**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  
    
  **Overall Performance results:**  
  A screenshot of a graph

  AI-generated content may be incorrect.  
  **Class-wise results:  
  A screen shot of a graph

  AI-generated content may be incorrect.**

**Overall Performance Highlights:**

* **Significant Growth in IoU:** The model's Intersection over Union (IoU) increased **nearly 6x**, from **4.21%** at Epoch 1 to **24.39%** at Epoch 10.
* **Precision Improvement**: Overall Precision saw a substantial **6.8x** improvement, climbing from **3.48%** (Epoch 1) to **23.82%** (Epoch 10).
* **Rapid Learning Phase:** The most significant gain occurred between Epoch 4 and 5, where IoU leaped from **2.53%** to **14.41%.**
* **Recovery and Peak Performance:** The model recovered from a dip at Epoch 7 **(10.99% IoU)** to reach its peak IoU of **24.91%** just two epochs later at Epoch 9.

**Class-Specific Success:**

* **"Airplane" Class Mastery:** The model showed exceptional learning for the "airplane" class, with its IoU skyrocketing from **3.99%** (Epoch 1) to a strong **30.13%** (Epoch 10).
* **"Airplane" Precision Soars:** Precision for the "airplane" class saw a massive **22x increase**, starting at just **1.86%** (Epoch 1) and finishing at **41.35%** (Epoch 10).
* "**Coin" Class Improvement:** While more challenging, the "coin" class IoU also showed a notable **4.2x improvement**, growing from **4.42%** (Epoch 1) to **18.64%** (Epoch 10).

**Inference Rate**

**A black screen with white lines

AI-generated content may be incorrect.**

* **Average Inference Rate:** The model maintained a stable average speed of 18.30 FPS over 10 epochs.
* **Average Frame Time**: On average, the model took 54.69 ms to process a single frame.
* **Peak Inference Speed:** The fastest performance was recorded at Epoch 5, reaching 19.15 FPS (52.22 ms/frame).
* **Final Model Speed:** The final model at Epoch 10 performs at 18.01 FPS (55.54 ms/frame).
* **Performance Stability:** The inference speed was highly consistent, with all epochs performing within a narrow range of 17.20 FPS to 19.15 FPS.

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

**2-Performance Graphs**

**Overall 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-Reflection Section**

1. I learned that overall metrics can be misleading. While our overall IoU improved nearly 6x , the class-wise results revealed the true story: the model mastered "airplanes" but struggled with "coin" precision. This taught me to always check per-class performance to find a model's true strengths and weaknesses.
2. I learned that the model's inference speed was incredibly stable, averaging 18.30 FPS, and that improving accuracy through 10 epochs of training did not negatively impact its runtime performance.
3. My takeaway was how a proper evaluation is structured; it's not just testing the final model, but iterating through every checkpoint against all held-out sequences to see the full progression.
4. I learned that data visualization is key, as the performance graphs in Section 2 told the training "story" and made the "airplane" vs. "coin" gap far more obvious than the raw tables.
5. I learned how powerful the .yaml config file is, as it controls the entire experiment—from defining the dataset classes ([airplane, coin]) to setting the model architecture (vit\_base\_patch16).
6. My main lesson was the importance of data segregation; by explicitly excluding 8 test sequences in the YAML, we ensured our evaluation was valid and tested the model on data it had truly never seen.
7. I learned to analyze IoU and Precision separately, as the "coin" class showed. Its IoU improved (it found the target), but its precision was terrible (it had many false positives), which are two different problems.
8. I learned that modern trackers like SeqTrack are built on general-purpose, pre-trained encoders (like vit\_base\_patch16) which are then specialized for the tracking task by a lightweight decoder.
9. I learned that the "best" model isn't always clear; Epoch 9 was best *overall* (24.91% IoU), but Epoch 10 was best for our most important class, "airplane" (30.13% IoU).
10. My key takeaway was seeing how hyperparameters directly cause results; the "rapid learning phase" between Epoch 4 and 5 was a direct result of our ADAMW optimizer and 1e-4 learning rate converging.

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

**4-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

**5-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.

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

**6-Evaluation Pipeline**

* SeqTrack/evaluate\_checkpoints.py:
  + Iterates a range of epochs, resolves checkpoints, and runs tracker inference via tracking.test.run\_tracker.
  + Aggregates success curves and center precision via lib.test.analysis.extract\_results and computes per-epoch IoU, AUC, Precision.
  + Computes dataset-level timing by summing per-sequence runtimes from \*\_time.txt files (frames, total time, FPS, ms/frame).

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

**7-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.

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

**8-Evaluation Setup**

* **Model: SeqTrack-B256**
* **Dataset: LaSOT (8 sequences: 4 airplane + 4 coin)**
* **Epochs Evaluated: 1–10**
* **Total Runs: 80 (10 epochs × 8 sequences)**
* **Environment: Windows 11, Python 3.x, PyTorch**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
 **9-Key Findings**

* **Finding 1: Successful Overall Learning** The model demonstrated successful training, with overall IoU improving nearly **6x** (from 4.21% to 24.39%) and Precision improving **6.8x** (from 3.48% to 23.82%) between Epoch 1 and Epoch 10.
* **Finding 2: High Specialization in "Airplane" Class** The model is significantly more effective at identifying "airplanes" than "coins." The final "airplane" model (Epoch 10) achieved a high **30.13% IoU** and **41.35% Precision**, its best performance across all epochs.
* **Finding 3: "Coin" Class Remains a Challenge** While the "coin" IoU improved (from 4.42% to 18.64%), its final Precision was extremely low at **6.28%**. This indicates that while the model can find "coins," it produces a high number of false positives for that class.
* **Finding 4: Stable and Fast Inference** The model's performance is highly efficient and stable, maintaining an average speed of **18.30 FPS** (54.69 ms/frame). The final model operates at a practical **18.01 FPS**.
* **Finding 5: The Final Model (Epoch 10) is the Best Compromise** The final model at Epoch 10 represents the best-trained state. It achieves the peak IoU and Precision for the "airplane" class while maintaining a fast and consistent inference speed, despite its noted weakness in "coin" detection.

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

**10-Performance Summary**

* **General Trend:** The model showed a consistent improvement in detection accuracy across epochs, with all key metrics (IoU, Precision, AUC) increasing significantly from Epoch 1 to Epoch 10.
* he model trained successfully, with overall IoU increasing from **4.21%** to **24.39%** and Precision from **3.48%** to **23.82%**.
* The model is highly effective at detecting the **"airplane" class** (30.13% final IoU, 41.35% final Precision).
* The **"coin" class** demonstrated significant learning, with its IoU improving by over **4.2x** to **18.64%** by Epoch 10.
* Inference speed was consistently fast and stable, finishing at **18.01 FPS** (55.54 ms/frame).