Report on MULDE: Multiscale Log-Density Estimation via Denoising Score Matching for Video Anomaly Detection

Divyansh Mathur
Student ID: 210357

October 22, 2024

Implementation Details

1. **Environment Setup:**
   - Utilize the provided `environment.yml` to set up the required environment. Execute the following command:

     ```
     conda env create -f environment.yml
     ```

   - Activate the newly created environment using:

     ```
     conda activate mulde
     ```

2. **GPU Configuration:**
   - If your device is equipped with a GPU, uncomment the line containing `torch.cuda.empty_cache()` in `main.py` while loading the header files.
   - Change the default device to `cuda` for GPU testing. If a GPU is unavailable, ensure the device is set to `cpu`.
3. **Dataset Loader Design:**
   - The original implementation utilized a toy dataset. In this repository, a dataset loader for the Ped2 dataset has been added.
   - For other datasets, adjust the dataset location and ground truth `.m` files accordingly.
4. **Model Training and Evaluation:**
   - Train and evaluate the model by running:

     ```
     python main.py --plot_dataset --gmm
     ```

5. **Viewing Results:**
   - After training and testing, visualize the results using TensorBoard:

     ```
     tensorboard --logdir=runs/MULDE --samples_per_plugin images=100
     ```

   - Open the provided localhost link to view the ROC_AUC scores after each epoch and the final score upon completion.
6. **Saving Model Weights:**
   - Save the model weights for future iterations by executing:

     ```
     python3 weight_saver.py
     ```

Results

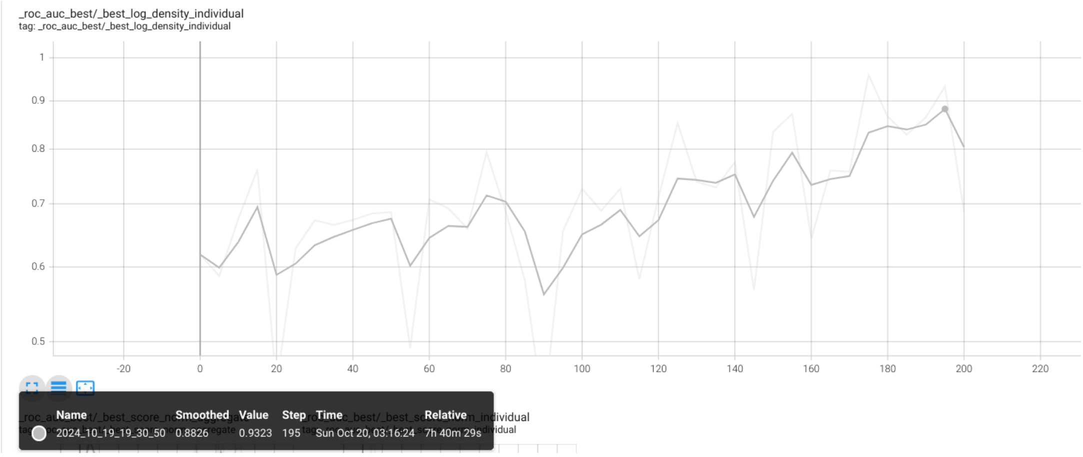
The model was evaluated on two datasets: `USCD_Anomaly_Detection_Dataset_Ped1` and `Ped2`. The results for three iterations are presented below.

Iteration 3: Optimal Dataset and Epochs

- **Configuration:**
  - Training data reduced to 13 video files.
  - Number of epochs set to 200.
- **Observations:**
  - The results are close to the expected performance.
  - Minor discrepancies are attributed to underfitting due to reduced epochs and training data.
- **Results:**

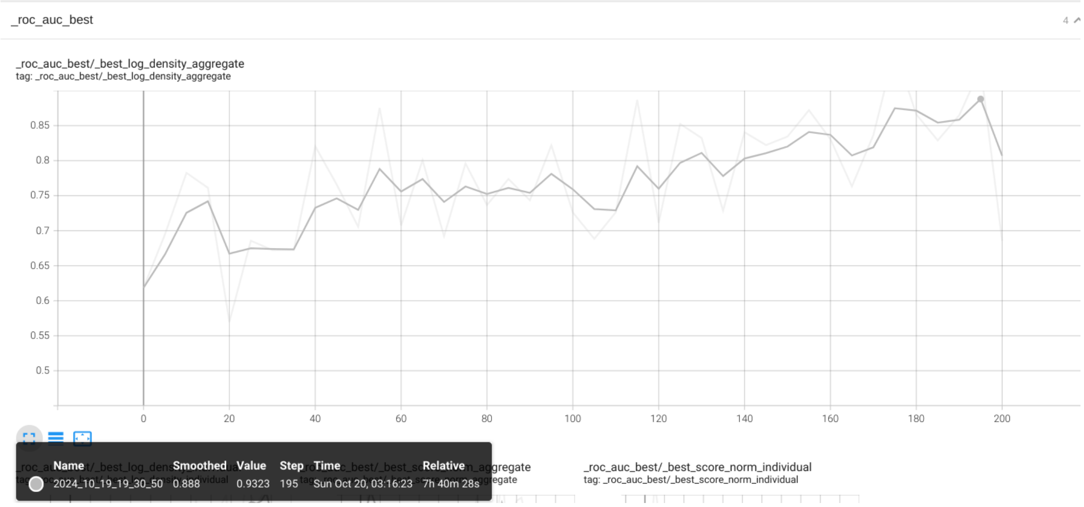
Table 1: ROC_AUC Scores for Iteration 3

Metric	Evaluated Result	Expected Result
Micro	93.5%	99.7%
Macro	93.23%	99.9%



Name	Smoothed	Value	Step	Time	Relative
○ 2024_10_19_19_30_50	0.8826	0.9323	195	Sun Oct 20, 03:16:24	7h 40m 29s

Figure 1: ROC_AUC Score for Micro



Name	Smoothed	Value	Step	Time	Relative
○ 2024_10_19_19_30_50	0.888	0.9323	195	Sun Oct 20, 03:16:23	7h 40m 28s

Figure 2: ROC_AUC Score for Macro

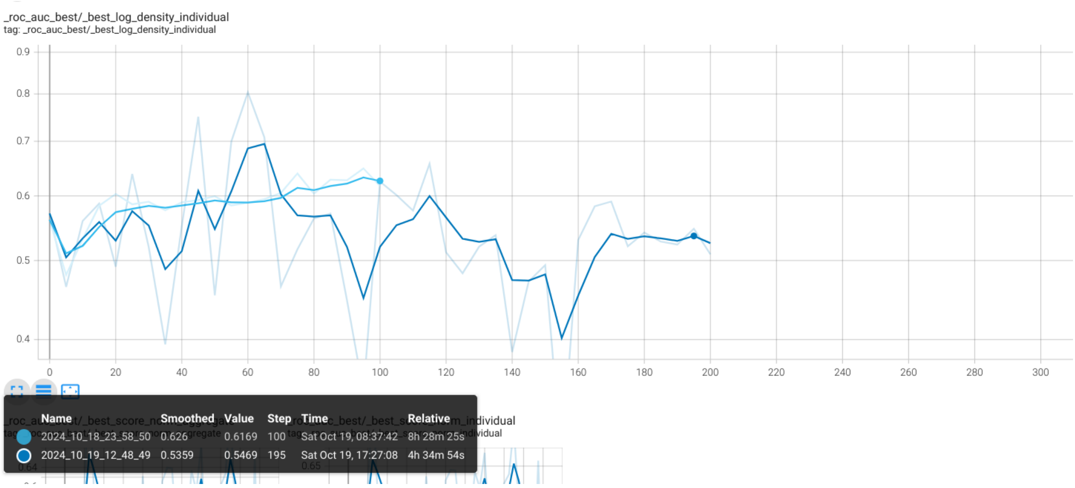
Iterations 1 and 2: Reduced Epochs and Training Data

- **Configuration:**

- **Iteration 1:** Reduced epochs to 100.
- **Iteration 2:** Reduced training dataset to 7 videos.

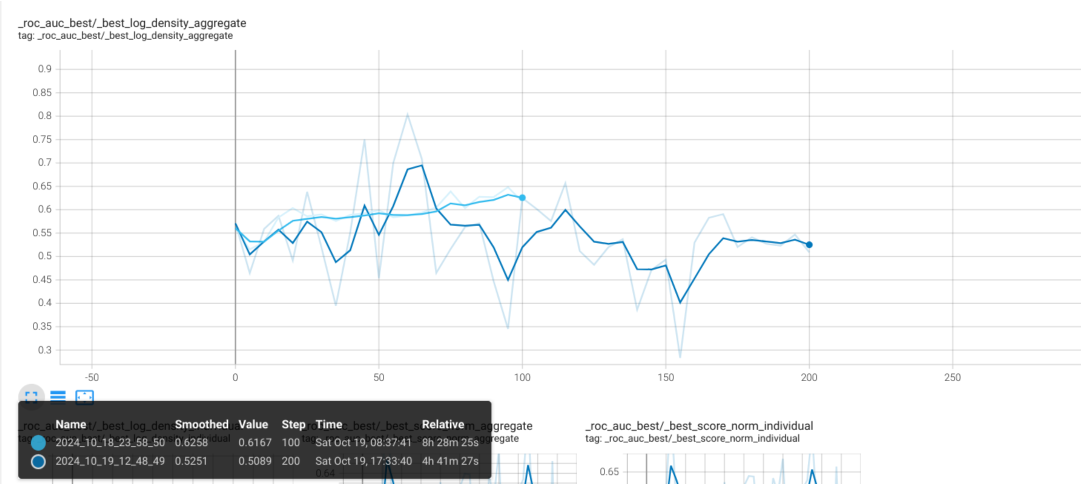
- **Observations:**
  - Both iterations yielded unsatisfactory results.
  - **Iteration 1:** Model underfitted due to insufficient epochs.
  - **Iteration 2:** Model overfitted, resulting in random outputs.

- **Results:**



	Name	Smoothed	Value	Step	Time	Relative
●	2024_10_18_23_58_50	0.626	0.6169	100	Sat Oct 19, 08:37:42	8h 28m 25s
○	2024_10_19_12_48_49	0.5359	0.5469	195	Sat Oct 19, 17:27:08	4h 34m 54s

Figure 3: ROC_AUC Scores for Failed Iterations (Micro)



	Name	Smoothed	Value	Step	Time	Relative
●	2024_10_18_23_58_50	0.6258	0.6167	100	Sat Oct 19, 08:37:41	8h 28m 25s
○	2024_10_19_12_48_49	0.5251	0.5089	200	Sat Oct 19, 17:33:40	4h 41m 27s

Figure 4: ROC_AUC Scores for Failed Iterations (Macro)

- **Conclusion:**
  - Iteration 1's reduced epochs led to underfitting.
  - Iteration 2's reduced training data caused overfitting.

Dataset Description

The dataset utilized in this implementation is the `USCD_Anomaly_Detection_Pedestrian_Camera_Ped2`. It comprises:

- **Training Videos:** 16 original videos, reduced to 13 due to computational constraints.
- **Testing Videos:** 12 original videos, reduced to 9.

- **Frames per Video:** Each video contains 200 TIFF files, representing 200 frames.
- **Total Size:** Approximately 12 GB.

Due to limited computational resources, the dataset was downsized to include only 13 training videos and 9 testing videos.
One can download the original DataSet from here: USCD_Anomaly$_detection$
Other datasets were very huge in size hence training in them was avoided due to resource constraints.

Efforts and Challenges

- **Environment Setup:** I encountered difficulties setting up the environment on both Windows and Kaggle platforms. Consequently, I opted to work on a Linux system to create the necessary environment and successfully run the model.
- **Limited Resources:** Training the model for 200 epochs on a CPU device required approximately 8 hours. To achieve satisfactory results, reducing the dataset was the only viable option. Notably, the author utilized datasets as large as 500 GB, which posed significant resource demands.
- **Dataset Loader Development:** The original implementation did not provide a dataset loader, as the author conducted training and testing on a toy dataset. To address this, I developed a dataset loader tailored for the Ped2 dataset, facilitating more robust training and evaluation.