

Train Dataloader - 48

Test Dataloader - 173

Device Used - cuda

Model Used - Base_3DCAE

Feature Extraction - True

Background Subtraction - True

Background Subtraction Algorithm - GMG

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:0.5615

epoch [2/20], loss:0.5586

epoch [3/20], loss:0.5573

epoch [4/20], loss:0.5570

epoch [5/20], loss:0.5570

epoch [6/20], loss:0.5564

epoch [7/20], loss:0.5562

epoch [8/20], loss:0.5560

epoch [9/20], loss:0.5561

epoch [10/20], loss:0.5559

epoch [11/20], loss:0.5558

epoch [12/20], loss:0.5558

epoch [13/20], loss:0.5557

epoch [14/20], loss:0.5556

epoch [15/20], loss:0.5556

epoch [16/20], loss:0.5556

epoch [17/20], loss:0.5555

epoch [18/20], loss:0.5554

epoch [19/20], loss:0.5554

```
# https://docs.opencv.org/4.x/d1/d5c/classcv_1_bgsegm_1_1BackgroundSubtractorGMG.html
def perform_background_subtraction_GMG(vid_total):
    background_subtracted_vid_total = []

    # Create background subtractor
    bg_subtractor = cv2.bgsegm.createBackgroundSubtractorGMG()
    kernel = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (3, 3))
    # bg_subtractor.setDecisionThreshold(0.8) # Default - 0.8
    # bg_subtractor.setDefualtLearningRate(0.025) # Default - 0.025
    bg_subtractor.setNumFrames(50) # Default - 120

    for frame in vid_total:
        frame = np.array(frame, dtype=np.float32)
        # print(frame.shape)
        # Perform background subtraction
        foreground_mask = bg_subtractor.apply(frame)
        foreground_mask = cv2.morphologyEx(foreground_mask, cv2.MORPH_OPEN, kernel)
        background_subtracted_vid_total.append(foreground_mask)

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

    return background_subtracted_vid_total
```

c:\Users\abdul\anaconda3\envs\fyp_base_paper_2\lib\site-packages\numpy\lib\npyio.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)
```

epoch [20/20], loss:0.5553
Training has Completed

Forward pass occurring
Forward pass completed

Thermal_T3_2024-03-14-04-26-48

STD Global Classification Results
TPR 0.816, FPR 0.203, Precision 0.061, Recall 0.816
tn 52892, fp 13497, fn 197, tp 875
std_AUROC 0.845

Mean Global Classification Results
TPR 0.847, FPR 0.283, Precision 0.046, Recall 0.847
tn 47603, fp 18786, fn 164, tp 908
mean_AUROC 0.822

```
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,
```





