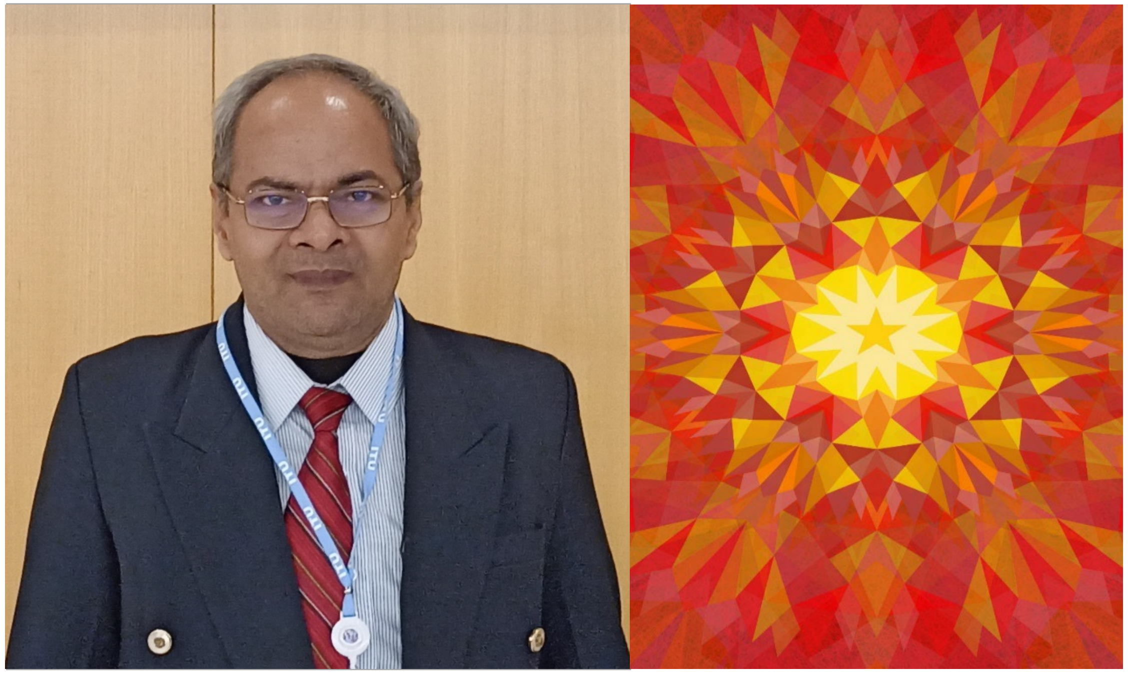
ITU KALEIDOSCOPE
ONLINE2020

7-11 December 2020

Visual Action Recognition Using Deep Learning in Video Surveillance Systems

Dhananjay Kumar
Department of Information Technology
Anna University, MIT Campus
Chennai, India



Session 8: Security in industrial applications

Paper S8.2

ITU KALEIDOSCOPE
ONLINE2020

Outlines

- Introduction
- Motivation
- System architecture
- Feature Vector Optimization
- Experimental results
- Summary

Introduction

Approaches in Sensor-based Human Activity Recognition

- Logic and reasoning
  - Inherent infeasibility to handle uncertainty
  - Limitation of learning ability with logic based techniques
- Probabilistic model
  - Generative (e.g., HMM, Bayesian Networks) or discriminative models (e.g., Conditional random fields)
- Data mining-based methods
  - Mining a set of pattern of features, activity model
  - **Skeleton featured-based**
    - Skeleton feature of human-subjects with different body positions

Motivation

Learning based Approaches for Vision-based Activity Recognition

- **Before Deep Learning**
- Hand crafted features (e.g., HOG) from sparsely / densely sampled trajectories
- Hand-crafted vs. learned features
  - Bag of words
  - Frame level processing
- **Post Deep Learning Approaches**
- The fusion of spatial and temporal data across streams
- Creation of multi-level loss to handle temporal dependencies in long term

System Architecture

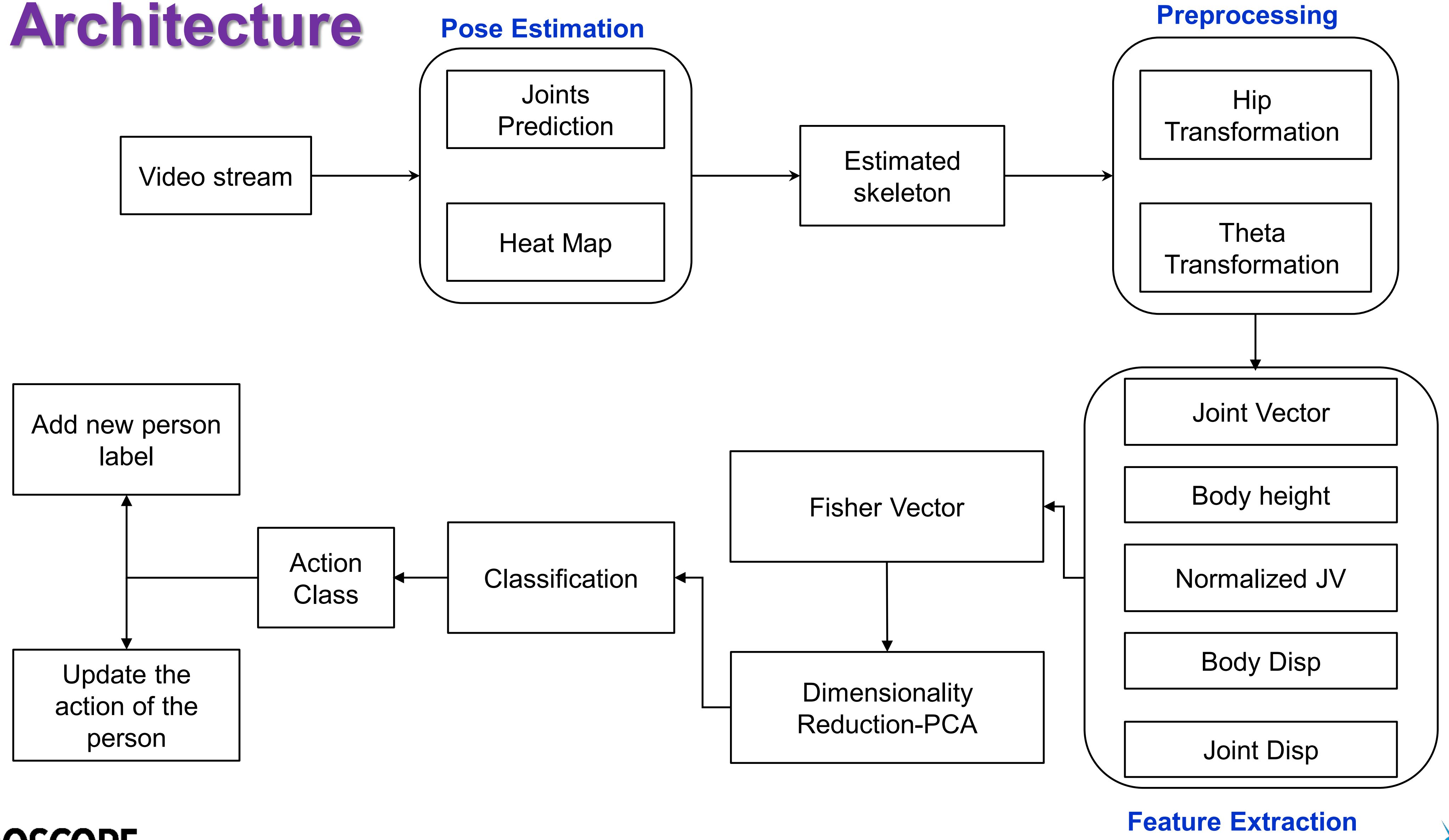
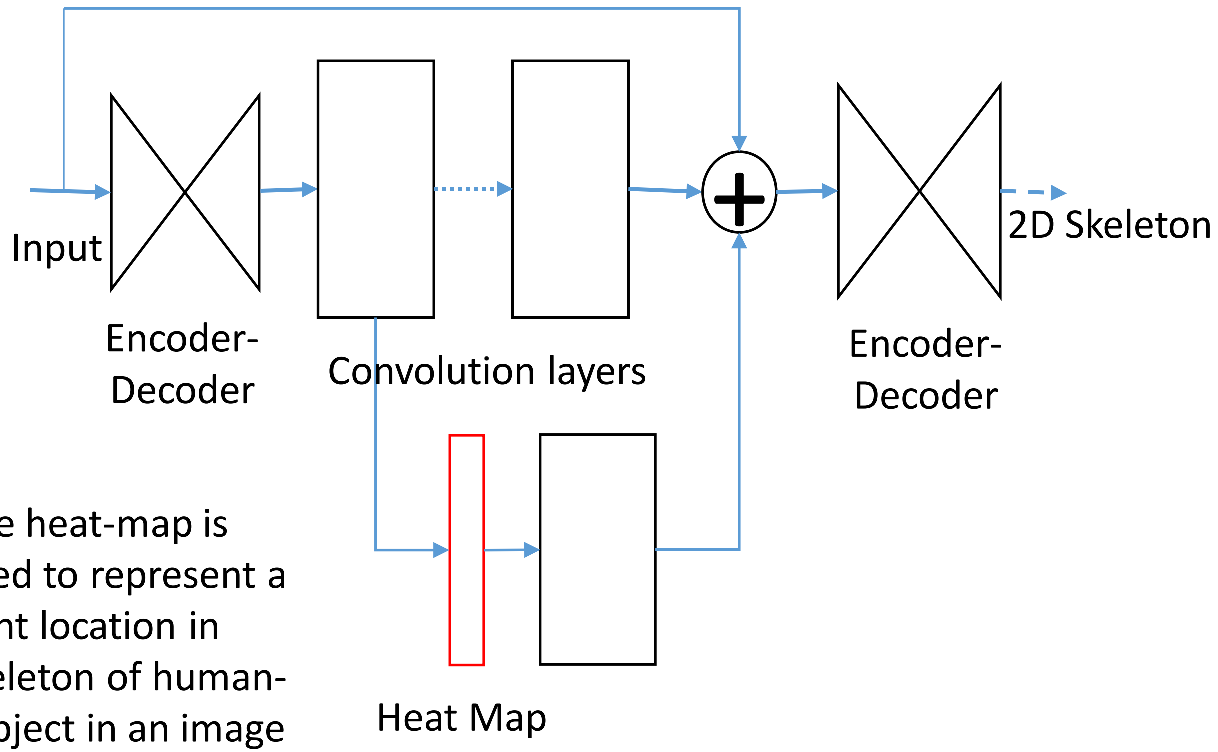


Figure 1 – The architecture of the proposed model

Skeleton Generation and Processing



The heat-map is used to represent a joint location in skeleton of human-subject in an image

Figure 2 - Stacked hourglass architecture for 2D skeleton

1. Hip Transformation

To make the skeletons invariant to the location of the subjects,

$$[x'_j, y'_j] = [x_j - x_{hipcenter}, y_j - y_{hipcenter}]$$

Where $x_{hipcenter}$ and $y_{hipcenter}$ represent the hip center of the input skeleton

2. Theta Transformation

To make the poses rotation invariant, a rotation operation is applied on the joints relative to the camera view angle ϑ.

$$\theta = \tan^{-1}\left(\frac{y_{\text{right\_hip}} - y_{\text{left\_hip}}}{x_{\text{right\_hip}} - x_{\text{left\_hip}}}\right)$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 1 \\ -\sin\theta & \cos\theta & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Feature Vector Optimization

The **Fisher Vector** (FV) and **dimensionality reduction** using **PCA** are applied for the optimization of the features.

The FV encodes the gradients of the log-likelihood of the features under the Gaussian-Mixture-Model (GMM), with respect to the GMM parameters.

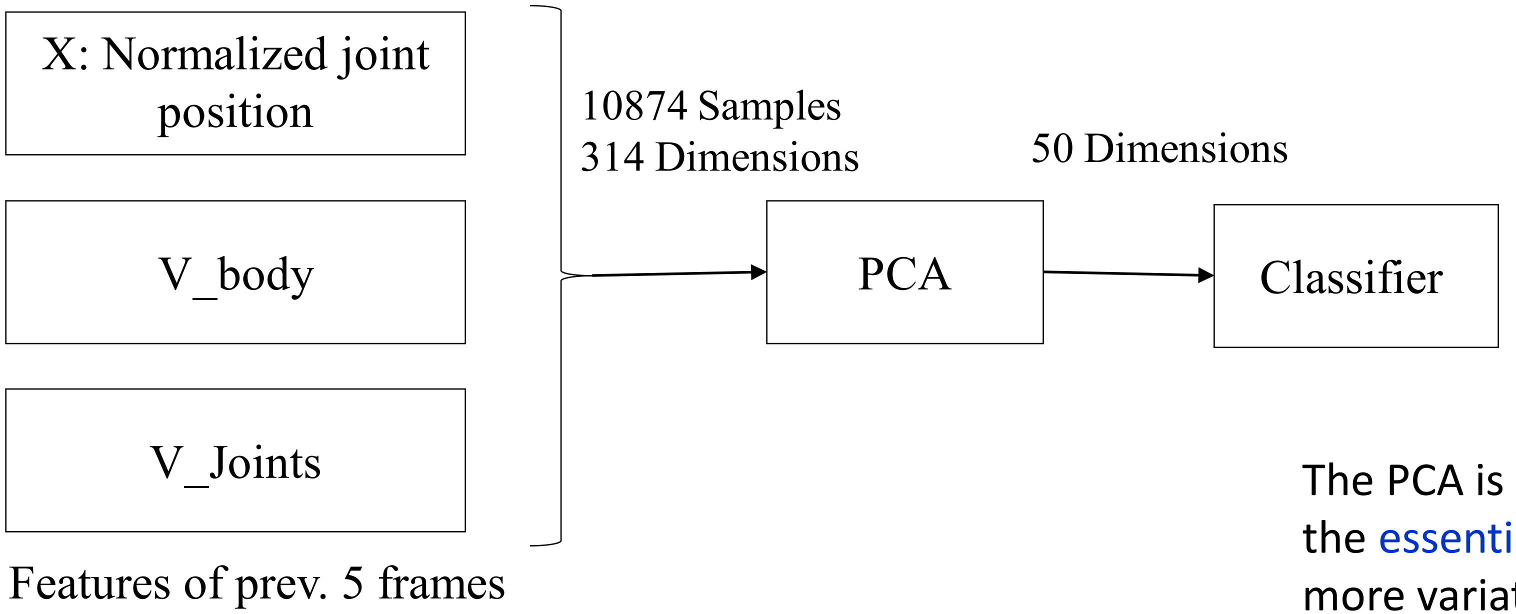


Figure 3 - Feature vector optimization

The multi-class classification is carried out by a DNN or SVM

The PCA is used to preserve the essential parts that have more variation of the data and remove the non-essential parts with fewer variations.

Implementation Overview

The data sets used to train and evaluate the model:

- *MSR Action* Dataset
- *NTU RGB+D 3D Skeletal* Dataset

Implementation using Python programming *with*

- ***Flask* framework** - Web server implementation
- ***OpenCV* library** – Processing of video stream at frame level
- ***Keras* library with *Tensorflow*** - *Design of* convolutional neural networks

The **multi-class classification** is carried out by either

(1) A **one-vs-rest** SVM or

(2) A **three-layer** multi-layer perceptron (MLP) **DNN**

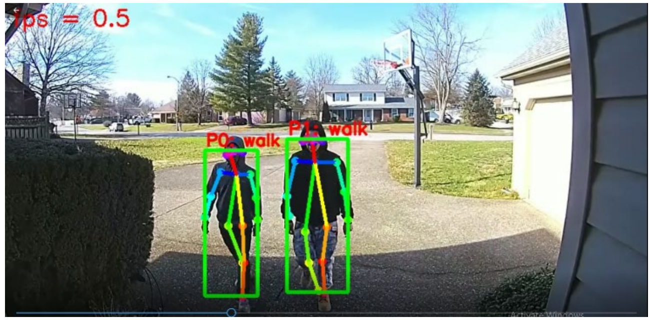


Fig: System display of the recognized action

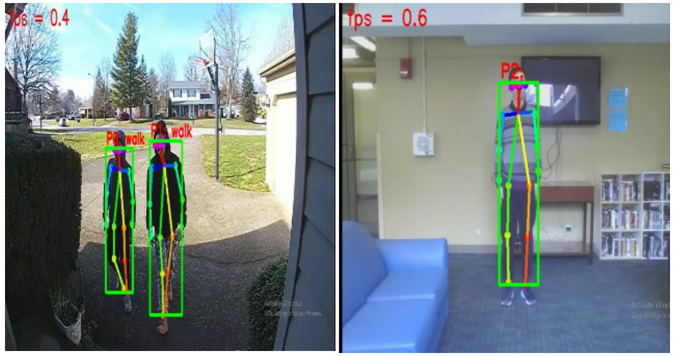


Fig. Bounding boxes in two MSR action data sets

Results

The SVM is trained with the help of a feature vector generated from the MSR Action Data Set.

Table 1: Confusion matrix of the one-vs.-rest SVM

Action label	Wave	Punch	Kick	Squat	Sit	Jump	Run	Walk	Stand
Wave	6	0	1	1	0	0	1	0	343
Punch	0	0	0	1	0	0	0	237	5
Kick	15	4	6	3	0	0	296	0	2
Squat	0	0	0	0	0	279	0	0	0
Sit	0	0	0	0	562	0	1	0	0
Jump	39	5	23	246	0	0	3	0	6
Run	6	6	274	11	0	0	9	0	0
Walk	26	327	5	1	0	0	2	0	0
Stand	460	33	7	15	0	0	0	1	3

Table 2: Performance metrics of the action classifier

Class	Precision	Recall	F1-score	Support
Stand	0.83	0.89	0.86	519
Walk	0.87	0.91	0.89	361
Run	0.87	0.9	0.88	306
Jump	0.88	0.76	0.82	322
Sit	1	1	1	563
Squat	1	1	1	279
Kick	0.95	0.93	0.93	326
Punch	1	0.99	0.99	243
Wave	0.96	0.96	0.96	352

$$Precision = \frac{TruePositives}{(TruePositives + FalsePositives\ )}$$

$$Recall = \frac{TruePositives}{(TruePositives + FalseNegatives\ )}$$

Results (cont.)

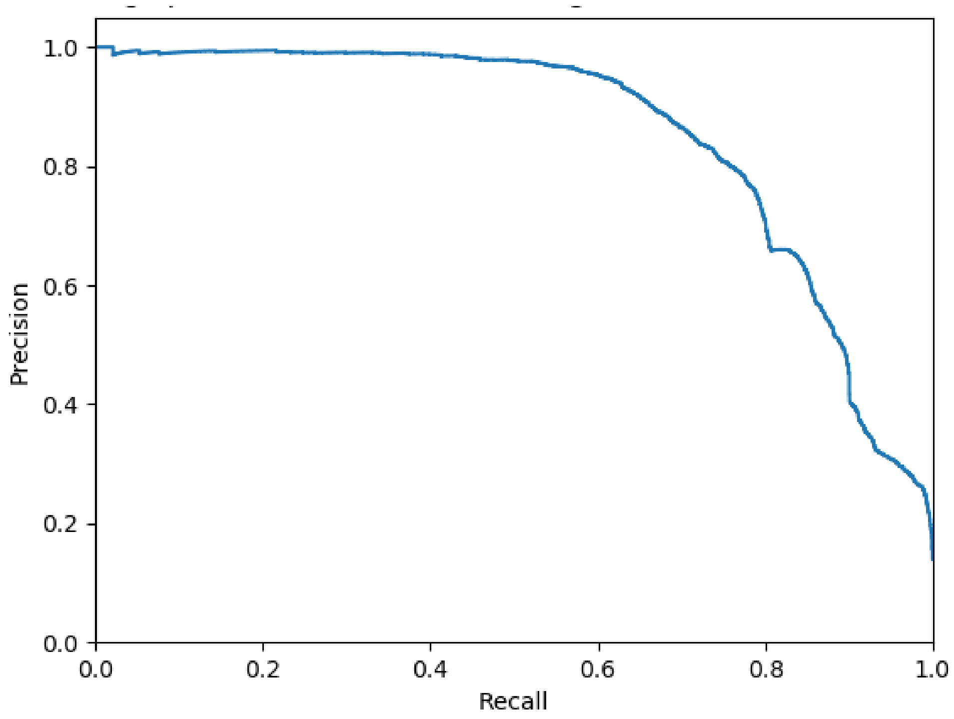


Figure 4 - Precision-recall plot of the proposed DNN-based classifier.

It shows the trade-off between precision, a measure of result relevancy, and recall, a measure of how many relevant results are returned.

A large area under the curve indicates high recall and corresponding precision values.

The average **precision score** of the **proposed DNN-based classifier**, micro-averaged over all the action classes, is 0.85.

Results (cont.)

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

where *TP* =True Positives, *TN* =True Negatives, *FP* = False Positives, and *FN* = False Negatives.

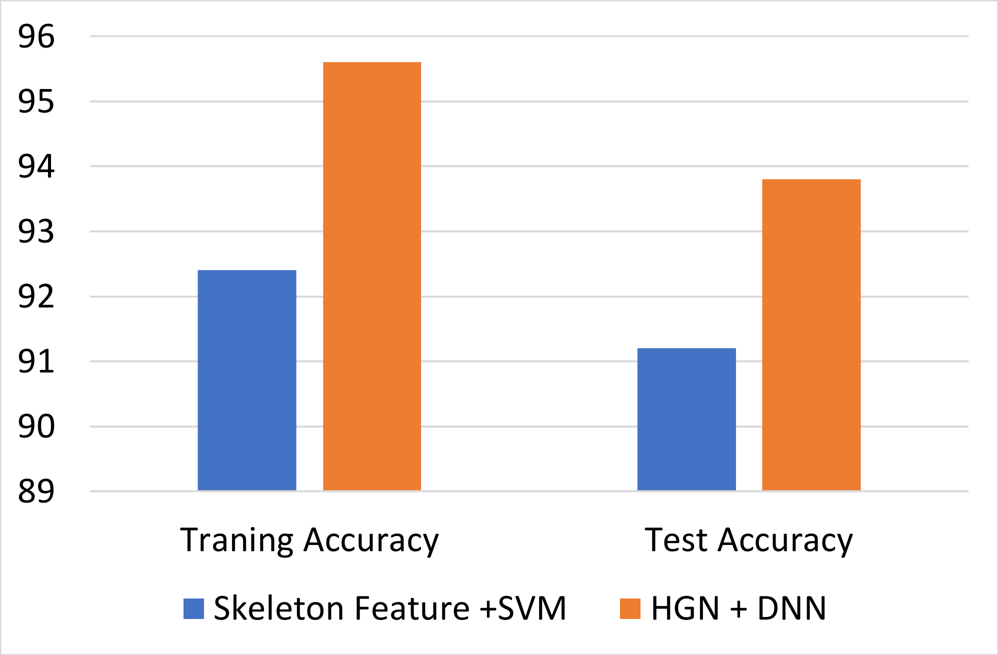


Figure 5: Training and test accuracy of SVM and DNN

Table 3: Comparison of methods based on accuracy

S. No.	Method	Accuracy %
1	Skeleton Feature + SVM	92.4
2	HGN+DNN	95.6

Results (cont.)

The classification model is trained on **two types** of processed skeleton data.

In the first type,

- the data from each frame of the video is processed separately and
- the skeleton data is used to extract and generate the feature vector on which the classifiers are trained.

In the second type,

- five frames are taken as a sliding window and the skeleton data obtained from these are concatenated and
- used to extract the features and generate the vector.

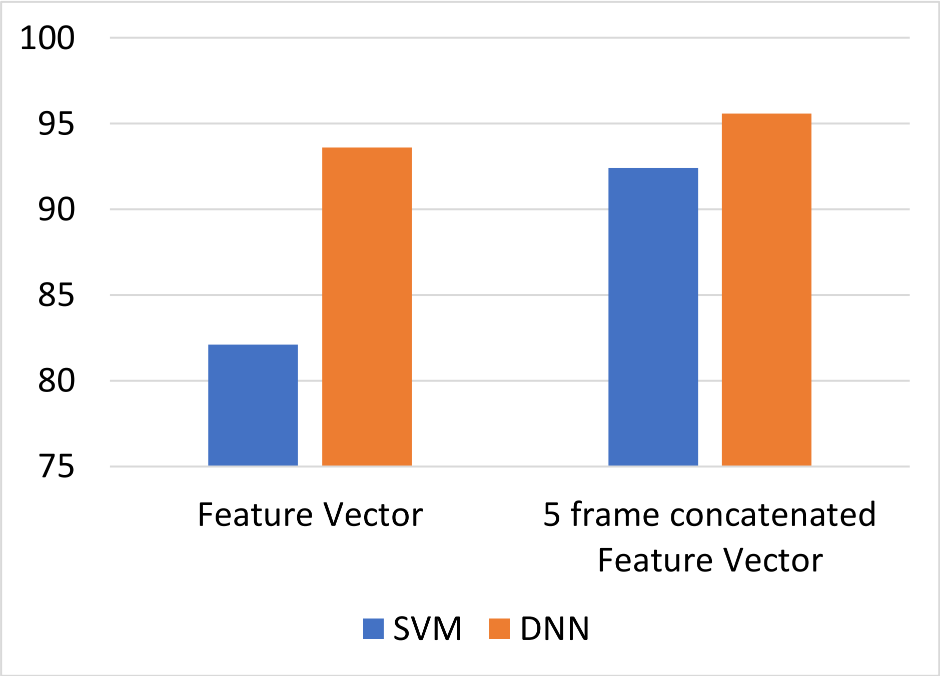


Figure 6: Accuracy of different types of Feature vector

When these concatenated frames are used, it improves the accuracy of both the SVM and DNN model.

Summary

Conclusion

- A combination of two models HGN and DNN to capture the action performed by the human subject and to recognize the action.
- The proposed system achieved an accuracy of 95.6% in action recognition on two different standard data sets of MSR Action and NTU RGB+D 3D skeleton.
- It meets the requirements of service description for video surveillance specified in Recommendation ITU-T F.743.

Future Work

- Standardization as an extension of the intelligent visual surveillance system architecture specified in Recommendation ITU-T H.626.5.

ITU KALEIDOSCOPE
ONLINE2020

Thank you!