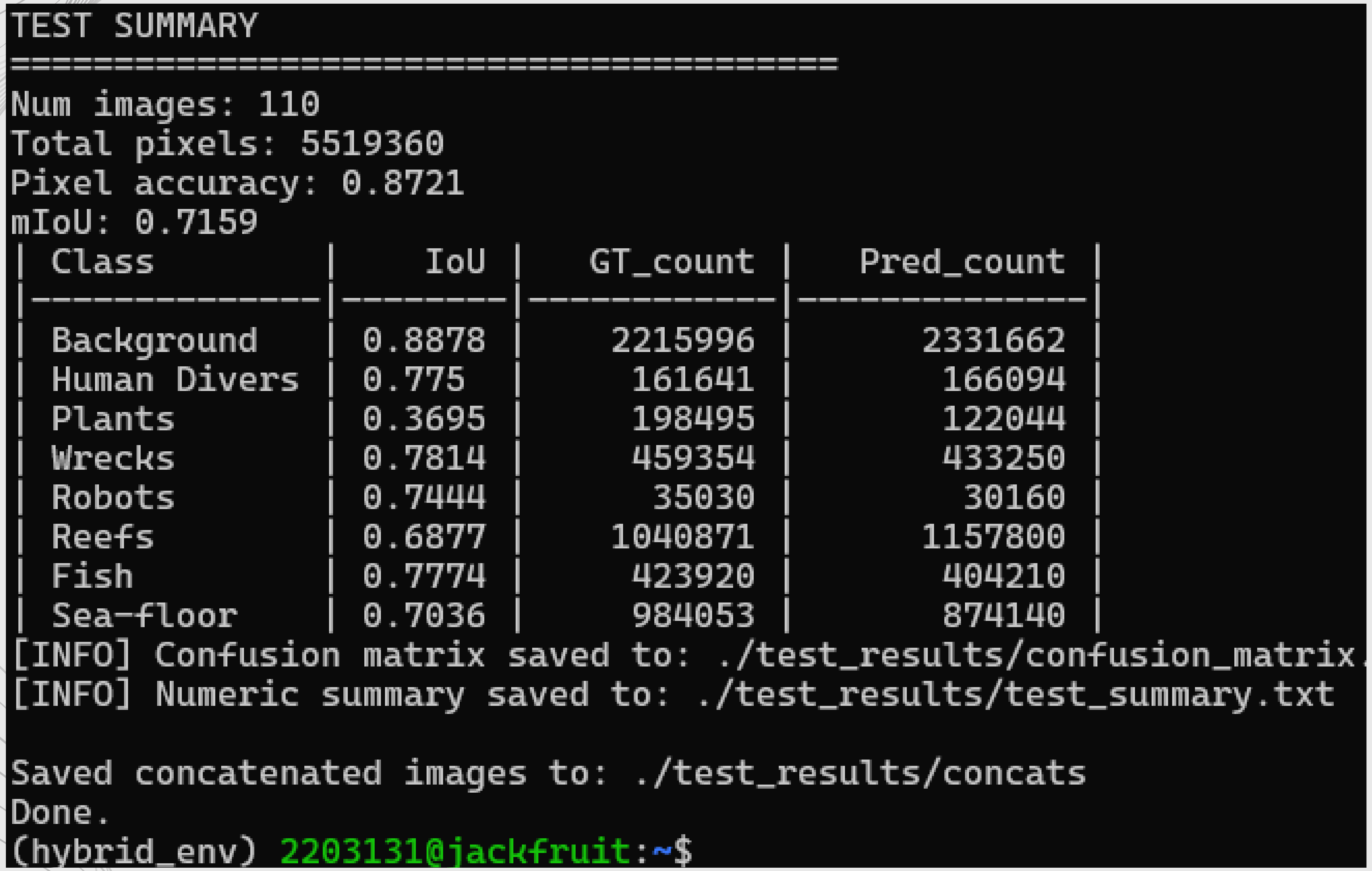



Loss Curve (Smoothed)
Train Loss
Val Loss
Loss
1.3
1.2
1.1
1.0
0.9
0.8
0
5
10
15
20
25
30
Epoch

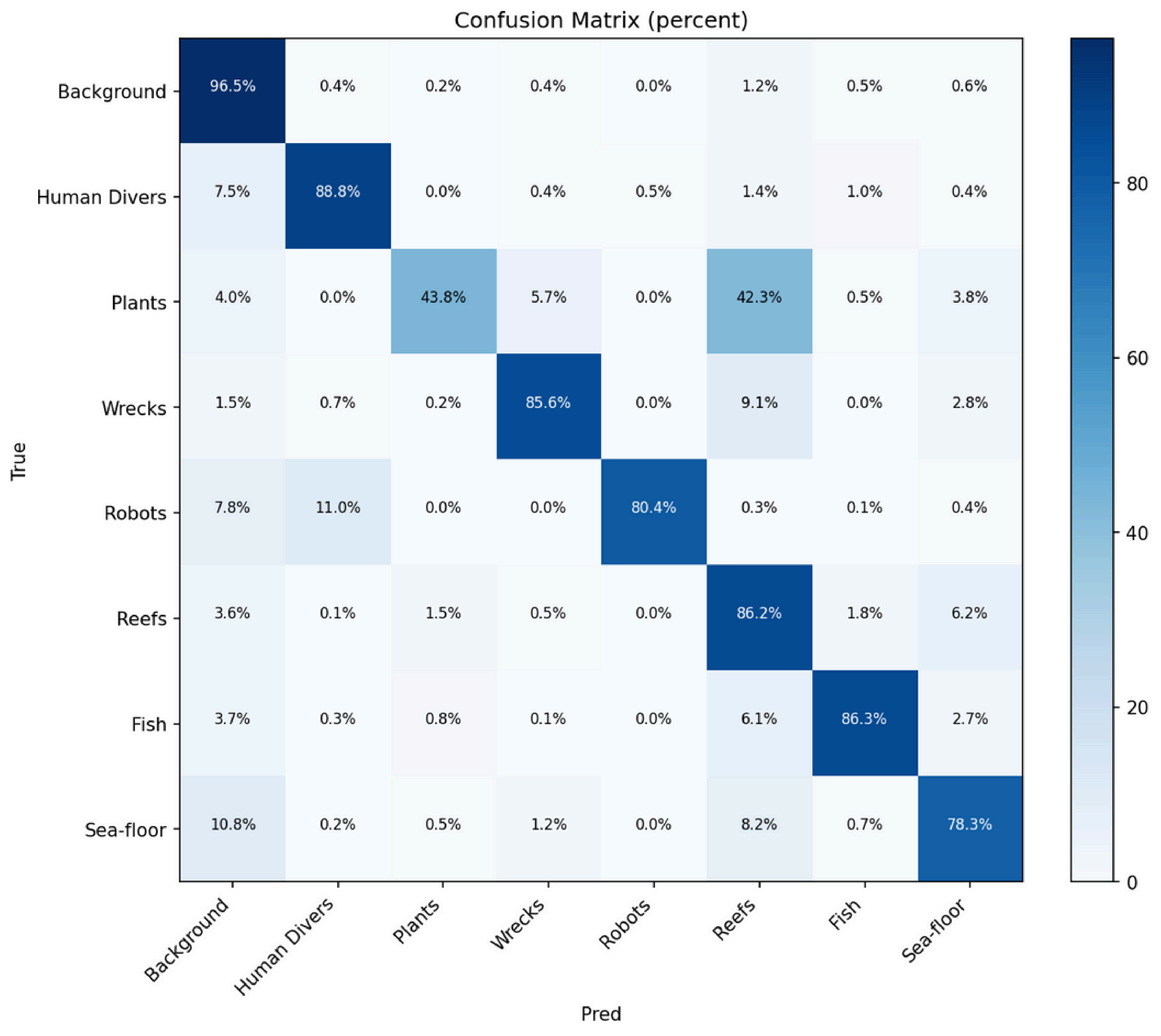

mIoU Curve (Smoothed)
0.7223
mIoU
0.7
0.6
0.5
0.4
0.3
Train mIoU
Val mIoU
Best Val: 0.7223
0
5
10
15
20
25
30
Epoch


Per-class IoU over Epochs
IoU
Epoch
0.0
0.2
0.4
0.6
0.8
1.0
0
5
10
15
20
25
30
Background
Human Divers
Plants
Wrecks
Robots
Reefs
Fish
Sea-floor

TEST RESULTS ON SUIM



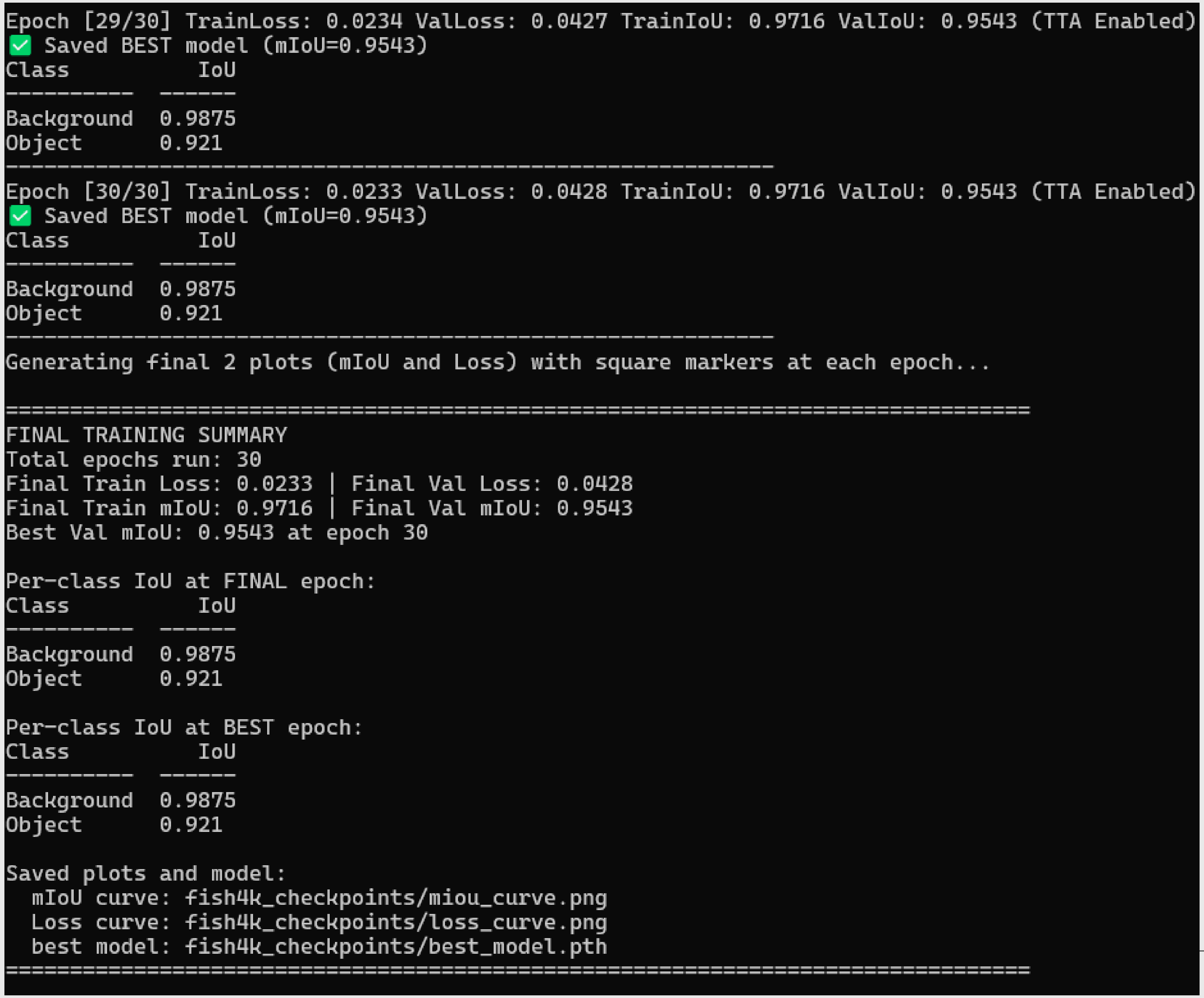
```
TEST SUMMARY
========================================
Num images: 110
Total pixels: 5519360
Pixel accuracy: 0.8721
mIoU: 0.7159
| Class         |    IoU |   GT_count |   Pred_count |
|---------------|--------|------------|--------------|
| Background    | 0.8878 |    2215996 |      2331662 |
| Human Divers  | 0.775  |     161641 |       166094 |
| Plants        | 0.3695 |     198495 |       122044 |
| Wrecks        | 0.7814 |     459354 |       433250 |
| Robots        | 0.7444 |      35030 |        30160 |
| Reefs         | 0.6877 |    1040871 |      1157800 |
| Fish          | 0.7774 |     423920 |       404210 |
| Sea-floor     | 0.7036 |     984053 |       874140 |
[INFO] Confusion matrix saved to: ./test_results/confusion_matrix.
[INFO] Numeric summary saved to: ./test_results/test_summary.txt

Saved concatenated images to: ./test_results/concats
Done.
(hybrid_env) 2203131@jackfruit:~$
```



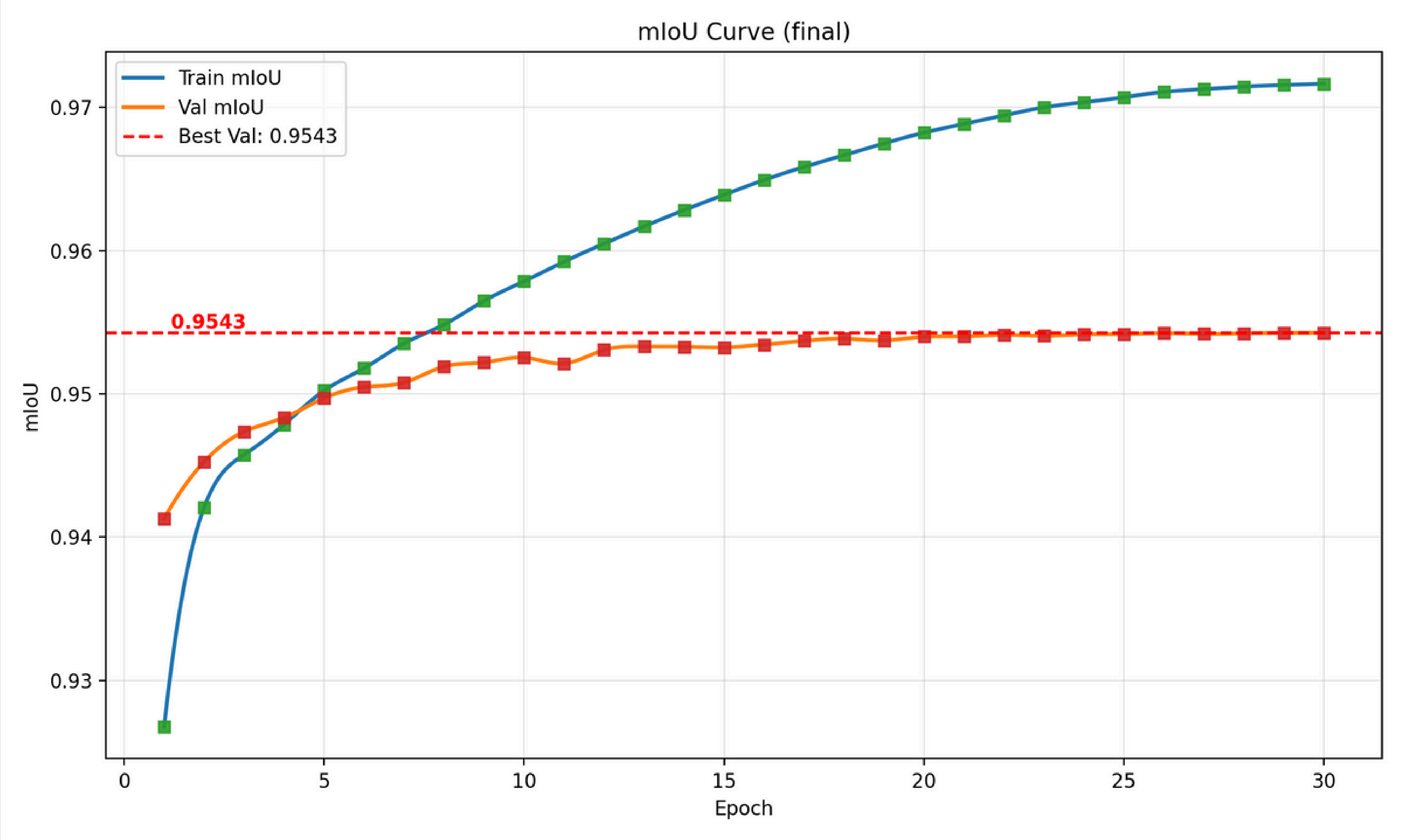
Confusion Matrix (percent)

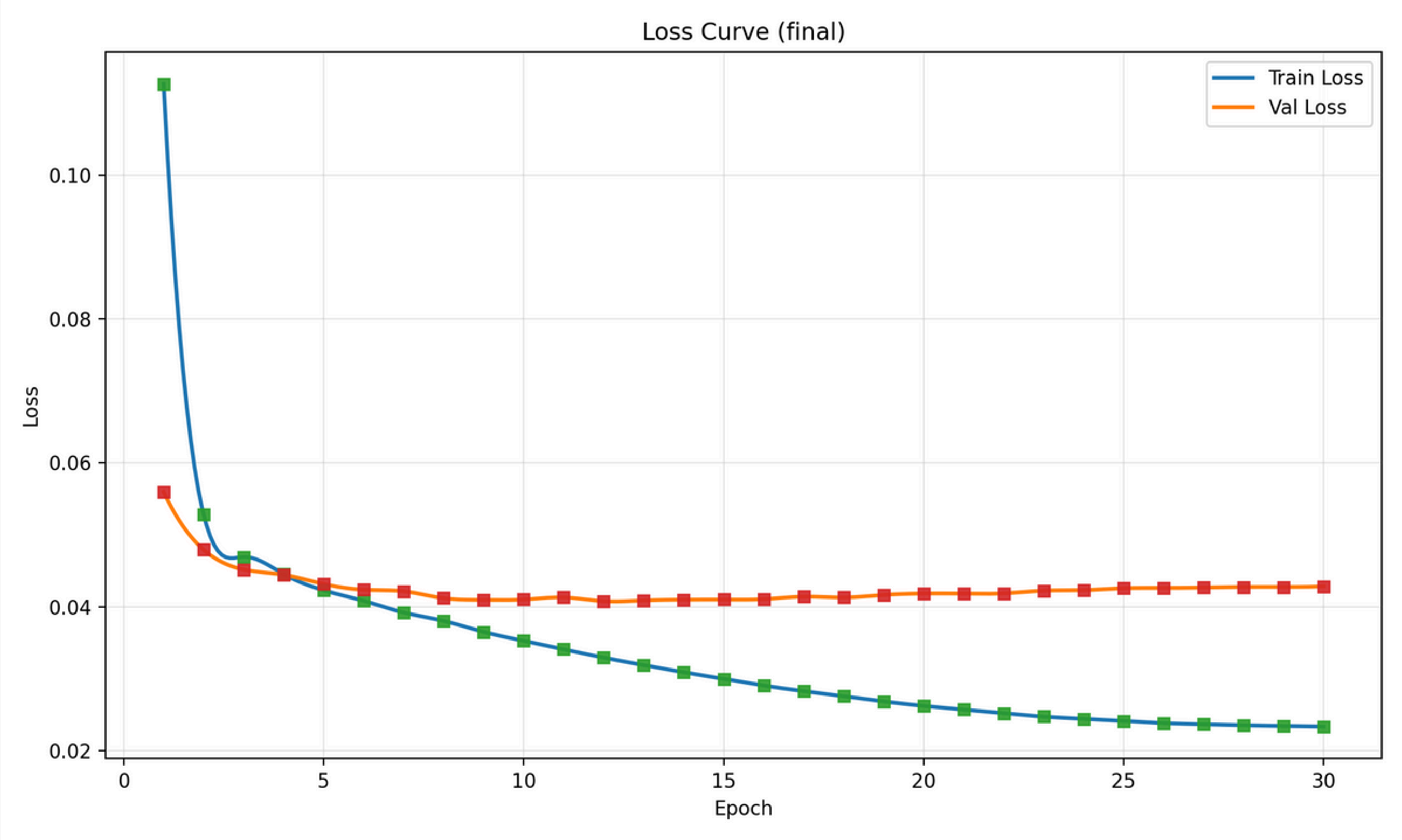
True \ Pred	Background	Human Divers	Plants	Wrecks	Robots	Reefs	Fish	Sea-floor
Background	96.5%	0.4%	0.2%	0.4%	0.0%	1.2%	0.5%	0.6%
Human Divers	7.5%	88.8%	0.0%	0.4%	0.5%	1.4%	1.0%	0.4%
Plants	4.0%	0.0%	43.8%	5.7%	0.0%	42.3%	0.5%	3.8%
Wrecks	1.5%	0.7%	0.2%	85.6%	0.0%	9.1%	0.0%	2.8%
Robots	7.8%	11.0%	0.0%	0.0%	80.4%	0.3%	0.1%	0.4%
Reefs	3.6%	0.1%	1.5%	0.5%	0.0%	86.2%	1.8%	6.2%
Fish	3.7%	0.3%	0.8%	0.1%	0.0%	6.1%	86.3%	2.7%
Sea-floor	10.8%	0.2%	0.5%	1.2%	0.0%	8.2%	0.7%	78.3%

TRAINING RESULTS ON FISH_4K

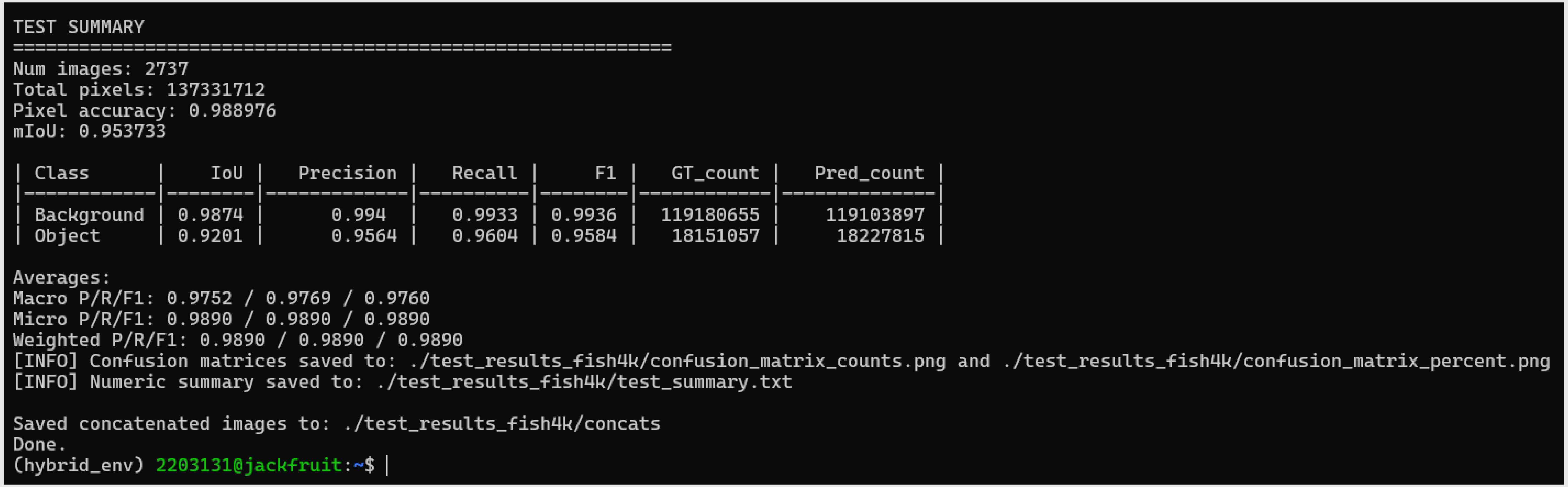


```
Epoch [29/30] TrainLoss: 0.0234 ValLoss: 0.0427 TrainIoU: 0.9716 ValIoU: 0.9543 (TTA Enabled)
✅ Saved BEST model (mIoU=0.9543)
Class           IoU
----------  ------
Background  0.9875
Object      0.921
------------------------------------------------------------
Epoch [30/30] TrainLoss: 0.0233 ValLoss: 0.0428 TrainIoU: 0.9716 ValIoU: 0.9543 (TTA Enabled)
✅ Saved BEST model (mIoU=0.9543)
Class           IoU
----------  ------
Background  0.9875
Object      0.921
------------------------------------------------------------
Generating final 2 plots (mIoU and Loss) with square markers at each epoch...

================================================================================
FINAL TRAINING SUMMARY
Total epochs run: 30
Final Train Loss: 0.0233 | Final Val Loss: 0.0428
Final Train mIoU: 0.9716 | Final Val mIoU: 0.9543
Best Val mIoU: 0.9543 at epoch 30

Per-class IoU at FINAL epoch:
Class           IoU
----------  ------
Background  0.9875
Object      0.921

Per-class IoU at BEST epoch:
Class           IoU
----------  ------
Background  0.9875
Object      0.921

Saved plots and model:
  mIoU curve: fish4k_checkpoints/miou_curve.png
  Loss curve: fish4k_checkpoints/loss_curve.png
  best model: fish4k_checkpoints/best_model.pth
================================================================================
```

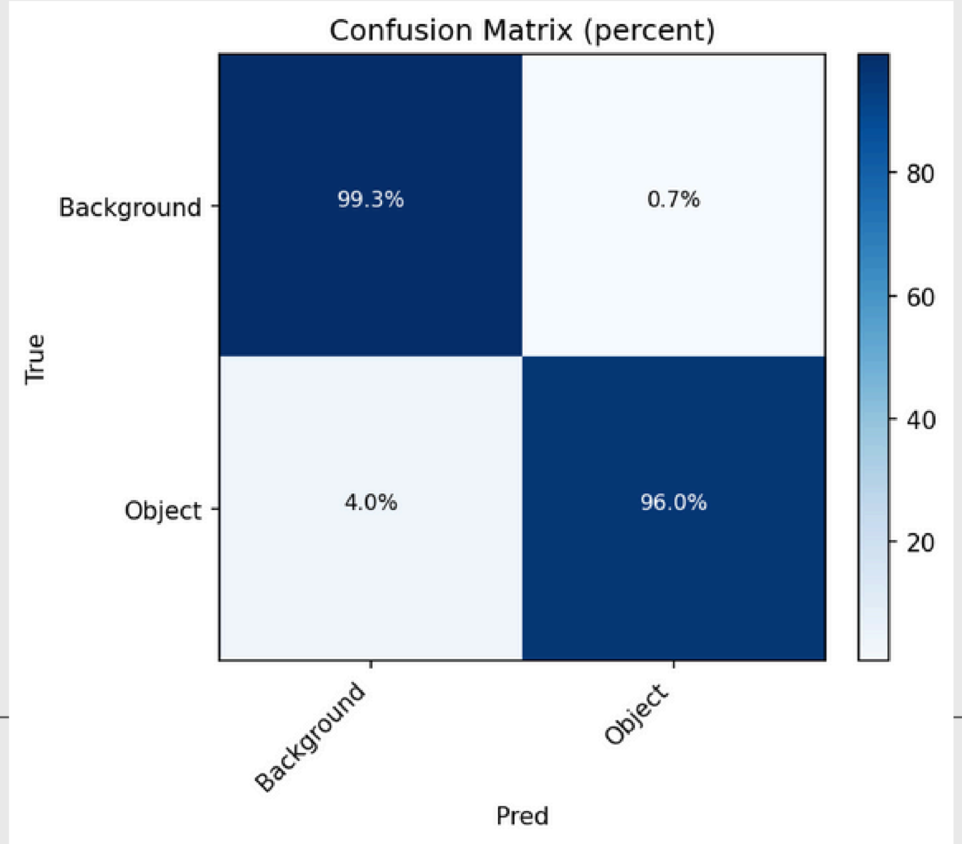

mIoU Curve (final)
Train mIoU
Val mIoU
Best Val: 0.9543
0.9543
mIoU
0.97
0.96
0.95
0.94
0.93
0
5
10
15
20
25
30
Epoch


Loss Curve (final)
Train Loss
Val Loss
Loss
0.10
0.08
0.06
0.04
0.02
0
5
10
15
20
25
30
Epoch

TEST RESULT ON FISH_4K



```
TEST SUMMARY
============================================================
Num images: 2737
Total pixels: 137331712
Pixel accuracy: 0.988976
mIoU: 0.953733

| Class      |    IoU |    Precision |    Recall |     F1 |    GT_count |    Pred_count |
|------------|--------|--------------|-----------|--------|-------------|---------------|
| Background | 0.9874 |        0.994 |    0.9933 | 0.9936 |   119180655 |     119103897 |
| Object     | 0.9201 |       0.9564 |    0.9604 | 0.9584 |    18151057 |      18227815 |

Averages:
Macro P/R/F1: 0.9752 / 0.9769 / 0.9760
Micro P/R/F1: 0.9890 / 0.9890 / 0.9890
Weighted P/R/F1: 0.9890 / 0.9890 / 0.9890
[INFO] Confusion matrices saved to: ./test_results_fish4k/confusion_matrix_counts.png and ./test_results_fish4k/confusion_matrix_percent.png
[INFO] Numeric summary saved to: ./test_results_fish4k/test_summary.txt

Saved concatenated images to: ./test_results_fish4k/concats
Done.
(hybrid_env) 2203131@jackfruit:~$ 
```



SOME OBSERVATIONS AND FURTHER IMPROVEMENTS

1. Blurry Object Boundaries

Observation:

Edges of fish, divers, and plants are not sharply segmented; boundaries look soft and incomplete.

Improvement:

- Add edge-aware loss (Boundary Loss / Sobel Loss).
- Use higher-capacity decoder or add ASPP / attention modules.
- Train with higher resolution (256/384 px).

2. Missing Small Objects (robots, tiny fish)

Observation:

Small classes are partially detected or completely ignored.

Improvement:

- Add multi-scale feature modules (FPN, ASPP).
- Increase decoder depth / use concatenation instead of addition.
- Apply small-object augmentation (Copy-Paste, random zoom-in).

3. Misclassification Between Similar Classes

Observation:

Fish identified as reefs or background; plants confused with background.

Improvement:

- Use class-balanced loss or weighted Dice.
- Add channel attention (CBAM/SE blocks) to separate textures.
- Introduce underwater style augmentations to improve robustness.

THANK YOU