**Assignment 4 – SeqTrack Inference Evaluation and Report**

Course: Image Processing

Team: [8]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**GitHub Repository**

<https://github.com/HossamAladin/Assignment_4.git>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**1-Performance Tables**

* **Summary**: Per-epoch evaluation metrics and speed for the modified SeqTrack on LaSOT (airplane, coin subsets). Metrics are aggregated by the evaluation pipeline .

A screenshot of a graph

AI-generated content may be incorrect.

* **Best AUC**: 24.91% (epoch 9)
* **Best IoU**: 24.91% (epoch 9)
* **Best Precision**: 27.24% (epoch 5)
* **Average FPS across epochs**: ~18.30

**--------------------------------------------------------------------------------------------------------------**

**Inference Rate**

**A black screen with white lines

AI-generated content may be incorrect.**

**-------------------------------------------------------------------------------------------------------------**

**Evaluation Results**

**A screenshot of a computer screen

AI-generated content may be incorrect.**

**Class-wise Results**

**A screen shot of a graph

AI-generated content may be incorrect.**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**2-Performance Graphs**

A graph with green and orange lines

AI-generated content may be incorrect.

* IoU, Precision, and AUC increase steadily from epochs 1–9, indicating consistent learning and convergence; a small dip at epoch 10 suggests the best overall checkpoint is epoch 9.

----------------------------------------------------------------------------------------------------------------

A graph with blue lines

AI-generated content may be incorrect.

* Success-overlap AUC climbs from ~6% in early epochs to ~25% by epoch 9, with a minor drop at epoch 10; this reflects improving tracking robustness across the evaluation split.

-----------------------------------------------------------------------------------------------------------------A graph with a line and a green line

AI-generated content may be incorrect.

* Inference speed is stable at ~18–19 FPS (≈52–56 ms/frame) across all epochs, showing training progress did not affect runtime.

-----------------------------------------------------------------------------------------------------------------A graph with blue and orange lines

AI-generated content may be incorrect.

* Airplane IoU improves more strongly and remains above coin from epoch 5 onward (≈30% vs ≈19% by epoch 10), indicating better localization on airplane sequences than coin.

------------------------------------------------------------------------------------------------------------------

A graph with blue and orange lines

AI-generated content may be incorrect.

* Both classes trend upward, with airplane AUC consistently higher after epoch 5 (≈30% airplane vs ≈18–21% coin by epochs 9–10), pointing to stronger overlap success on airplane.

--------------------------------------------------------------------------------------------------------------

A graph with blue and orange lines

AI-generated content may be incorrect.

* Airplane precision jumps sharply at epoch 5 and continues to rise toward ≈41 by epoch 10, while coin peaks around epoch 5 (≈20) and then declines, suggesting class-dependent sensitivity of center accuracy.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**3-Technical Implementation Details**

**Model Configuration**

* **Architecture**: Vision Transformer encoder with a lightweight decoder.
  + MODEL.ENCODER.TYPE: vit\_base\_patch16 (pretrained with mae)
  + MODEL.DECODER.DEC\_LAYERS: 6
  + MODEL.HIDDEN\_DIM: 256
  + MODEL.BINS: 4000
  + MODEL.FEATURE\_TYPE: x
* **Templates & Search**:
  + DATA.TEMPLATE.SIZE: 256, NUMBER: 2, FACTOR: 4.0
  + DATA.SEARCH.SIZE: 256, NUMBER: 1, FACTOR: 4.0
  + Jitter: template jitter disabled; search jitter enabled (CENTER\_JITTER=3.5, SCALE\_JITTER=0.5).
* **Training Hyperparameters** (used config: experiments/seqtrack/seqtrack\_b256.yaml):
  + TRAIN.EPOCH: 10
  + TRAIN.BATCH\_SIZE: 2
  + TRAIN.ENCODER\_MULTIPLIER: 0.1, OPTIMIZER: ADAMW, LR: 1e-4, WEIGHT\_DECAY: 1e-4
  + TRAIN.NUM\_WORKER: 2, PRINT\_INTERVAL: 50, SCHEDULER.TYPE: step, DECAY\_RATE: 0.1
  + TRAIN.SAVE\_EVERY\_EPOCH: true, SEED: 42
* **Inference Setup**:
  + TEST.SEARCH\_SIZE/TEMPLATE\_SIZE: 256, WINDOW: true, NUM\_TEMPLATES: 2

**Dataset Modifications**

* **Dataset**: LaSOT subset restricted to two classes.
  + DATA.TRAIN.DATASETS\_NAME: LASOT
  + DATA.TRAIN.SUBSET.CLASSES: [airplane, coin]
  + DATA.TRAIN.SUBSET.TRAIN\_SPLIT: 0.8
  + DATA.TRAIN.SUBSET.EXCLUDE: held-out sequences for evaluation: airplane-{1,9,13,15}, coin-{3,6,7,18}
* **Sampling**: DATA.TRAIN.SAMPLE\_PER\_EPOCH: 3500

**4-Dependency Fixes**

* requirements.txt added with: PyYAML, easydict, cython, opencv-python, pandas, tqdm, pycocotools, jpeg4py, tb-nightly, tikzplotlib, colorama, lmdb, scipy, visdom, timm, yacs, git+https://github.com/votchallenge/vot-toolkit-python.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**5-Evaluation Pipeline**

* {SeqTrack/evaluate\_checkpoints.Py} runs each epoch (1–10), calling tracking.test.run\_tracker to generate results under testing/results/....
* Metrics (IoU, AUC, Precision) are computed via lib.test.analysis.extract\_results over the evaluated sequences.
* Speed is computed from \*\_time.txt per-sequence files (frames, total time → FPS, ms/frame) and summarized per epoch.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**6-YAML Configuration Modifications**

* Active configuration used for this evaluation: seqtrack\_b256.yaml
  + DATA:
    - TRAIN.DATASETS\_NAME: LASOT
    - TRAIN.SAMPLE\_PER\_EPOCH: 3500
    - TRAIN.SUBSET.CLASSES: airplane, coin
    - TRAIN.SUBSET.TRAIN\_SPLIT: 0.8
    - TRAIN.SUBSET.EXCLUDE: airplane-1, airplane-9, airplane-13, airplane-15, coin-3, coin-6, coin-7, coin-18
    - SEARCH: SIZE=256, NUMBER=1, FACTOR=4.0, CENTER\_JITTER=3.5, SCALE\_JITTER=0.5
    - TEMPLATE: SIZE=256, NUMBER=2, FACTOR=4.0
  + MODEL:
    - ENCODER.TYPE: vit\_base\_patch16 (PRETRAIN\_TYPE: mae)
    - DECODER.DEC\_LAYERS: 6
    - HIDDEN\_DIM: 256; BINS: 4000; FEATURE\_TYPE: x
  + TRAIN:
    - EPOCH: 10; BATCH\_SIZE: 2; ENCODER\_MULTIPLIER: 0.1
    - OPTIMIZER: ADAMW; LR: 1e-4; WEIGHT\_DECAY: 1e-4
    - SCHEDULER.TYPE: step; DECAY\_RATE: 0.1
    - NUM\_WORKER: 2; PRINT\_INTERVAL: 50; SAVE\_EVERY\_EPOCH: true; SEED: 42
  + TEST:
    - SEARCH\_SIZE/TEMPLATE\_SIZE: 256; SEARCH\_FACTOR/TEMPLATE\_FACTOR: 4.0
    - WINDOW: true; NUM\_TEMPLATES: 2
    - UPDATE\_INTERVALS: VOT20/21/22 = 1
    - UPDATE\_THRESHOLD: VOT20/21/22 = 0.55
    - EPOCH: 500
* Evaluation registry present at SeqTrack/external/vot20/seqtrack/config.yaml registering VOT2020 INI parameter files.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**7-Evaluation Setup**

* **Model: SeqTrack-B256**
* **Dataset: LaSOT (8 sequences: 4 airplane + 4 coin)**
* **Sequence coverage: Full-length sequences evaluated end-to-end (no frame sampling or downsampling); timings aggregated from per-sequence \*\_time.txt logs.**
* **Epochs Evaluated: 1–10**
* **Total Runs: 80 (10 epochs × 8 sequences)**
* **Environment: Windows 11, Python 3.x, PyTorch**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**8-Key Findings**

* IoU and AUC rise steadily through epoch 9, with a slight dip at epoch 10. Precision peaks earlier (epoch 5) and then stabilizes.
* Inference speed is stable across epochs at ~18–19 FPS.
* The short run (10 epochs) on a small class-restricted subset behaves as expected: early checkpoints underperform; by epochs 5–10 the tracker stabilizes and metrics increase. On such a tiny split, one failed sequence can noticeably affect averages.

**9-Reflection Section**

1. Running full-length LaSOT sequences taught me how small changes in configuration (like class filtering and template count) propagate into real evaluation numbers. The steady FPS reassured me that accuracy gains didn’t come at runtime cost.
2. Evaluating 10 checkpoints highlighted how performance can peak before the last epoch; epoch 9 outperformed 10 in AUC/IoU. I learned to pick checkpoints by metrics, not by training order.
3. Using fixed seeds and per-epoch checkpoints made evaluations repeatable; re-runs matched within rounding. This gave me confidence that observed improvements were due to training progress, not randomness in the pipeline.
4. Class-wise graphs showed airplane outperforming coin after epoch 5, which suggests class characteristics matter for template updates. Next time I’d balance examples or tune class-specific thresholds.
5. The YAML made hyperparameters explicit and reproducible. Keeping TRAIN.EPOCH=10 with small BATCH\_SIZE=2 was enough to see clear trends without long training times.
6. Automating evaluation with evaluate\_checkpoints.py reduced manual errors and ensured consistent output. Having a single JSON file for tables made reporting fast and less error‑prone.
7. Early epochs behaved like underfitting—low IoU/Precision and unstable tracking. By epochs 5–10, the curves smoothed, confirming that even short fine‑tunes can meaningfully improve SeqTrack.
8. I appreciated that inference speed stayed ~18–19 FPS across epochs, which helps plan real‑time deployments. Accuracy improvements didn’t slow the tracker, which is practical for live use.
9. The evaluation pipeline clarified the meaning of each metric: IoU for overlap quality, Precision for center accuracy, and AUC as a robust success indicator over thresholds. Combining them avoided misleading conclusions.
10. The class-wise plots revealed different failure modes that averages hid—coin sometimes dropped while airplane improved. Next, I’ll audit per-sequence failures and tune update thresholds/number of templates to stabilize late-epoch performance.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**10-Performance Summary**

* **Best AUC**: 24.91% (epoch 9)
* **Best IoU**: 24.91% (epoch 9)
* **Best Precision**: 27.24% (epoch 5)
* **Average FPS**: ~18.30
* **General trend**: Clear improvement across epochs with stable runtime; epoch 9 is the strongest overall checkpoint.