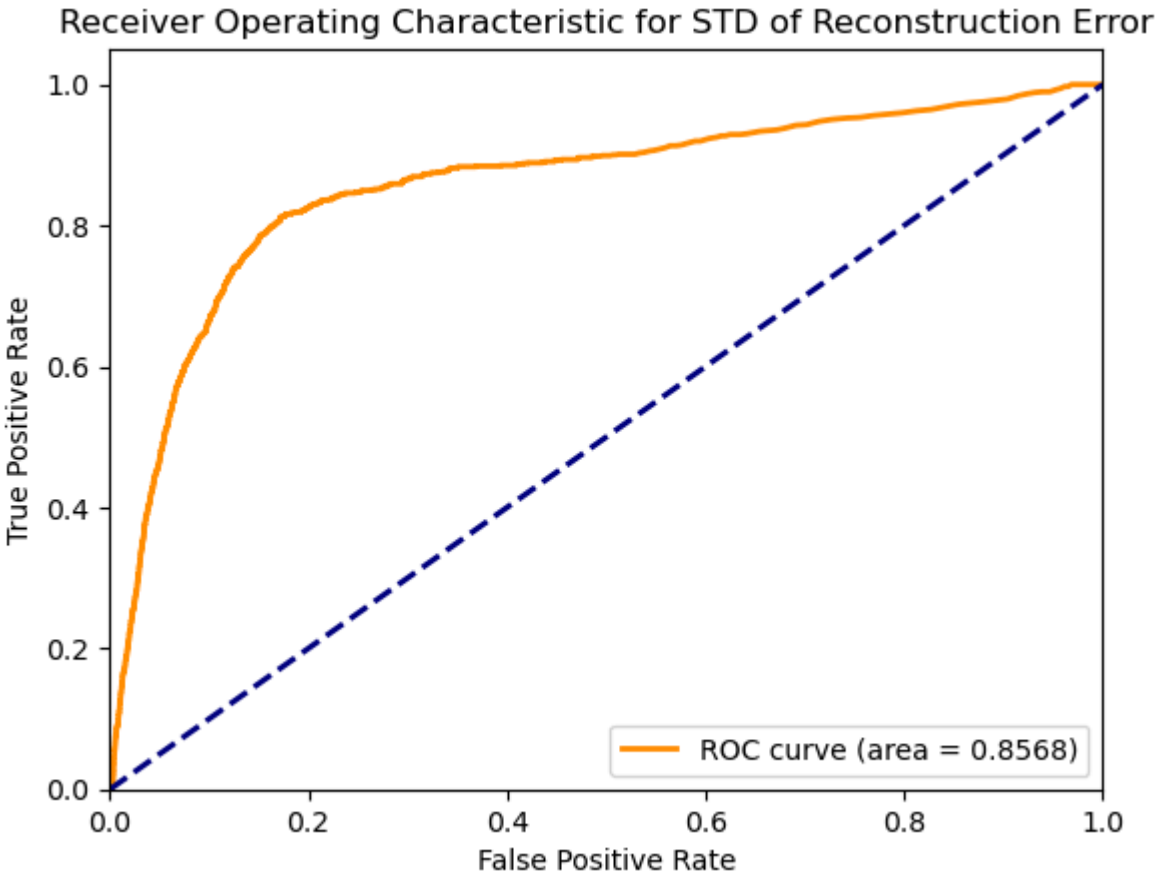
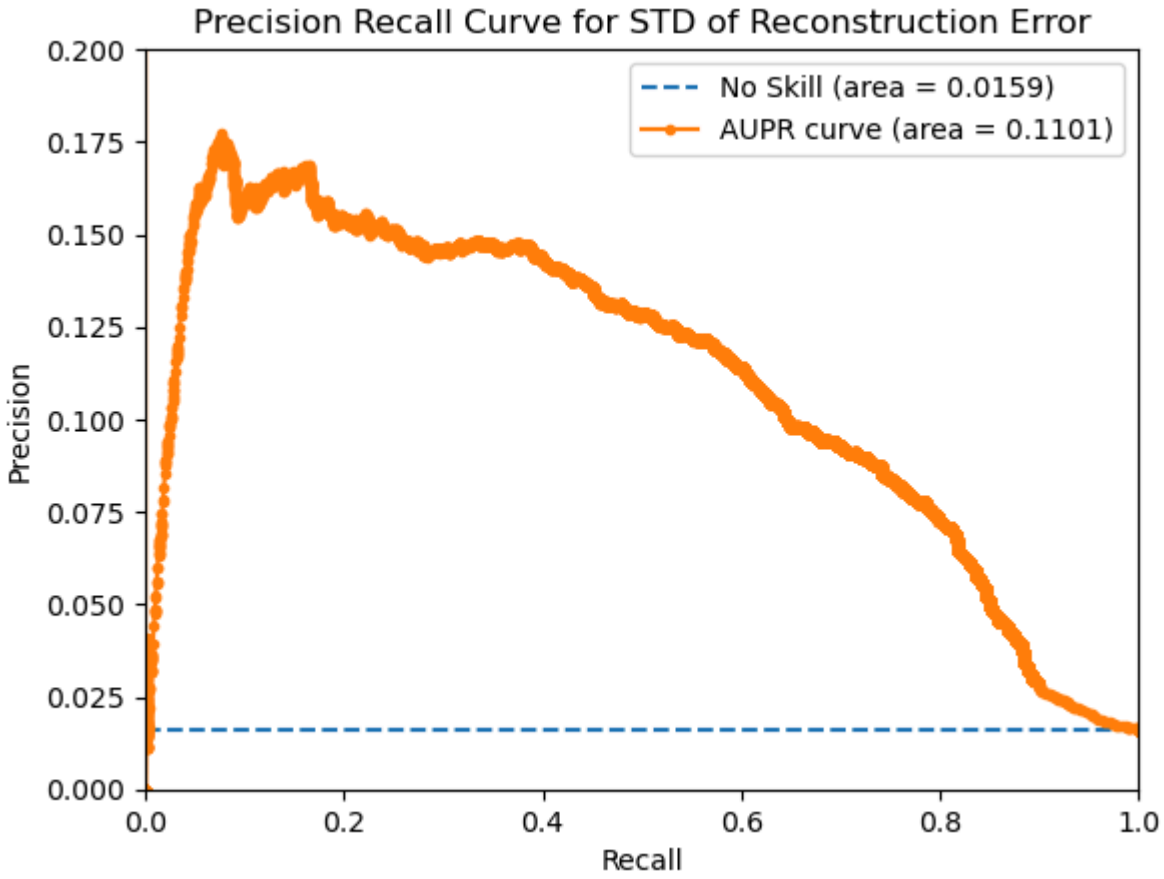
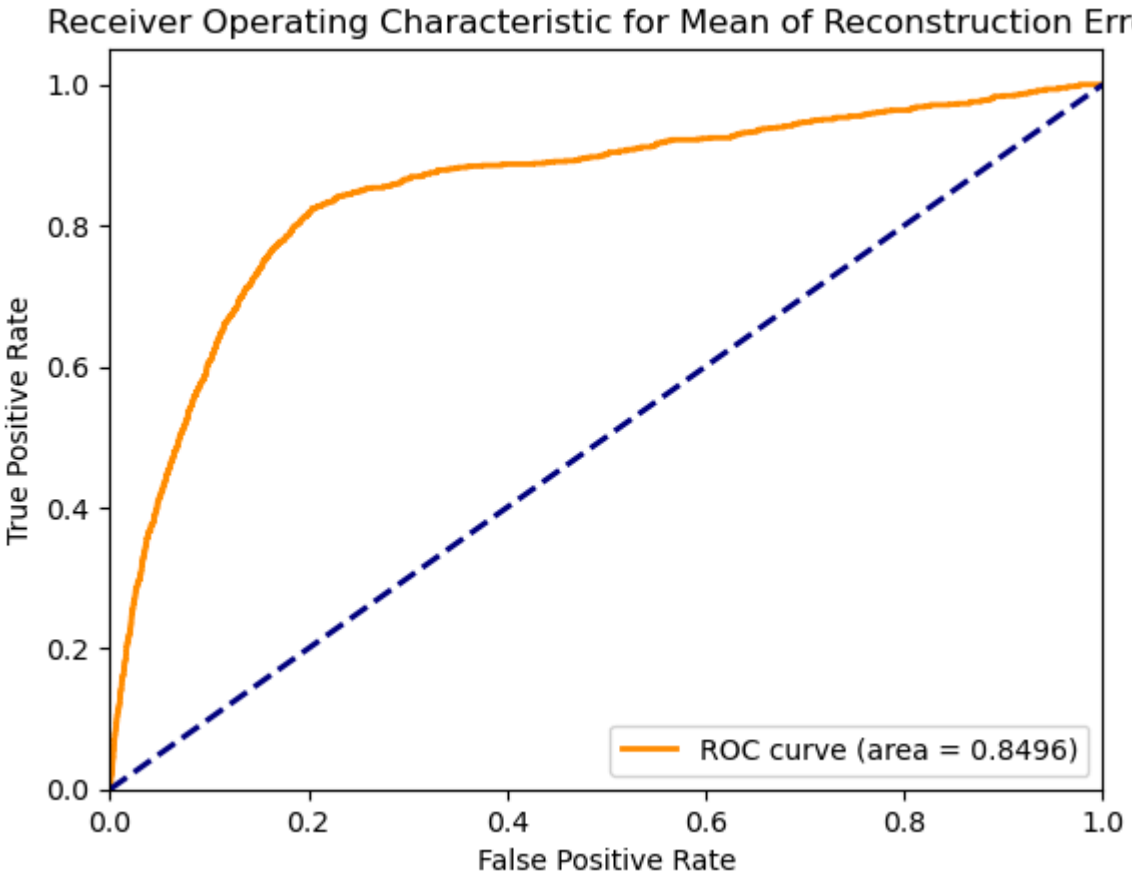
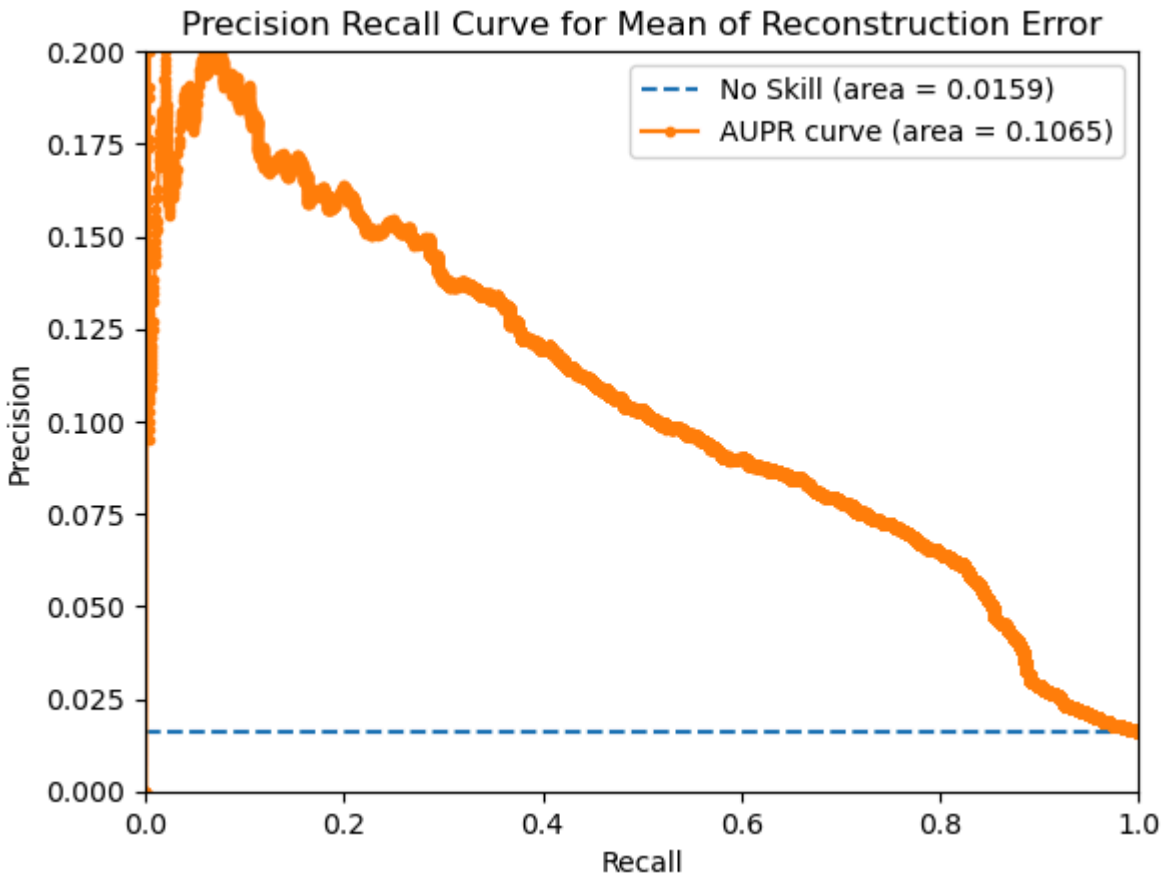
```
Train Dataloader - 48
Test Dataloader - 173

Device Used - cuda

Model Used - Base_3DCAE_2
Feature Extraction - True
Background Subtraction - True
Background Subtraction Algorithm - GMG
Data Augmentation - False
Spatial Temporal Loss - True
w1 - 1, w2 - 1e-05

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:0.6084
epoch [2/20], loss:0.6051
epoch [3/20], loss:0.6042
epoch [4/20], loss:0.6036
epoch [5/20], loss:0.6033
epoch [6/20], loss:0.6031
epoch [7/20], loss:0.6029
epoch [8/20], loss:0.6028
epoch [9/20], loss:0.6027
epoch [10/20], loss:0.6026
epoch [11/20], loss:0.6025
epoch [12/20], loss:0.6024
epoch [13/20], loss:0.6024
epoch [14/20], loss:0.6023
epoch [15/20], loss:0.6023
epoch [16/20], loss:0.6022
epoch [17/20], loss:0.6022
epoch [18/20], loss:0.6021
epoch [19/20], loss:0.6021
c:\Users\abdul\anaconda3\envs\fyp_base_paper_2\lib\site-packages\numpy\lib\npyio.py:528: V
isibleDeprecationWarning: Creating an ndarray from ragged nested sequences (which is a lis
t-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)
```

```
epoch [20/20], loss:0.6021
Training has Completed

Forward pass occuring
Forward pass completed

Thermal_T3_2024-03-20-01-25-06

---------------------------------
STD Global Classification Results
TPR 0.812, FPR 0.172, Precision 0.071, Recall 0.812
tn 54965, fp 11424, fn 201, tp 871
std_AUROC  0.857
---------------------------------
---------------------------------
Mean Global Classification Results
TPR 0.825, FPR 0.204, Precision 0.061, Recall 0.825
tn 52863, fp 13526, fn 188, tp 884
mean_AUROC 0.850
---------------------------------
d:\Abdul Rasheed NITT\Academics\Eigth Semester\FYP\Implementation\FallDetection\Code\funct
ions.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.p
y: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
No Skill (area = 0.0159)
AUPR curve (area = 0.1101)
Precision
Recall
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0.0
0.2
0.4
0.6
0.8
1.0


Receiver Operating Characteristic for Mean of Reconstruction Error
ROC curve (area = 0.8496)
True Positive Rate
False Positive Rate
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0


Precision Recall Curve for Mean of Reconstruction Error
No Skill (area = 0.0159)
AUPR curve (area = 0.1065)
Precision
Recall
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0.0
0.2
0.4
0.6
0.8
1.0