COMPUTER VISION

PROJECT EVALUATION REPORT

View Adaptive Neural Networks for High Performance Skeleton-based Human Action Recognition

Team No: 15.

AIM:

To design View Adaptive SubNetwork that determines most suitable viewpoints for skeleton data and transforms the skeleton data to the virtual viewpoint to obtain efficient recognition of the action.

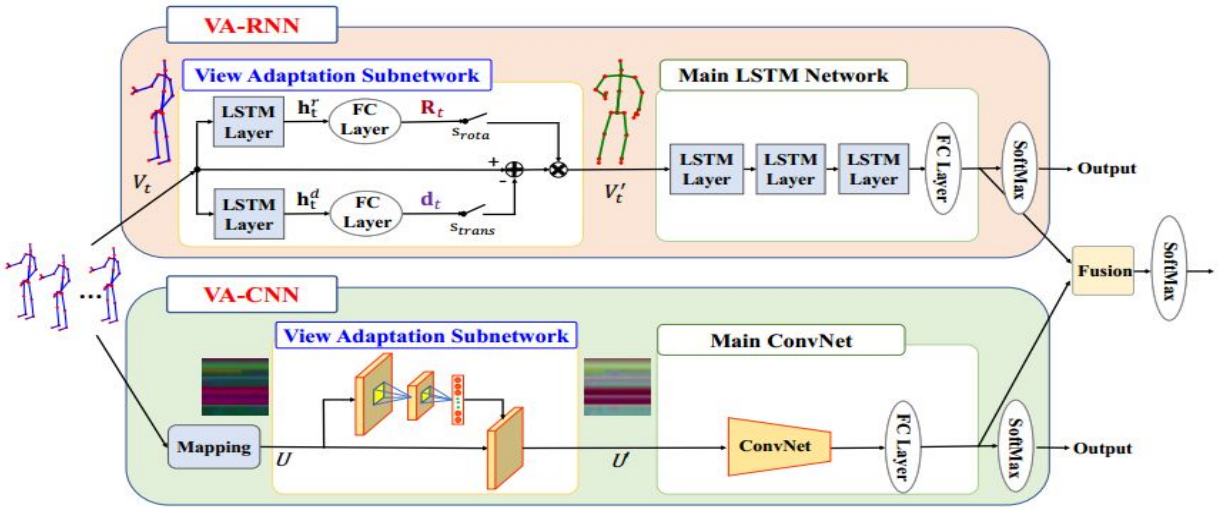
DATASET USED:

- We used NTU Dataset which is currently the largest dataset with skeleton data for human action recognition, with 56880 video samples. It contains 60 different action classes.
- Each subject has 25 joints.

There are two standard evaluations :

- Cross-subject (CS) where the 40 subjects are split into training and testing.
- Cross-view(CV) where the samples of camera 2 and 3 are used for training and those of camera 1 for testing.

ARCHITECTURE:



Algorithm:

- We first map the skeleton to image using the formula

$$\mathbf{u_{t,j}} = floor(255 \times \frac{\mathbf{V_{t,j}} - \mathbf{C_{min}}}{c_{max} - c_{min}})$$

  Where v(t,j) is the coordinates of jth joint in t'th frame. Cmax and Cmin are the maximum and minimum of all the joint coordinates in the training dataset respectively.

- Using the VA CNN subnet we regress the rotational and translation parameters required to find the new observation view point.
- Using the found parameters, we can compute the skeleton representation of jth joint in t'th frame under the new observation viewpoint using:

$$\mathbf{u'_{t,j}} = 255 \times \frac{\mathbf{v'}_{t,j} - \mathbf{c_{min}}}{c_{max} - c_{min}}$$
$$= \mathbf{R}_{t,j}\mathbf{u}_{t,j} + 255 \times \frac{\mathbf{R}_{t,j}(\mathbf{c}_{min} - \mathbf{d}_{t,j}) - \mathbf{c}_{min}}{c_{max} - c_{min}}.$$

Where

$$\mathbf{v}'_{t,j} = [x'_{t,j}, y'_{t,j}, z'_{t,j}]^{\mathrm{T}} = \mathbf{R}_t(\mathbf{v}_{t,j} - \mathbf{d}_t).$$

$$\mathbf{R}_t = \mathbf{R}^x_{t,\alpha}\mathbf{R}^y_{t,\beta}\mathbf{R}^z_{t,\gamma}$$

$$\mathbf{R}^x_{t,\alpha} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha_t) & \sin(\alpha_t) \\ 0 & -\sin(\alpha_t) & \cos(\alpha_t) \end{bmatrix},$$
$$\mathbf{R}^y_{t,\beta} = \begin{bmatrix} \cos(\beta_t) & \sin(\beta_t) & 0 \\ -\sin(\beta_t) & \cos(\beta_t) & 0 \\ 0 & 0 & 1 \end{bmatrix},$$
$$\mathbf{R}^z_{t,\gamma} = \begin{bmatrix} \cos(\gamma_t) & 0 & -\sin(\gamma_t) \\ 0 & 1 & 0 \\ \sin(\gamma_t) & 0 & \cos(\gamma_t) \end{bmatrix}.$$

- We send this to main Classification Network (Resnet here)
- This is End-to-End trained with the view determination optimized by minimizing the classification loss (Cross Entropy is used).

Implementation Details:

- From .skeleton files of dataset, we retrieved the data in the form of an np array.

- Then converted it into (no. of videos * no. of frames * 50* 3). 50 corresponds to 2 subjects in the frame where each subject has 25 joints.
- The evaluation we considered is Cross - Subject (CS).
- The IDs of training subjects in this evaluation are: 1, 2, 4, 5, 8, 9, 13, 14, 15,16, 17, 18, 19, 25, 27, 28, 31, 34, 35, 38; remaining subjects are reserved for testing.
- Convert skeleton sequence to image map to facilitate spatial temporal dynamics modelling by ConvNet.
  The image map is resized to (244,244) with 3 channels.
- For VA CNN subnet we stacked two convolutional layers with 128 kernels and each filter of size 5 used with a stride length of 2.
- Each Convolutional layer is followed by a batch Normalization layer and a ReLu layer.
- We used a max pool layer to reduce the resolution after the 2nd conv. Layer.
- Finally a Fully Connected layer is used to regress the rotational and translational parameters pertaining to the sequence. The initial weights are set to 0.
- The Main ConvNet is a Resnet installed with pre trained values.

Github Link: https://github.com/saisoorya2000/CV_Project_19