

Train Dataloader - 48

Test Dataloader - 173

Device Used - cuda

Model Used - Base_3DCAE

Feature Extraction - True

Background Subtraction - True

Background Subtraction Algorithm - MOG2

Data Augmentation - False

Window Length = 8

Stride = 1

Fair Comparison = True

Dropout = 0.25

Learning Rate = 0.0002

Num Epochs = 20

Chunk Size = 64

Forward Chunk = 8

Forward Chunk Size = 8

Loss Fn = L1Loss()

Training has Begun

epoch [1/20], loss:1.0110

epoch [2/20], loss:1.0091

epoch [3/20], loss:1.0084

epoch [4/20], loss:1.0080

epoch [5/20], loss:1.0077

epoch [6/20], loss:1.0075

epoch [7/20], loss:1.0074

epoch [8/20], loss:1.0073

epoch [9/20], loss:1.0073

epoch [10/20], loss:1.0073

epoch [11/20], loss:1.0072

epoch [12/20], loss:1.0071

epoch [13/20], loss:1.0071

epoch [14/20], loss:1.0071

epoch [15/20], loss:1.0071

epoch [16/20], loss:1.0071

epoch [17/20], loss:1.0071

epoch [18/20], loss:1.0070

epoch [19/20], loss:1.0070

```
c:\Users\abdul\anaconda3\envs\fyp_base_paper_2\lib\site-packages\numpy\lib\numpy.py:528: VisibleDeprecationWarning: Creating an ndarray from ragged nested sequences (which is a list-or-tuple of lists-or-tuples-or ndarrays with different lengths or shapes) is deprecated. If you meant to do this, you must specify 'dtype=object' when creating the ndarray.
```

```
arr = np.asanyarray(arr)
```

```
# https://docs.opencv.org/4.x/d7/d7b/classcv_1_1BackgroundSubtractorMOG2.html
def perform_background_subtraction_MOG2(vid_total):
    background_subtracted_vid_total = []

    # Create background subtractor
    bg_subtractor = cv2.createBackgroundSubtractorMOG2()
    # Sets the number of last frames that affect the background model.
    # bg_subtractor.setHistory(history=50) # Default - 500
    # bg_subtractor.setDetectShadows(False) # Default - True
    # Sets the variance threshold for the pixel-model match. (Pixels whose vari
    # bg_subtractor.setVarThreshold(200) # Default - 16.0

    for frame in vid_total:
        # Generated foreground is always black due to the preprocessing step -
        frame = frame * 255
        # print(frame.shape)
        # Perform background subtraction. learningRate = -1 is default
        foreground_mask = bg_subtractor.apply(frame, learningRate=-1)
        background_subtracted_vid_total.append(foreground_mask)

        ## To view the images
        # cv2.imshow("Original Frame", frame)
        # cv2.imshow("Foreground Mask - MOG2", foreground_mask)
        # # Exit on 'q' press
        # k = cv2.waitKey(30) & 0xFF
        # if k == 27:
        #     break

    return background_subtracted_vid_total
```

epoch [20/20], loss:1.0071
Training has Completed

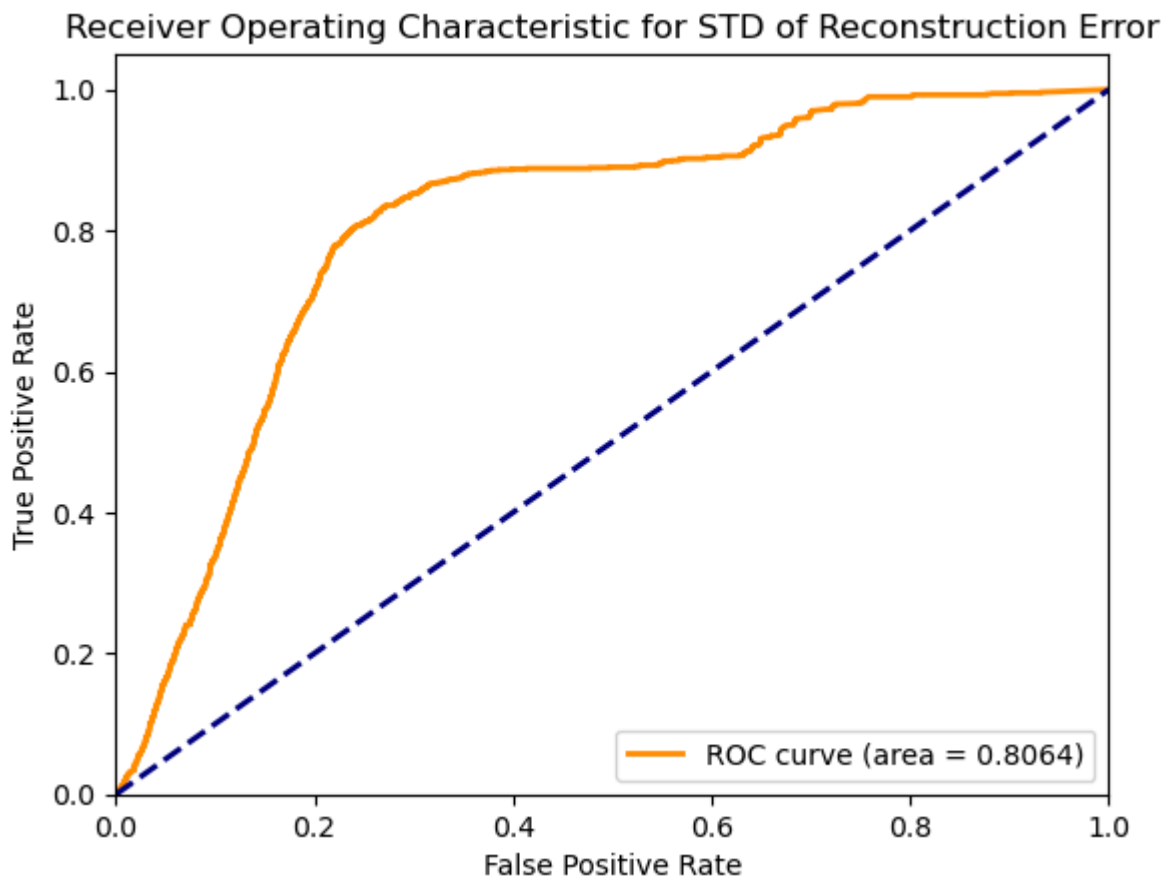
Forward pass occuring
Forward pass completed

Thermal_T3_2024-03-14-04-05-44

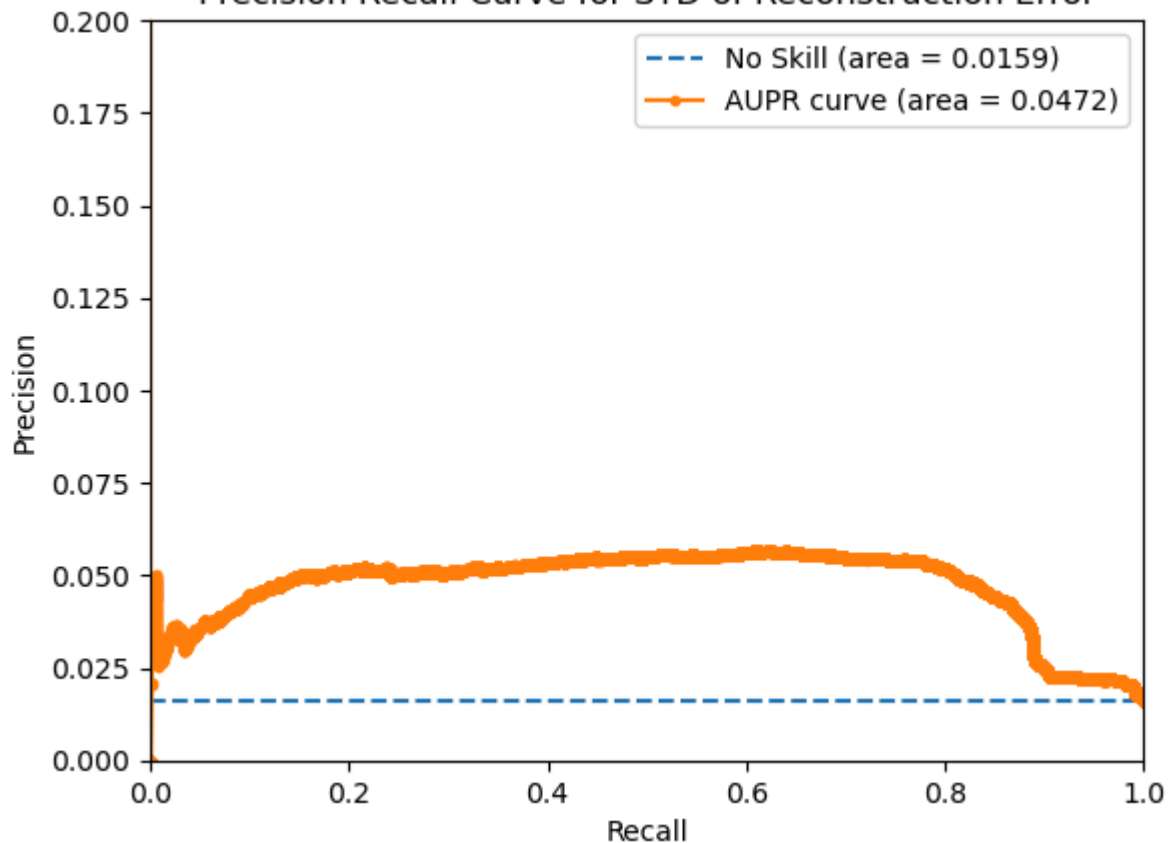
STD Global Classification Results
TPR 0.807, FPR 0.242, Precision 0.051, Recall 0.807
tn 50322, fp 16067, fn 207, tp 865
std_AUROC 0.806

Mean Global Classification Results
TPR 0.765, FPR 0.271, Precision 0.044, Recall 0.765
tn 48413, fp 17976, fn 252, tp 820
mean_AUROC 0.792

```
d:\Abdul Rasheed NITT\Academics\Eigth Semester\FYP\Implementation\FallDetection\Code\functions.py:250: RuntimeWarning: Mean of empty slice
  final_performance_mean = np.nanmean(video_metrics, axis=0) # get the mean performance a
cross all videos
c:\Users\abdul\anaconda3\envs\fyp_base_paper_2\lib\site-packages\numpy\lib\nanfunctions.py:1670: RuntimeWarning: Degrees of freedom <= 0 for slice.
  var = nanvar(a, axis=axis, dtype=dtype, out=out, ddof=ddof,
```



Precision Recall Curve for STD of Reconstruction Error



Receiver Operating Characteristic for Mean of Reconstruction Error

