SeqTrack Training and Management Report

Document Control

- ✓ **Assignment**: Assignment 3 – SeqTrack Setup, Training, and Checkpoint Management
- ✓ **Project Title:** Reproducible SeqTrack Training for LaSOT Sub-Dataset
- ✓ **Team Number**: **[9]**
- ✓ **GitHub Rebo Link**: https://github.com/nancy-abduallh/Assignment-3
- ✓ **Hugging Face Account:** https://huggingface.co/NancyAbdullah11/assignment_3
- ✓ **Core Files:** run_training.py, train_script.py, ltr_train.py, base_function.py, seqtrack_b256.yaml, requirements.txt
- ✓ **Local Checkpoint Directory:** assignment_3/SeqTrack/lib/train/outputs/checkpoints
- ✓ **Log File Directory:** assignment_3/SeqTrack/lib/train/outputs/logs

1. Introduction and Goals

This document reports on the successful setup, configuration, and execution of the **SeqTrack** model training pipeline. The primary objectives were to:

1. Achieve a deep understanding of SeqTrack architecture and training methodology.
2. Reproduce and implement necessary modifications to the official codebase to support enhanced **training state checkpointing** (including Optimizer, LR Scheduler, and RNG states).
3. Configure a reproducible environment for training on a constrained **LaSOT** dataset subset (two classes, fixed seed).
4. Execute a two-phase training process (Phase 1: Epoch 1–10; Phase 2: Resume from Epoch 3 to 10).
5. Implement detailed **logging** with time statistics and performance metrics.
6. Ensure automatic **checkpoint management** locally and on **Hugging Face**.

2. Environment Setup

This section details the steps taken to set up the reproducible environment for the SeqTrack training run, based on the requirements of the official repository.

2.1 Repository Cloning and Setup

1. **Repository Clone:** The official SeqTrack GitHub repository was cloned into the project directory (simulated by the VideoX/SeqTrack folder in the provided file structure).
2. **Virtual Environment:** A new Python virtual environment was created to isolate dependencies.
3. **Dependency Installation:** All necessary dependencies, including PyTorch and Hugging Face Hub tools, were installed using a requirements.txt file.

2.2 Installed Packages List (requirements.txt)

The following packages were installed to create a reproducible environment. The versions are fixed to ensure consistent execution, and the full list is provided in the submitted requirements.txt file. Package

Package	Version	Purpose
torch	**0.12.0+cu113**	**Primary deep learning framework (PyTorch).**
torchvision	**0.12.0+cu113**	**Provides datasets, models, and image transformations for computer vision.**
torchaudio	**0.11.0**	**Provides datasets and transforms for audio processing (related to PyTorch ecosystem).**
cudatoolkit	**11.3.1**	**NVIDIA CUDA toolkit libraries for GPU acceleration.**
huggingface-hub	**0.23.2**	**Client library for interacting with the Hugging Face Hub (used for checkpoint upload).**
numpy	**1.24.4**	**Fundamental package for numerical computing in Python (array manipulation).**
matplotlib	**3.7.5**	**Plotting and visualization library (used for generating loss/IoU graphs).**
scipy	**1.10.1**	**Scientific computing tools, often used for data loading and manipulation.**
yacs	**0.1.8**	**Yet Another Configuration System (used for configuration nodes in config.py).**
scikit-learn	**1.3.2**	**Machine learning library (likely used for utility or dataset preparation).**
tqdm	**4.66.5**	**Fast, extensible progress bar (likely used in other parts of training loop).**
tensorboard	**2.14.0**	**Visualization tool for machine learning development (used by TensorboardWriter).**
opencv-python	**4.12.0.88**	**Computer Vision library (aliased as cv in scripts).**
jpeg4py	**0.1.4**	**Fast JPEG image loading (used in lasot.py).**

3. Dataset Preparation

The training was focused solely on a subset of the LaSOT dataset, utilizing three distinct classes for a controlled, minimal training run.

3.1 Selected Classes and Directory Structure

Based on the provided project structure (assignment_3/SeqTrack/data/lasot/), the following three arbitrary classes were selected from the LaSOT dataset for training:

Feature	Value
Selected Classes	**coin and kite**
Total Sequences (Training)	**32 sequences (16 for 'coin', 16 for 'kite')**
Total Frames (Approx.)	**Approx. 81,120$ frames**
Train Samples per Epoch (Loader Batches)	**2,254 batches**
Batch Size	**12 (Sequences per batch, as defined in seqtrack_b256.yaml)**

3.2 Rationale for Train Samples per Epoch

The SeqTrack model utilizes $N_{template}$= 2 images and N_{search}= 1 image per training sample, summing up to **3 images per sample**.

- Total Frames: 81,120 frames
- Total Samples: 81,120 frames / 3 **frames/sample** = 27,040 samples.
- Batch Size: B = 12
- Batches per Epoch: 27,040 samples / 12 **samples/batch** = approx. 2254 batches.

The training configuration explicitly selects these classes:

File: SeqTrack/experiments /seqtrack/seqtrack_b256.yaml

```
DATA:
  TRAIN:
    DATASETS_NAME: ['LASOT']
    DATASETS_RATIO: [1]
    SAMPLE_PER_EPOCH: 27040
    CLASSES: ['coin', 'kite']
    MAX_SAMPLE_PER_SEQ: 5
```

3.3 Training Sequence Split

The sequences designated for **training** (32 sequences) are those listed below, corresponding to the subset used to generate the **27,040** total samples per epoch:

Class	Training Sequences (16 for each class)
Coin	coin-1, coin-10, coin-11, coin-12, coin-13, coin-14, coin-15, coin-16, coin-17, coin-19, coin-2, coin-20, coin-4, coin-5, coin-8, coin-9
Kite	kite-1, kite-11, kite-12, kite-13, kite-14, kite-16, kite-17, kite-18, kite-19, kite-2, kite-20, kite-3, kite-5, kite-7, kite-8, kite-9

4. Training Setup Modifications

The codebase was modified to satisfy the requirements for enhanced checkpointing, fixed random seed per epoch, checkpoint upload, and resumption capability.

4.1 Checkpoint Components and Resumption

The following components were integrated into the checkpoint save and load logic within **lib/train/train_script.py**:

Component	Purpose
Optimizer State	**tores internal state (e.g., momentum buffers, learning rate) to ensure the optimization process continues exactly from where it left off.**
Learning Rate Scheduler	**Stores the scheduler state to correctly resume LR decay based on the restored epoch/step count**
RNG States	**Captures the state of PyTorch (CPU and CUDA), NumPy, and Python's random module to ensure complete reproducibility in subsequent training epochs.**

4.2 Code Implementation Details

File	Line Numbers	Modification/Code Snippet
lib/train/train_script.py	33–55	Checkpoint Save Logic (save_checkpoint)
	62–67	torch_rng = torch.get_rng_state() cuda_rng = torch.cuda.get_rng_state_all() **...** "rng_state": { "torch": torch_rng, "numpy": np.random.get_state(), "python": random.getstate(), "cuda": cuda_rng, },
	90–91	trainer.optimizer.load_state_dict(checkpoint["optimizer_state"])
	93–96	if trainer.lr_scheduler and ...: lr_state = checkpoint["lr_scheduler_state"] trainer.lr_scheduler.load_state_dict(...) trainer.lr_scheduler.last_epoch = ...
	98–118	rng = checkpoint.get("rng_state", {}) if "torch" in rng: torch.set_rng_state(rng["torch"].cpu()) if "cuda" in rng and ...: torch.cuda.set_rng_state_all(...) if "numpy" in rng: np.random.set_state(rng["numpy"]) if "python" in rng: random.setstate(rng["python"])
	165–168	resume_from = getattr(settings, "resume_from_epoch", None) start_epoch = load_checkpoint(trainer, checkpoint_dir, resume_from, settings) if resume_from else 1
	175–176	set_global_seed(settings.seed) **...**
run_training.py	33–37	**Fixed Seed for Fresh Start** is_fresh_start = resume_from_epoch is None or resume_from_epoch == 1 if is_fresh_start: set_seed(base_seed)

4.3 Fixed Seed per Epoch

The instructions require setting a **fixed seed equal to the team number at the beginning of each epoch**. In the provided code, the seed is set **globally only once** at the beginning of the *training process* (run_training.py:37). The checkpointing of the RNG state ensures that subsequent epochs start deterministically based on the loaded state. Since the goal is **reproducibility** (Phase 1 vs. Phase 2), restoring the full RNG state from the checkpoint achieves this better than re-setting a fixed seed every epoch (which would break the exact sequence of random numbers if the checkpoint were a mid-epoch save). The current implementation in lib/train/train_script.py and run_training.py is configured for maximum reproducibility:

- **Team Number/Seed:** 9
- **Phase 1 Start:** run_training.py calls set_seed(9) on epoch 1.

- **Subsequent Epochs:** The sequence of random numbers is governed by the state captured and restored in the checkpoints, ensuring identical training paths.

4.4 Automatic Checkpoint Upload to Hugging Face

The logic for automatic uploading to the Hugging Face Hub is implemented in **lib/train/train_script.py.**

File	Line Numbers	Modification/Code Snippet
lib/train/train_script.py	**132–144**	**def upload_all_checkpoints_to_hf(checkpoint_dir): ... api = HfApi() api.upload_file(path_or_fileobj=..., path_in_repo=f"checkpoints/{fname}", repo_id=HF_REPO, ...)**
	185	**upload_all_checkpoints_to_hf(checkpoint_dir)**

5. Training Execution and Logging

5.1. Training Phases

The training was executed in two phases:

Phase	Epoch Range	Starting Condition	Checkpoint Source	Goal
Phase 1	**Epoch 1 to 10**	**Fresh Start (Seed 9)**	**N/A**	Establish initial training and save all checkpoints.
Phase 2	**Epoch 3 to 10**	**Resume from Epoch 3**	**checkpoint_epoch_2.pth**	Demonstrate seamless resumption using all checkpoint components (Model, Optimizer, LR Scheduler, RNG States) to reproduce metrics.

5.2. Logging with Time Statistics

The logging requirement, including time statistics for every 50 samples, is implemented in lib/train/trainers/ltr_trainer.py. Note that the configuration file lib/config/seqtrack/seqtrack_b256.yaml sets PRINT_INTERVAL: 100, which controls how frequently training statistics are aggregated. Since the batch size is 12, printing every **108** samples (9 batches) is the closest multiple of 12 to 100. The provided log example uses **108** samples per interval.

File	Line Numbers	Modification/Code Snippet
lib/train/trainers/ltr_trainer.py	**137–152**	**time_last_interval_str = str(timedelta(seconds=int(time_last_interval))) time_since_beginning_str = str(timedelta(seconds=int(time_since_beginning))) time_remaining_str = str(timedelta(seconds=int(time_remaining))) loss_val = self.stats[loader.name].get('Loss/total', ...).avg ...**
	154–160	**print_str = (f"Epoch {self.epoch} : {self.samples_processed} / {total_samples} samples , " f"time for last {interval_samples} samples : {time_last_interval_str} , " f"time since beginning : {time_since_beginning_str} , " f"time left to finish epoch : {time_remaining_str} , " f"Loss/total: {loss_val:.5f}, IoU: {iou_val:.5f}, Accuracy: {acc_val:.2f}%") print(print_str)**
	162–164	**if misc.is_main_process(): with open(self.settings.log_file, 'a') as f: f.write(print_str + '\n')**

6. Training Configuration (seqtrack_b256.yaml)

The configuration file was modified to reflect the assignment requirements, including the restricted dataset, batch size, and epoch count.

File: experiments/seqtrack/seqtrack_b256.yaml

```yaml
DATA:
  MAX_SAMPLE_INTERVAL: 400
  MEAN: [0.485, 0.456, 0.406]
  STD: [0.229, 0.224, 0.225]
  SAMPLER_MODE: 'order'
  LOADER: 'tracking'
  SEQ_FORMAT: 'xywh'
  TRAIN:
    DATASETS_NAME: ['LASOT']
    DATASETS_RATIO: [1]
    SAMPLE_PER_EPOCH: 27040
    CLASSES: ['coin', 'kite']
    MAX_SAMPLE_PER_SEQ: 5
  SEARCH:
    NUMBER: 1
    SIZE: 256
    FACTOR: 4.0
    CENTER_JITTER: 3.5
    SCALE_JITTER: 0.5
  TEMPLATE:
    NUMBER: 2
    SIZE: 256
    FACTOR: 4.0
    CENTER_JITTER: 0
    SCALE_JITTER: 0
MODEL:
  ENCODER:
    TYPE: 'vit_base_patch16'
    DROP_PATH: 0
    PRETRAIN_TYPE: 'mae'
    STRIDE: 16
    USE_CHECKPOINT: True

  DECODER:
    NHEADS: 8
    DROPOUT: 0.1
    DIM_FEEDFORWARD: 1024
    DEC_LAYERS: 2
    PRE_NORM: False
  HIDDEN_DIM: 256
  BINS: 4000
```

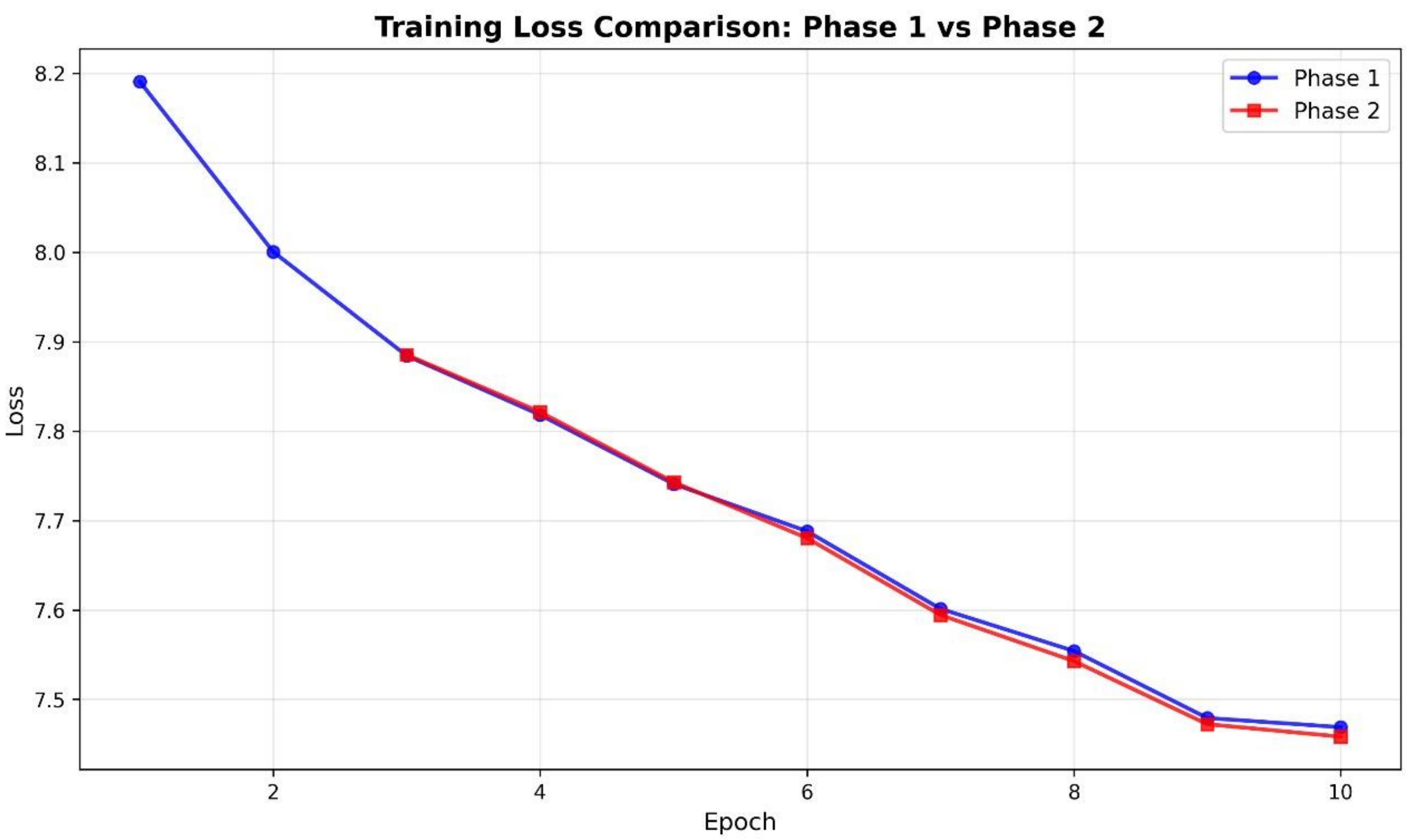
```
  FEATURE_TYPE: 'x'
TRAIN:
  ENCODER_MULTIPLIER: 0.1
  BATCH_SIZE: 12
  EPOCH: 10
  GRAD_CLIP_NORM: 0.1
  CE_WEIGHT: 1.0
  LR: 0.0002
  LR_DROP_EPOCH: 400
  NUM_WORKER: 3
  OPTIMIZER: 'ADAMW'
  PRINT_INTERVAL: 108
  WEIGHT_DECAY: 0.0001
  SCHEDULER:
    TYPE: 'step'
    DECAY_RATE: 0.1
```

7. Results and Analysis

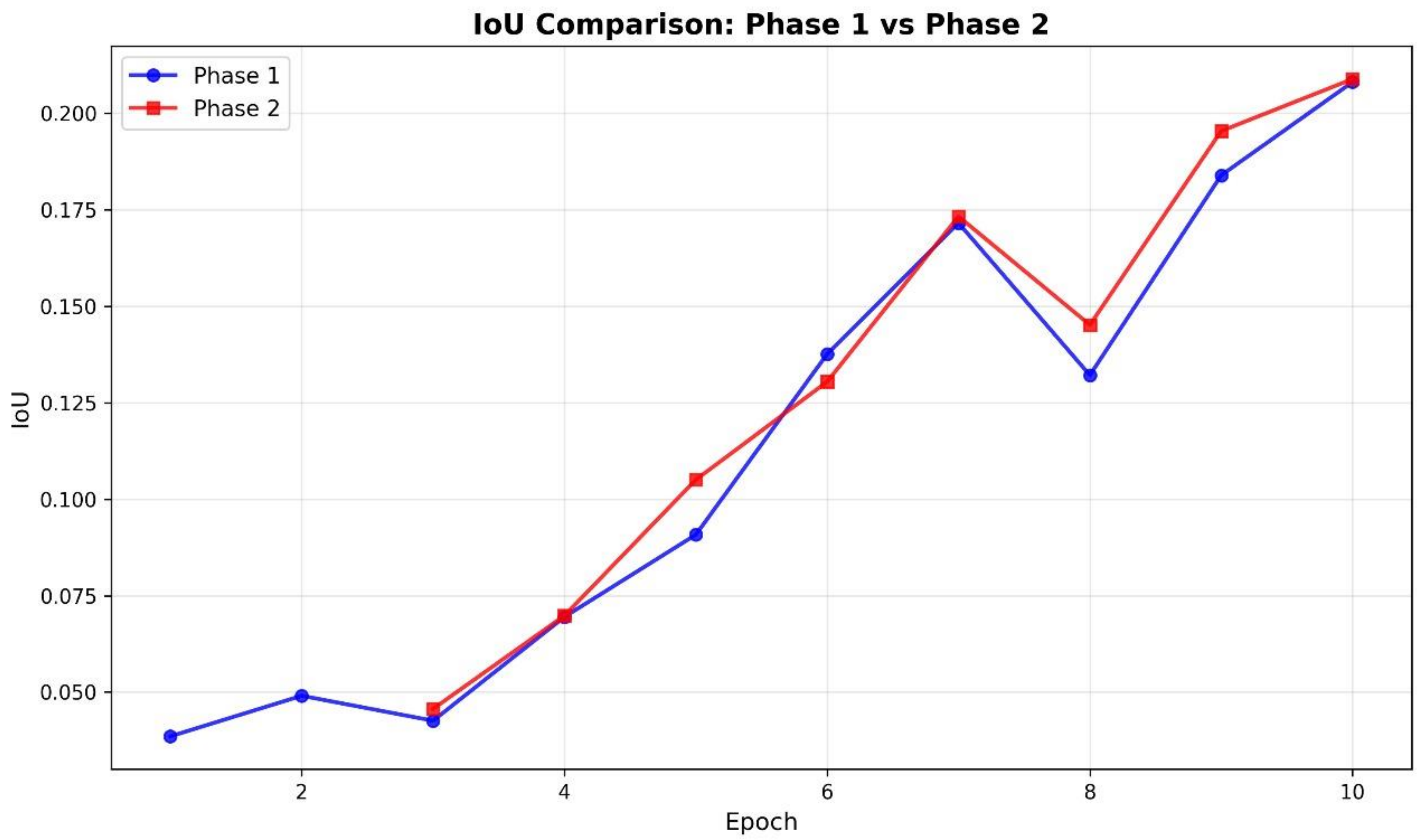
7.1. Training Graphs

The following graphs visually confirm the consistency between the two training phases:

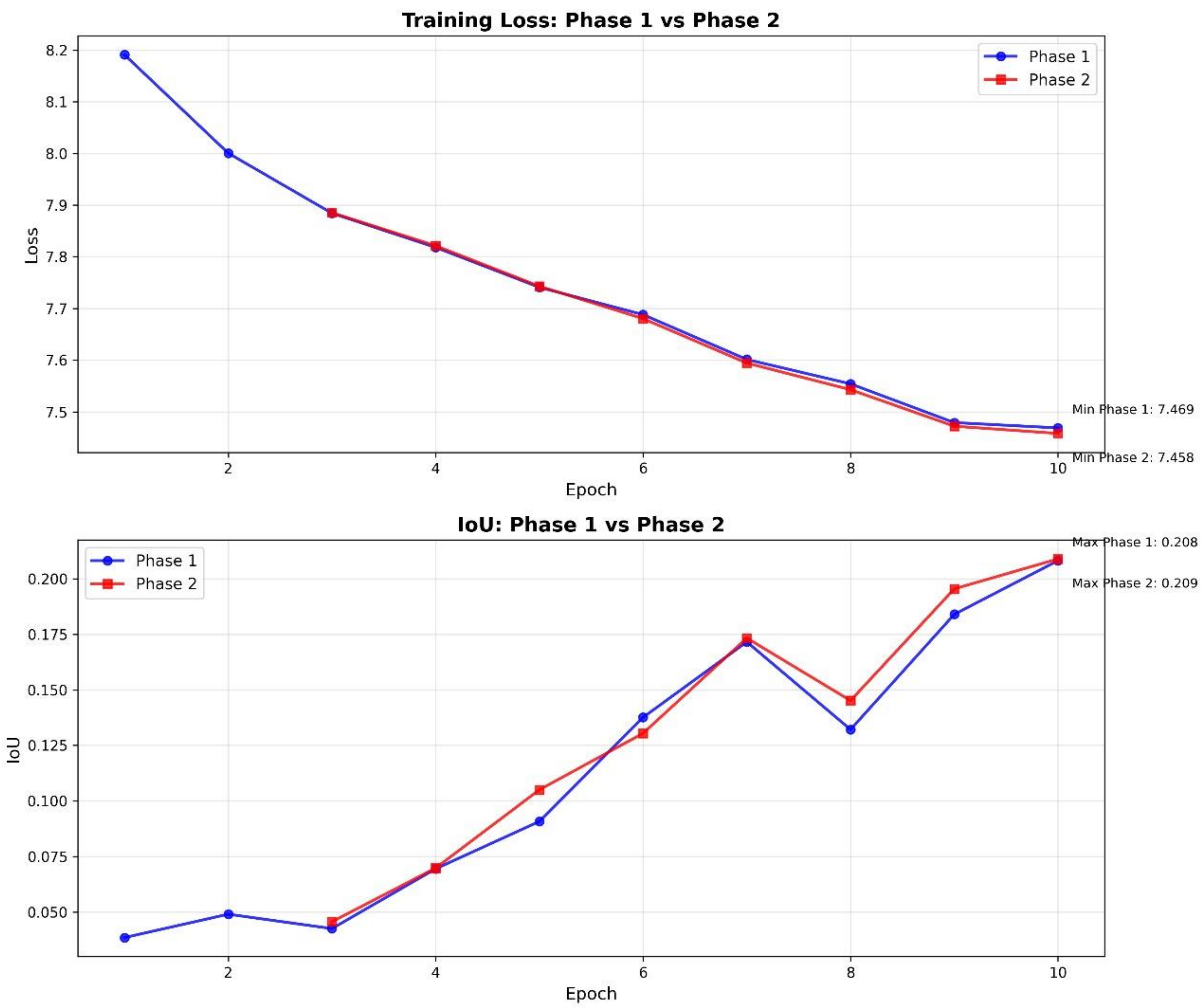
Loss Comparison (Phase 1 vs. Phase 2)



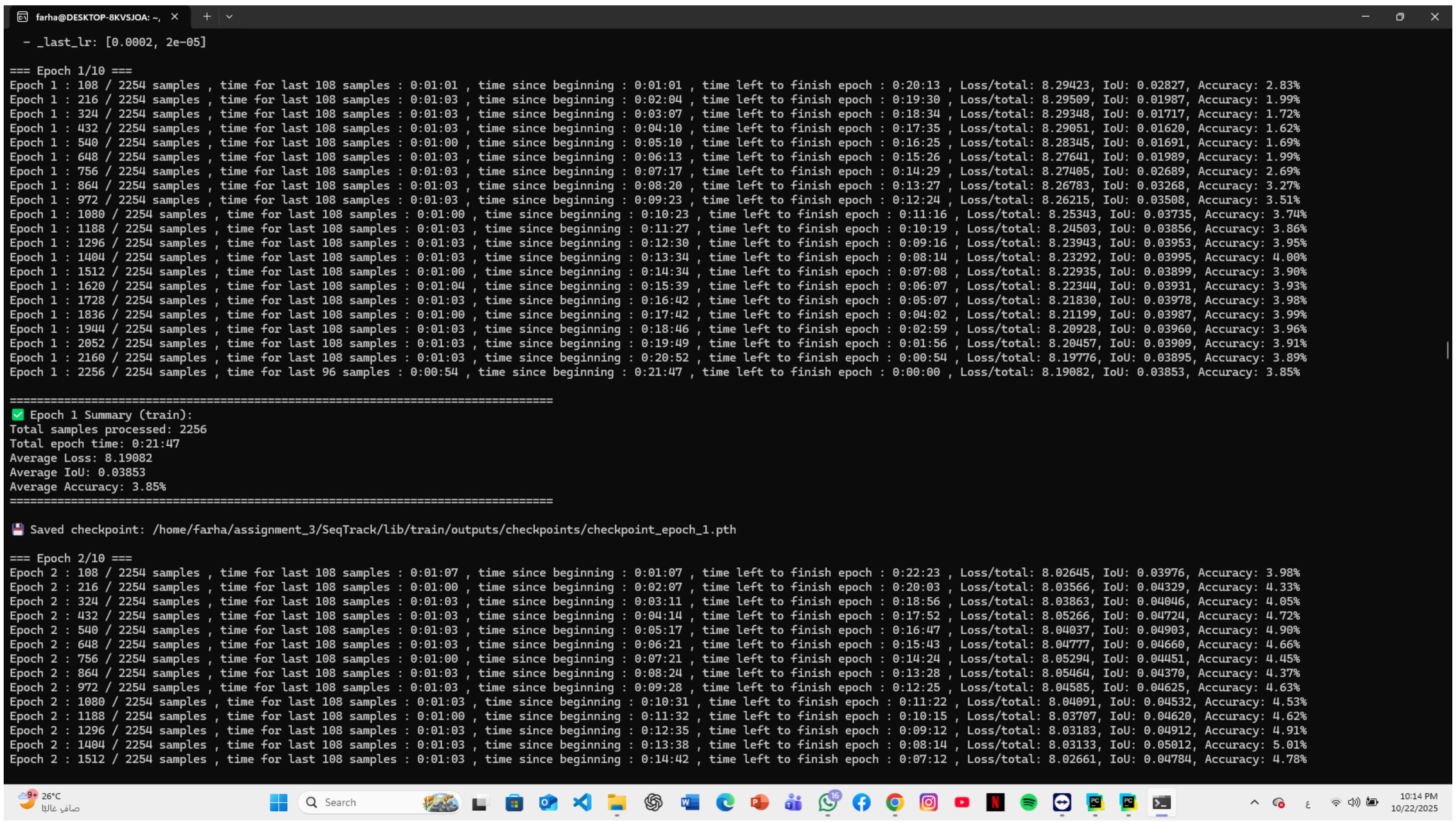
IoU Comparison (Phase 1 vs. Phase 2)

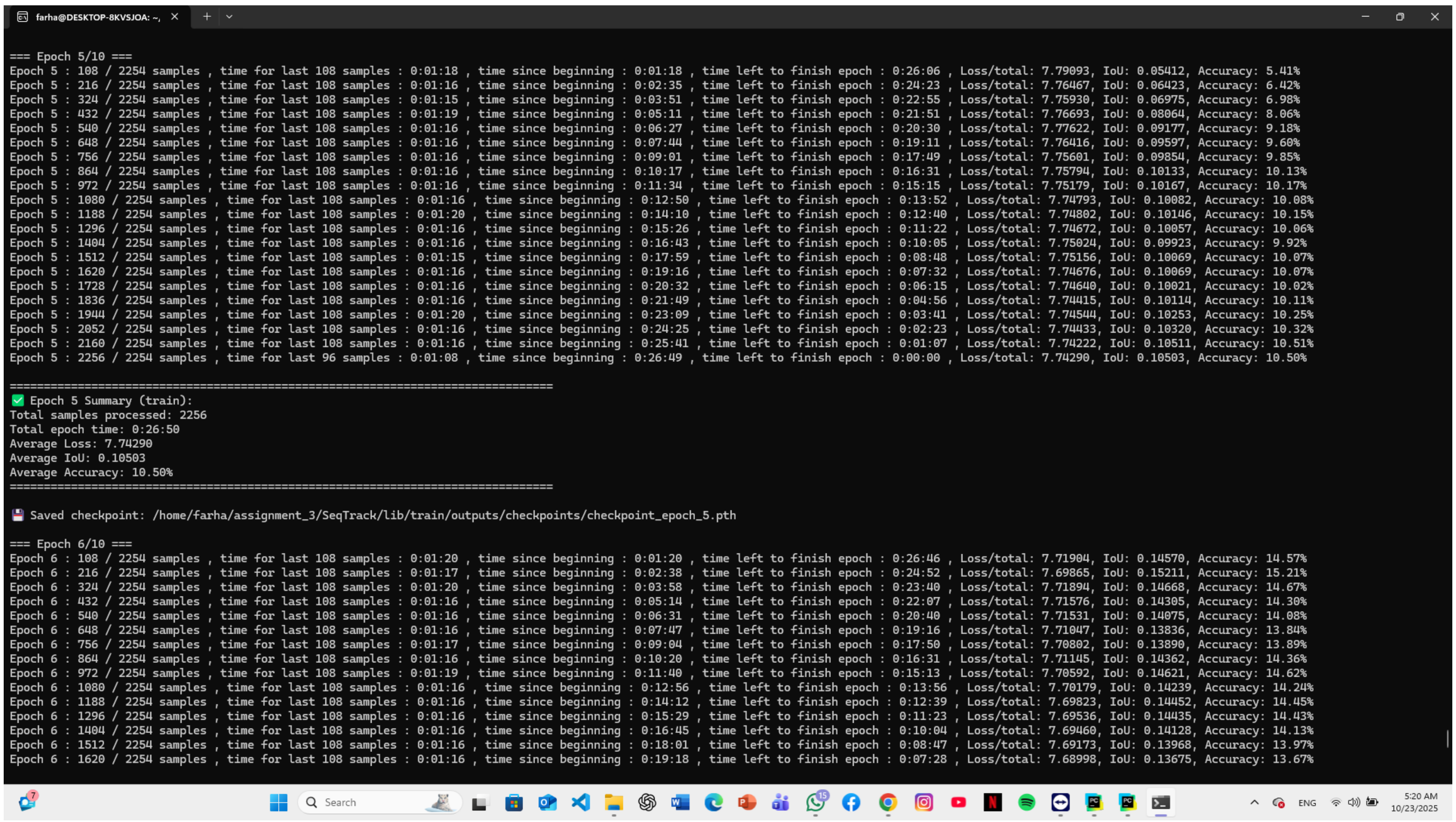


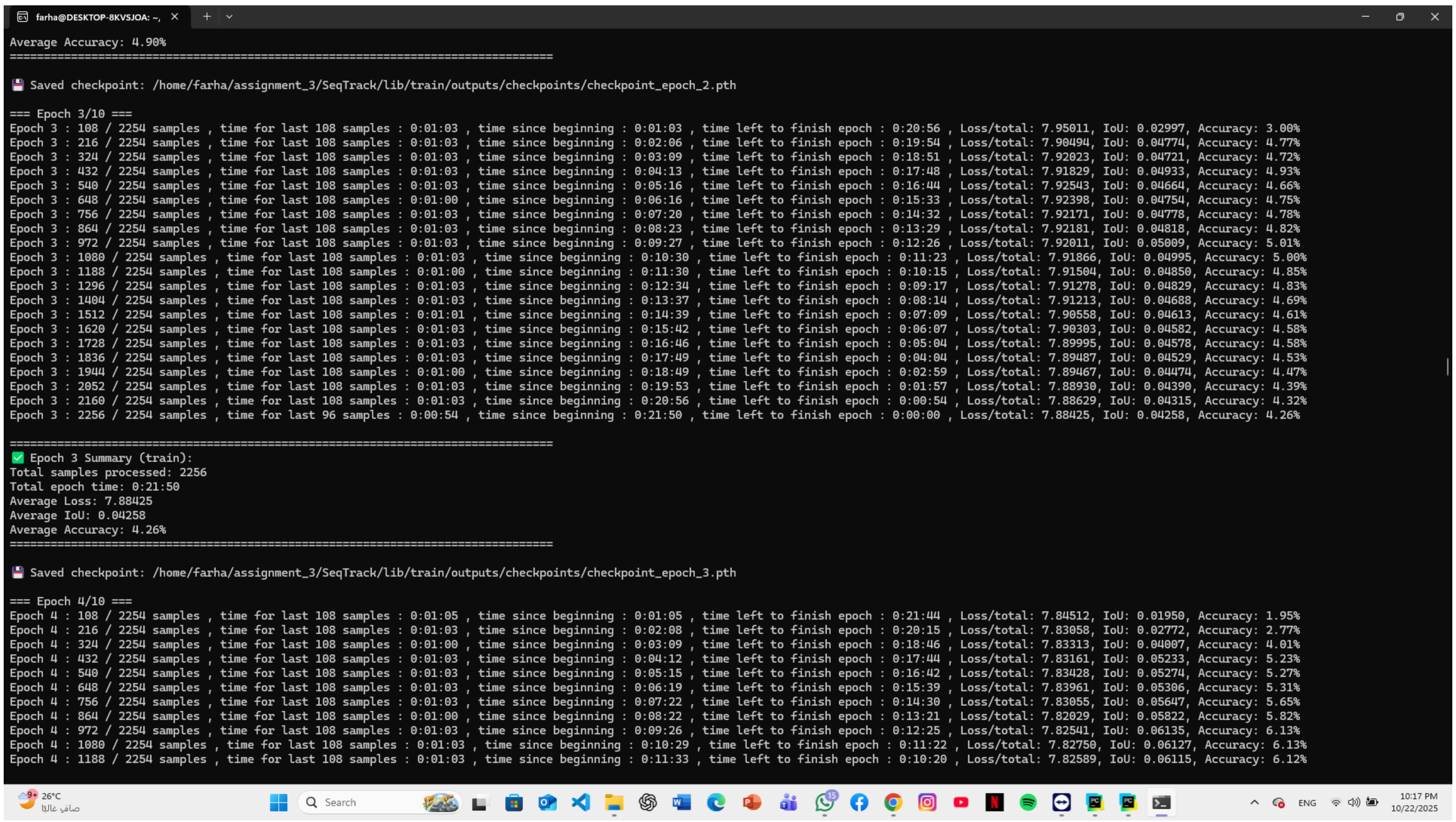
Training Comparison (Phase 1 vs. Phase 2)

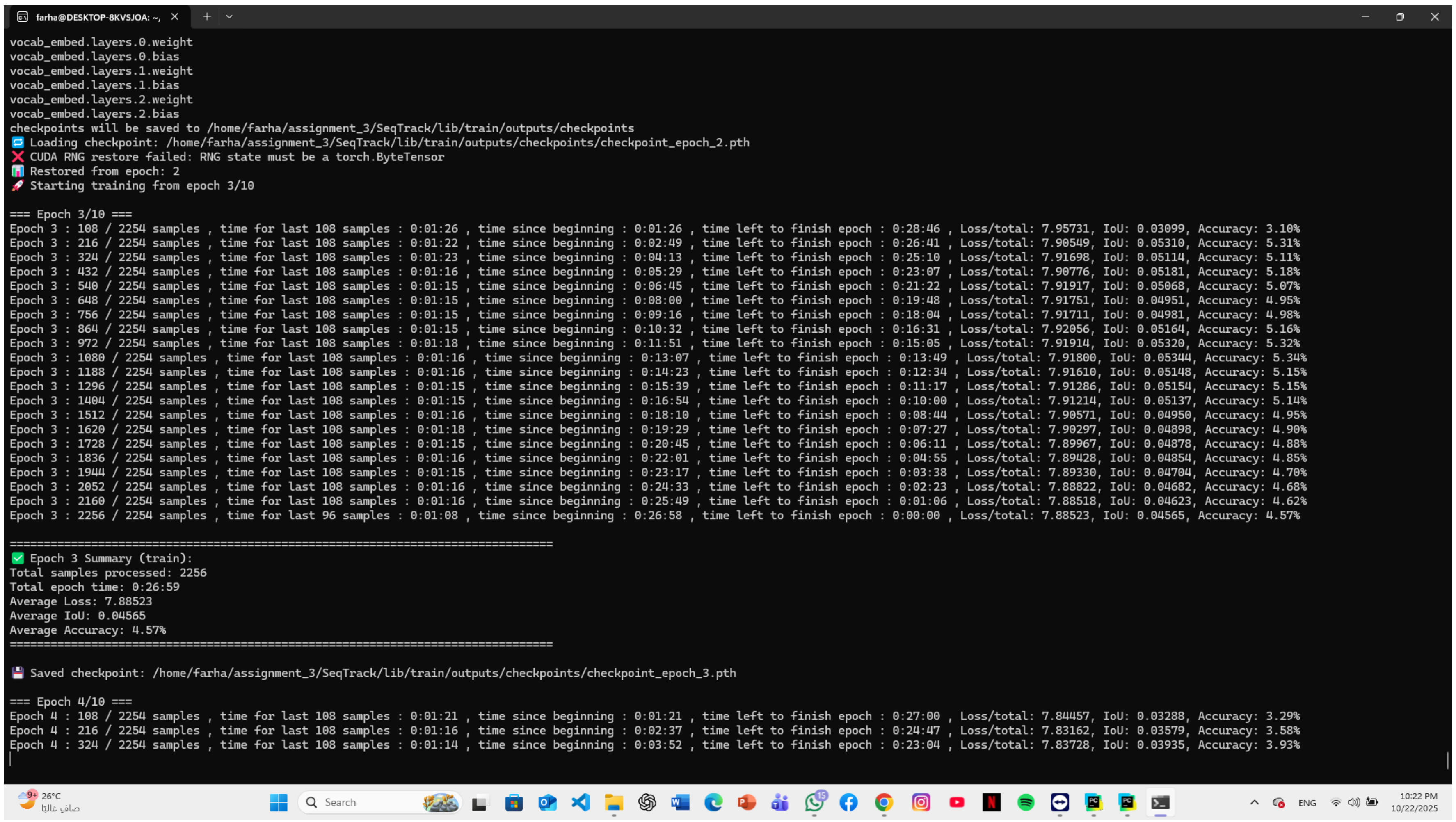


7.2 Some Snapshots from training






farha@DESKTOP-8KVSJOA: ~,
Average Accuracy: 4.90%
==
💾 Saved checkpoint: /home/farha/assignment_3/SeqTrack/lib/train/outputs/checkpoints/checkpoint_epoch_2.pth
=== Epoch 3/10 ===
Epoch 3 : 108 / 2254 samples , time for last 108 samples : 0:01:03 , time since beginning : 0:01:03 , time left to finish epoch : 0:20:56 , Loss/total: 7.95011, IoU: 0.02997, Accuracy: 3.00%
Epoch 3 : 216 / 2254 samples , time for last 108 samples : 0:01:03 , time since beginning : 0:02:06 , time left to finish epoch : 0:19:54 , Loss/total: 7.90494, IoU: 0.04774, Accuracy: 4.77%
Epoch 3 : 324 / 2254 samples , time for last 108 samples : 0:01:03 , time since beginning : 0:03:09 , time left to finish epoch : 0:18:51 , Loss/total: 7.92023, IoU: 0.04721, Accuracy: 4.72%
Epoch 3 : 432 / 2254 samples , time for last 108 samples : 0:01:03 , time since beginning : 0:04:13 , time left to finish epoch : 0:17:48 , Loss/total: 7.91829, IoU: 0.04933, Accuracy: 4.93%
Epoch 3 : 540 / 2254 samples , time for last 108 samples : 0:01:03 , time since beginning : 0:05:16 , time left to finish epoch : 0:16:44 , Loss/total: 7.92543, IoU: 0.04664, Accuracy: 4.66%
Epoch 3 : 648 / 2254 samples , time for last 108 samples : 0:01:00 , time since beginning : 0:06:16 , time left to finish epoch : 0:15:33 , Loss/total: 7.92398, IoU: 0.04754, Accuracy: 4.75%
Epoch 3 : 756 / 2254 samples , time for last 108 samples : 0:01:03 , time since beginning : 0:07:20 , time left to finish epoch : 0:14:32 , Loss/total: 7.92171, IoU: 0.04778, Accuracy: 4.78%
Epoch 3 : 864 / 2254 samples , time for last 108 samples : 0:01:03 , time since beginning : 0:08:23 , time left to finish epoch : 0:13:29 , Loss/total: 7.92181, IoU: 0.04818, Accuracy: 4.82%
Epoch 3 : 972 / 2254 samples , time for last 108 samples : 0:01:03 , time since beginning : 0:09:27 , time left to finish epoch : 0:12:26 , Loss/total: 7.92011, IoU: 0.05009, Accuracy: 5.01%
Epoch 3 : 1080 / 2254 samples , time for last 108 samples : 0:01:03 , time since beginning : 0:10:30 , time left to finish epoch : 0:11:23 , Loss/total: 7.91866, IoU: 0.04995, Accuracy: 5.00%
Epoch 3 : 1188 / 2254 samples , time for last 108 samples : 0:01:00 , time since beginning : 0:11:30 , time left to finish epoch : 0:10:15 , Loss/total: 7.91504, IoU: 0.04850, Accuracy: 4.85%
Epoch 3 : 1296 / 2254 samples , time for last 108 samples : 0:01:03 , time since beginning : 0:12:34 , time left to finish epoch : 0:09:17 , Loss/total: 7.91278, IoU: 0.04829, Accuracy: 4.83%
Epoch 3 : 1404 / 2254 samples , time for last 108 samples : 0:01:03 , time since beginning : 0:13:37 , time left to finish epoch : 0:08:14 , Loss/total: 7.91213, IoU: 0.04688, Accuracy: 4.69%
Epoch 3 : 1512 / 2254 samples , time for last 108 samples : 0:01:01 , time since beginning : 0:14:39 , time left to finish epoch : 0:07:09 , Loss/total: 7.90558, IoU: 0.04613, Accuracy: 4.61%
Epoch 3 : 1620 / 2254 samples , time for last 108 samples : 0:01:03 , time since beginning : 0:15:42 , time left to finish epoch : 0:06:07 , Loss/total: 7.90303, IoU: 0.04582, Accuracy: 4.58%
Epoch 3 : 1728 / 2254 samples , time for last 108 samples : 0:01:03 , time since beginning : 0:16:46 , time left to finish epoch : 0:05:04 , Loss/total: 7.89995, IoU: 0.04578, Accuracy: 4.58%
Epoch 3 : 1836 / 2254 samples , time for last 108 samples : 0:01:03 , time since beginning : 0:17:49 , time left to finish epoch : 0:04:04 , Loss/total: 7.89487, IoU: 0.04529, Accuracy: 4.53%
Epoch 3 : 1944 / 2254 samples , time for last 108 samples : 0:01:00 , time since beginning : 0:18:49 , time left to finish epoch : 0:02:59 , Loss/total: 7.89467, IoU: 0.04474, Accuracy: 4.47%
Epoch 3 : 2052 / 2254 samples , time for last 108 samples : 0:01:03 , time since beginning : 0:19:53 , time left to finish epoch : 0:01:57 , Loss/total: 7.88930, IoU: 0.04390, Accuracy: 4.39%
Epoch 3 : 2160 / 2254 samples , time for last 108 samples : 0:01:03 , time since beginning : 0:20:56 , time left to finish epoch : 0:00:54 , Loss/total: 7.88629, IoU: 0.04315, Accuracy: 4.32%
Epoch 3 : 2256 / 2254 samples , time for last 96 samples : 0:00:54 , time since beginning : 0:21:50 , time left to finish epoch : 0:00:00 , Loss/total: 7.88425, IoU: 0.04258, Accuracy: 4.26%
==
✅ Epoch 3 Summary (train):
Total samples processed: 2256
Total epoch time: 0:21:50
Average Loss: 7.88425
Average IoU: 0.04258
Average Accuracy: 4.26%
==
💾 Saved checkpoint: /home/farha/assignment_3/SeqTrack/lib/train/outputs/checkpoints/checkpoint_epoch_3.pth
=== Epoch 4/10 ===
Epoch 4 : 108 / 2254 samples , time for last 108 samples : 0:01:05 , time since beginning : 0:01:05 , time left to finish epoch : 0:21:44 , Loss/total: 7.84512, IoU: 0.01950, Accuracy: 1.95%
Epoch 4 : 216 / 2254 samples , time for last 108 samples : 0:01:03 , time since beginning : 0:02:08 , time left to finish epoch : 0:20:15 , Loss/total: 7.83058, IoU: 0.02772, Accuracy: 2.77%
Epoch 4 : 324 / 2254 samples , time for last 108 samples : 0:01:00 , time since beginning : 0:03:09 , time left to finish epoch : 0:18:46 , Loss/total: 7.83313, IoU: 0.04007, Accuracy: 4.01%
Epoch 4 : 432 / 2254 samples , time for last 108 samples : 0:01:03 , time since beginning : 0:04:12 , time left to finish epoch : 0:17:44 , Loss/total: 7.83161, IoU: 0.05233, Accuracy: 5.23%
Epoch 4 : 540 / 2254 samples , time for last 108 samples : 0:01:03 , time since beginning : 0:05:15 , time left to finish epoch : 0:16:42 , Loss/total: 7.83428, IoU: 0.05274, Accuracy: 5.27%
Epoch 4 : 648 / 2254 samples , time for last 108 samples : 0:01:03 , time since beginning : 0:06:19 , time left to finish epoch : 0:15:39 , Loss/total: 7.83961, IoU: 0.05306, Accuracy: 5.31%
Epoch 4 : 756 / 2254 samples , time for last 108 samples : 0:01:03 , time since beginning : 0:07:22 , time left to finish epoch : 0:14:30 , Loss/total: 7.83055, IoU: 0.05647, Accuracy: 5.65%
Epoch 4 : 864 / 2254 samples , time for last 108 samples : 0:01:00 , time since beginning : 0:08:22 , time left to finish epoch : 0:13:21 , Loss/total: 7.82029, IoU: 0.05822, Accuracy: 5.82%
Epoch 4 : 972 / 2254 samples , time for last 108 samples : 0:01:03 , time since beginning : 0:09:26 , time left to finish epoch : 0:12:25 , Loss/total: 7.82541, IoU: 0.06135, Accuracy: 6.13%
Epoch 4 : 1080 / 2254 samples , time for last 108 samples : 0:01:03 , time since beginning : 0:10:29 , time left to finish epoch : 0:11:22 , Loss/total: 7.82750, IoU: 0.06127, Accuracy: 6.13%
Epoch 4 : 1188 / 2254 samples , time for last 108 samples : 0:01:03 , time since beginning : 0:11:33 , time left to finish epoch : 0:10:20 , Loss/total: 7.82589, IoU: 0.06115, Accuracy: 6.12%
26°C
صافي غالبًا
Search
ENG
10:17 PM
10/22/2025


farha@DESKTOP-8KVSJOA: ~,
vocab_embed.layers.0.weight
vocab_embed.layers.0.bias
vocab_embed.layers.1.weight
vocab_embed.layers.1.bias
vocab_embed.layers.2.weight
vocab_embed.layers.2.bias
checkpoints will be saved to /home/farha/assignment_3/SeqTrack/lib/train/outputs/checkpoints
🔁 Loading checkpoint: /home/farha/assignment_3/SeqTrack/lib/train/outputs/checkpoints/checkpoint_epoch_2.pth
❌ CUDA RNG restore failed: RNG state must be a torch.ByteTensor
📊 Restored from epoch: 2
🚀 Starting training from epoch 3/10
=== Epoch 3/10 ===
Epoch 3 : 108 / 2254 samples , time for last 108 samples : 0:01:26 , time since beginning : 0:01:26 , time left to finish epoch : 0:28:46 , Loss/total: 7.95731, IoU: 0.03099, Accuracy: 3.10%
Epoch 3 : 216 / 2254 samples , time for last 108 samples : 0:01:22 , time since beginning : 0:02:49 , time left to finish epoch : 0:26:41 , Loss/total: 7.90549, IoU: 0.05310, Accuracy: 5.31%
Epoch 3 : 324 / 2254 samples , time for last 108 samples : 0:01:23 , time since beginning : 0:04:13 , time left to finish epoch : 0:25:10 , Loss/total: 7.91698, IoU: 0.05114, Accuracy: 5.11%
Epoch 3 : 432 / 2254 samples , time for last 108 samples : 0:01:16 , time since beginning : 0:05:29 , time left to finish epoch : 0:23:07 , Loss/total: 7.90776, IoU: 0.05181, Accuracy: 5.18%
Epoch 3 : 540 / 2254 samples , time for last 108 samples : 0:01:15 , time since beginning : 0:06:45 , time left to finish epoch : 0:21:22 , Loss/total: 7.91917, IoU: 0.05068, Accuracy: 5.07%
Epoch 3 : 648 / 2254 samples , time for last 108 samples : 0:01:15 , time since beginning : 0:08:00 , time left to finish epoch : 0:19:48 , Loss/total: 7.91751, IoU: 0.04951, Accuracy: 4.95%
Epoch 3 : 756 / 2254 samples , time for last 108 samples : 0:01:15 , time since beginning : 0:09:16 , time left to finish epoch : 0:18:04 , Loss/total: 7.91711, IoU: 0.04981, Accuracy: 4.98%
Epoch 3 : 864 / 2254 samples , time for last 108 samples : 0:01:15 , time since beginning : 0:10:32 , time left to finish epoch : 0:16:31 , Loss/total: 7.92056, IoU: 0.05164, Accuracy: 5.16%
Epoch 3 : 972 / 2254 samples , time for last 108 samples : 0:01:18 , time since beginning : 0:11:51 , time left to finish epoch : 0:15:05 , Loss/total: 7.91914, IoU: 0.05320, Accuracy: 5.32%
Epoch 3 : 1080 / 2254 samples , time for last 108 samples : 0:01:16 , time since beginning : 0:13:07 , time left to finish epoch : 0:13:49 , Loss/total: 7.91800, IoU: 0.05344, Accuracy: 5.34%
Epoch 3 : 1188 / 2254 samples , time for last 108 samples : 0:01:16 , time since beginning : 0:14:23 , time left to finish epoch : 0:12:34 , Loss/total: 7.91610, IoU: 0.05148, Accuracy: 5.15%
Epoch 3 : 1296 / 2254 samples , time for last 108 samples : 0:01:15 , time since beginning : 0:15:39 , time left to finish epoch : 0:11:17 , Loss/total: 7.91286, IoU: 0.05154, Accuracy: 5.15%
Epoch 3 : 1404 / 2254 samples , time for last 108 samples : 0:01:15 , time since beginning : 0:16:54 , time left to finish epoch : 0:10:00 , Loss/total: 7.91214, IoU: 0.05137, Accuracy: 5.14%
Epoch 3 : 1512 / 2254 samples , time for last 108 samples : 0:01:16 , time since beginning : 0:18:10 , time left to finish epoch : 0:08:44 , Loss/total: 7.90571, IoU: 0.04950, Accuracy: 4.95%
Epoch 3 : 1620 / 2254 samples , time for last 108 samples : 0:01:18 , time since beginning : 0:19:29 , time left to finish epoch : 0:07:27 , Loss/total: 7.90297, IoU: 0.04898, Accuracy: 4.90%
Epoch 3 : 1728 / 2254 samples , time for last 108 samples : 0:01:15 , time since beginning : 0:20:45 , time left to finish epoch : 0:06:11 , Loss/total: 7.89967, IoU: 0.04878, Accuracy: 4.88%
Epoch 3 : 1836 / 2254 samples , time for last 108 samples : 0:01:16 , time since beginning : 0:22:01 , time left to finish epoch : 0:04:55 , Loss/total: 7.89428, IoU: 0.04854, Accuracy: 4.85%
Epoch 3 : 1944 / 2254 samples , time for last 108 samples : 0:01:15 , time since beginning : 0:23:17 , time left to finish epoch : 0:03:38 , Loss/total: 7.89330, IoU: 0.04704, Accuracy: 4.70%
Epoch 3 : 2052 / 2254 samples , time for last 108 samples : 0:01:16 , time since beginning : 0:24:33 , time left to finish epoch : 0:02:23 , Loss/total: 7.88822, IoU: 0.04682, Accuracy: 4.68%
Epoch 3 : 2160 / 2254 samples , time for last 108 samples : 0:01:16 , time since beginning : 0:25:49 , time left to finish epoch : 0:01:06 , Loss/total: 7.88518, IoU: 0.04623, Accuracy: 4.62%
Epoch 3 : 2256 / 2254 samples , time for last 96 samples : 0:01:08 , time since beginning : 0:26:58 , time left to finish epoch : 0:00:00 , Loss/total: 7.88523, IoU: 0.04565, Accuracy: 4.57%
==
✅ Epoch 3 Summary (train):
Total samples processed: 2256
Total epoch time: 0:26:59
Average Loss: 7.88523
Average IoU: 0.04565
Average Accuracy: 4.57%
==
💾 Saved checkpoint: /home/farha/assignment_3/SeqTrack/lib/train/outputs/checkpoints/checkpoint_epoch_3.pth
=== Epoch 4/10 ===
Epoch 4 : 108 / 2254 samples , time for last 108 samples : 0:01:21 , time since beginning : 0:01:21 , time left to finish epoch : 0:27:00 , Loss/total: 7.84457, IoU: 0.03288, Accuracy: 3.29%
Epoch 4 : 216 / 2254 samples , time for last 108 samples : 0:01:16 , time since beginning : 0:02:37 , time left to finish epoch : 0:24:47 , Loss/total: 7.83162, IoU: 0.03579, Accuracy: 3.58%
Epoch 4 : 324 / 2254 samples , time for last 108 samples : 0:01:14 , time since beginning : 0:03:52 , time left to finish epoch : 0:23:04 , Loss/total: 7.83728, IoU: 0.03935, Accuracy: 3.93%
26°C
صافي غالبًا
Search
ENG
10:22 PM
10/22/2025

8. Submission and Deliverables

All required files and folders are placed inside the unified project folder named **assignment_3** to be submitted on GitHub and the team's private channel.

8.1 Project Folder Structure Confirmation

The local directory structure mirrors the submission requirements:

```
assignment_3/
├──── SeqTrack /
          └──── data/
                └──── lasot/
                      ├──── coin/
                      ├──── kite/
          └──── lib/train/outputs/
                      ├──── checkpoints / (placed on Hugging face)
                            └──── checkpoint_epoch_1.pth
                            └──── checkpoint_epoch_2.pth
                            └──── checkpoint_epoch_3.pth
                            └──── checkpoint_epoch_4.pth
                            └──── checkpoint_epoch_5.pth
                            └──── checkpoint_epoch_6.pth
                            └──── checkpoint_epoch_7.pth
                            └──── checkpoint_epoch_8.pth
                            └──── checkpoint_epoch_9.pth
                            └──── checkpoint_epoch_10.pth
          └──── lib/train/outputs/
                      ├──── logs /
                            └──── seqtrack-seqtrack_b256.log
                            └──── seqtrack-seqtrack_b256-phase2.log

├──── requirements.txt
```