Chapter 4

Human Behaviour Traits aware Action Recognition

Human actions recognition is a complicated task in real-world videos, as it often further requires the understanding of human with involved emotion and gesture cues. For instance, identifying whether two persons are "talking" or "quarreling" with each other can only be distinguished through emotions, though actions may look similar in terms of gesture. Similarly, two different actions, like "chopping" and "frying" in the kitchen, may have similar emotions but have different gesture. However, ambiguous actions, like "crying" and "happy tears" are indistinguishable, even focusing on gestures and emotion. These actions are recognized only through long-term temporal context. Thus, identifying complex human actions is still in its rudimentary phase to the best of our knowledge.

Complex and ambiguous actions that are shown in Figure 4.1 have not been well addressed in the literature. This motivated us to propose an effective framework, known as *Human Action Attention Network (HAANet)* for action recognition. It can effectively distinguish complex human actions in videos considering attention on emotion, gesture, and long-term temporal context. Although our HAANet is illustrated for the

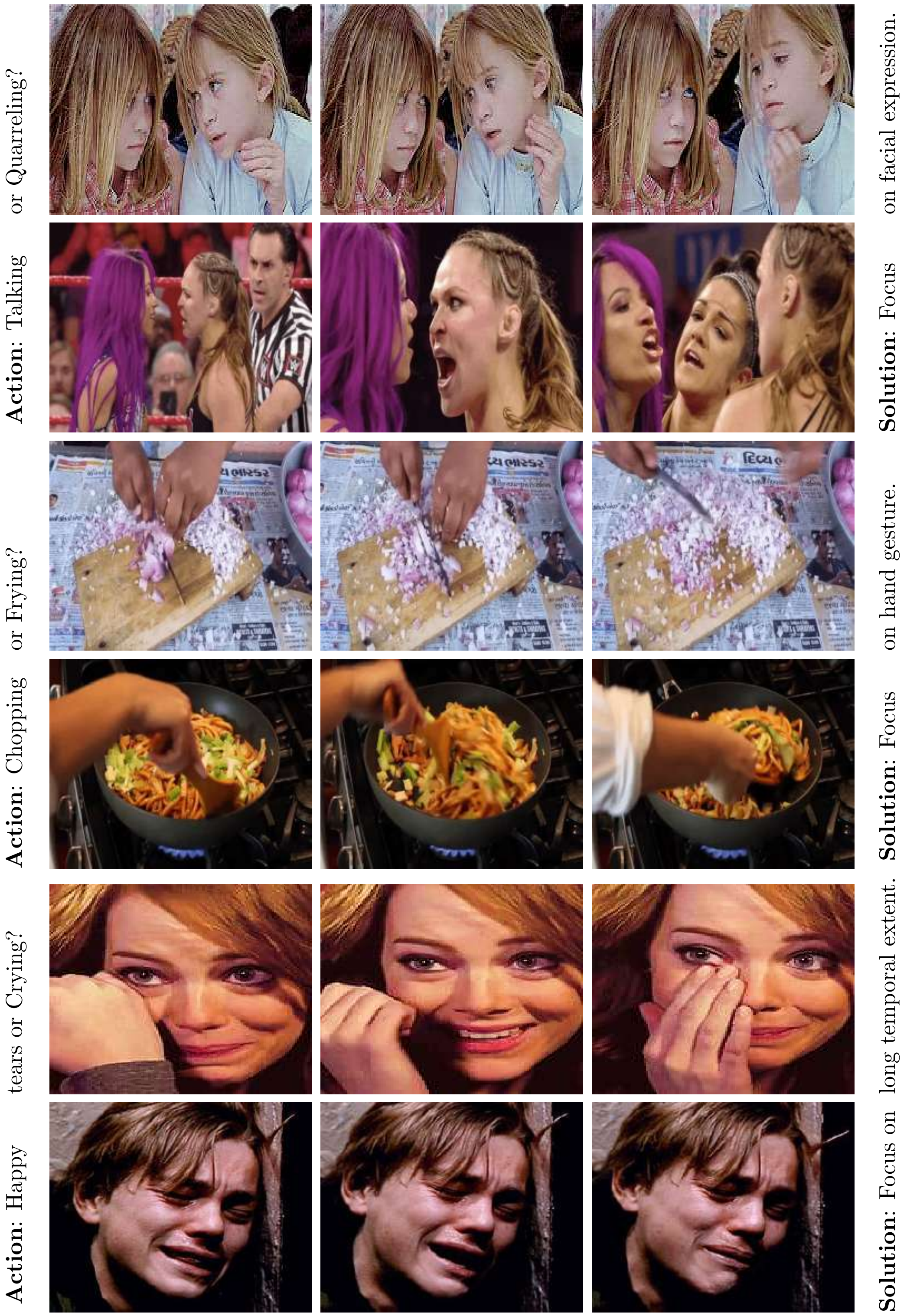

Action: Talking or Quarreling?
Solution: Focus on facial expression.
Action: Chopping or Frying?
Solution: Focus on hand gesture.
Action: Happy tears or Crying?
Solution: Focus on long temporal extent.

Figure 4.1: Video-based human actions can be understood through reasoning on multiple traits, such as ‘what is the facial expression?’, ‘what is the pattern of gesture?’, and ‘what has been happening?’. These aspects help in recognizing ambiguous actions. Video frames are from our VALC dataset.

applications of action recognition, it may also be extended to other applications under domain of computer vision, such as action localization [155, 156] and salient visual detection [157, 158].

4.1 Our Contribution

The major contributions of our work are briefly abstracted as follows:

1. We present a joint trainable multi-attention network to capture visual saliency along with long temporal context for action recognition in videos. Our network addresses ambiguous action classes which can distinguish only by learning semantic information in terms of facial expressions and gestures. We also capture motion cues over long-duration videos to exploit the temporal correlations.
2. We design a new shallow backbone network (SBN) that integrates 3D convolution layers with lateral connections to merge low, mid, and high-level spatio-temporal feature tensors. Our backbone network acquires the short-term finer details and high-level semantics to output spatio-temporal features. It addresses the degradation problem and requires an optimal number of training parameters.
3. We incorporate a visual attention network (VAN) to extract visual saliency features from salient regions related to emotion and gesture in a video. It helps to add clarity in identifying complex and ambiguous actions in a video and enhance the performance of recognition. Feature tensor obtained from backbone network is supervised using learned weights in RAVDESS and IsoGD datasets to produce the attention scores.
4. We devise a long-term attention network (LAN) consisting of spatial and temporal attention module to capture action features in long-duration videos. LAN identifies salient spatio-temporal contextual cues using attention mechanism in ConvLSTM. Spatial and temporal attention are designed to assign an attentive score to each pixel location and different frames, respectively. We reformulate

ConvLSTM to mitigate the ambiguity of recognizing classes, which can be classified precisely only through long-term temporal context.

5. We propose a temporal attention pooling (TAP) that incorporates class-aware attentional pooling to capture discriminative semantic features. The class-aware attentional pooling is a trainable layer that extracts a class-aware feature vector over the time. It classifies the video segments into the related action class based on generated spatio-temporal feature tensor. In addition, we estimate a regularized objective function in the joint training model for efficiently training overall architecture.
6. We perform an extensive set of experiments on five benchmark datasets to manifest efficiency of HAANet. An abundant ablation studies is performed to highlight the impact of different modules of our network. As per knowledge, no dataset exist in the literature that contains ambiguous actions pairs that can only be classified based on important visual saliency cues and long-term temporal context. We therefore create a new dataset, named as *Visual Attention with Long-term Context (VALC)*, that contains 32 complex and ambiguous action classes with about 100 videos per class.

4.2 Organization of the Chapter

The subsequent section summarizes the SOTA literature for action recognition in videos and visual cues. In Section 4.4, we embellish our HAANet. We examine the results obtain from different experiments and ablation studies in Section 4.5. Lastly, we conclude the chapter and related publication in Sections 4.6 and 4.7, respectively.

4.3 Literature Survey

In [159], authors have proposed hardware-efficient model where a part of channels are shifted along temporal dimension. Distillation mechanism with 3D convolutions in [146] is used to recognize human actions. Authors have illustrated global diffusion network that intends to learn local and global representations in unified fashion [160]. In [137], non-local operations are used as an attention mechanism to capture long-term dependencies for video classification. In [143], the authors have introduced network that includes long-term content to nimble relationships within the actions. Timeception [121] layer is designed to learn long-term temporal dependencies and variations in temporal extents of complex actions. SlowFast Network [150], consists of two-path way, *i.e.*, one for low frame rate and other for higher ones. Quan *et al.* [161] have proposed LSTM network with an attention mechanism for action recognition. FactorNet [162] focus attention to separate activity of person performing action from the relevant objects and co-occurring background bias.

In [163], authors introduce global and local knowledge-aware attention network to recognize action classes. R(2+1)D BERT [164] is combination of late temporal modeling and 3D convolutions to recognize action in video. Authors in [165, 166] have proposed a vision transformer that achieves SOTA without any convolution layers for image classification and action recognition tasks. In [167], single-stage continuous gesture recognition framework is proposed for detecting and classifying multiple gestures in a video. In [168–170], the authors perform gesture recognition tasks using machine learning algorithms.

In [171–173] 3D-CNNs are adopted to learn appearance and motion features in gesture videos. Some works are related to video-based emotion recognition in [174,175]. These works have evaluated deep features extracted from different CNNs, including AlexNet, VGGNet and GoogleNet for emotion recognition in videos. In [176] authors have improved representation capacity of an architecture and reduced confusion between pairs of ambiguous action classes using discriminative filter bank. Nevertheless, to the best of our knowledge, there is scarcely any feature fusion of human traits methods developed for improving recognition of actions in videos.

Many challenging public datasets were proposed, such as UCF101 [1], HMDB51 [2], and Kinentics400 [133] that focused on capturing a broad diversity in terms of the single unambiguous classes of human activities and their motion patterns. This motivates us to create a novel video-based action dataset containing only ambiguous actions.

4.4 Proposed Approach

We propose a deep neural network, known as *HAANet*, for action recognition in videos. Figure 4.2 depicts block diagram of the overall network. Firstly, we design a backbone network for capturing short-term video spatio-temporal features. Secondly, we introduce a visual attention network that focuses on selective regions of video to capture visual saliency features related to facial expression and gesture. Thirdly, we propose a long-term attention network to learn contextual features over long-term temporal dependencies of actions in video. Finally, reformulate the optimized objective function for joint training that can train the overall architecture.

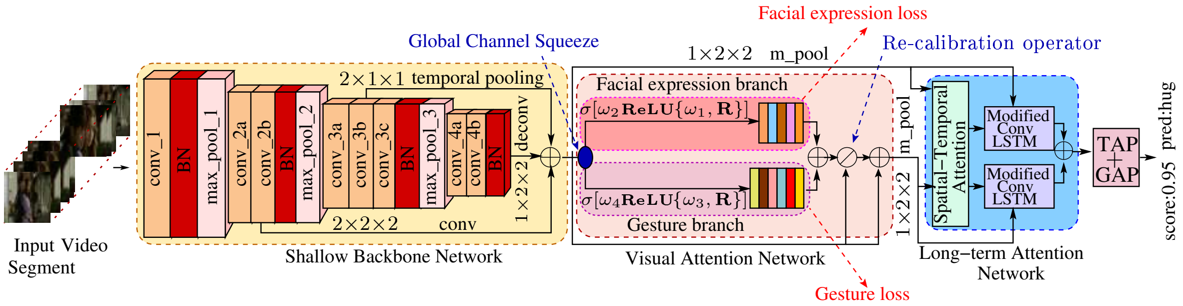


Figure 4.2: Overall architecture of our HAANet. "deconv", "conv", and "m_pool" depict deconvolution, convolution, and max-pooling operations, respectively.

Table 4.1: Architecture of proposed SBN. ↑ and ↓ indicate up-sampling and down-sampling, whereas $\mathbb{S}$ and $\mathbb{T}$ mean spatially and temporally, respectively.

Layers	**Backbone Network**		**LCs**
	Blocks	**Output Size**	**[Out]**
conv_1	$[3 \times 3 \times 3, 64]$	$T \times 112 \times 112 \times 64$	-
max_pool_1	$1 \times 2 \times 2$	$T \times 56 \times 56 \times 64$	-
conv_2(a,b)	$[3 \times 3 \times 3, 128] \times 2$	$T \times 56 \times 56 \times 128$	conv2b ($2\times \downarrow \mathbb{ST}$)
max_pool_2	$1 \times 2 \times 2$	$T \times 28 \times 28 \times 128$	-
conv_3(a,b,c)	$[3 \times 3 \times 3, 256] \times 3$	$T \times 28 \times 28 \times 256$	conv3b ($2\times \downarrow \mathbb{T}$)
max_pool_3	$2 \times 2 \times 2$	$T/2 \times 14 \times 14 \times 256$	-
conv_4(a,b)	$[3 \times 3 \times 3, 512] \times 2$	$T/2 \times 14 \times 14 \times 512$	conv4b ($2\times \uparrow \mathbb{S}$)
output tensor $\mathbf{M}$ of size $T/2 \times 28 \times 28 \times \zeta$ (all LCs are concatenated)			

4.4.1 Backbone Network

Video-based human actions are spatio-temporal signals comprising of visual appearance that dynamically progress over time. To differentiate several action categories, 3D convolutions learn spatio-temporal kernels over spatial, temporal, and semantic channel subspace. 3D kernel $\mathsf{W} \in \mathbb{R}^{t \times x \times y}$ with ζ filters learns spatio-temporal patterns by convolving with an input RGB video segment $\mathbf{V} \in \mathbb{R}^{\dagger' \times \mathcal{X}' \times \mathcal{Y}' \times \mathcal{C}'}$ as follows: $\mathbf{M} = \mathbf{V} * \mathsf{W}$, where $*$ denotes a 3D convolution operator and $\mathbf{M} \in \mathbb{R}^{\dagger \times \mathcal{X} \times \mathcal{Y} \times \mathcal{C}}$ is resultant spatio-temporal feature tensor. $\{\mathcal{X}', \mathcal{Y}'\}$ and $\{\mathcal{X}, \mathcal{Y}\}$ are spatial resolution of input and output tensors whereas $\dagger'$ and $\dagger$ are input and output temporal length, respectively. An input video segment has RGB channels. Inspired by [12], we have implemented a shallow backbone network, named as *SBN* that contains eight 3D convolutional layers (conv_1−conv_4b) and three max pooling layers (max_pool_1−max_pool_3). The main design criteria of SBN is to reduce the number of parameters as compared to other backbone networks such as C3D, 3D ResNet, 3D ResNet34, 3D ResNet101. According to experimental observation, parameters in C3D, 3D ResNet, 3D ResNet34, 3D ResNet101 are too high as compare to SBN, shown in Table 4.5. SBN takes as input a group of RGB frames, denoted by segment. The convolution layers are used to extract complex features of these segments to improve recognition accuracy. In backbone network, pooling layers gradually pools reduced features to have fewer parameters and computations in network. These layers are utilized after first, third, and sixth convolutional layers. Size of kernel of the first and second maxpool layers is $1 \times 2 \times 2$, while third layer is of size $2 \times 2 \times 2$. Purpose to merge features of the temporal domain later is to conserve long-term temporal span. The early convolution layer conv_2a,b of the backbone network detects low-level features, such as moving edges, blobs or corners. The middle convolution layer conv_3a,b,c of network detects mid-level features. The deepest layer conv_4a,b learns complex moving patterns, such as moving circular objects, face related motion, and biking-like motion. Low-level and mid-level spatio-temporal features are essential for capturing the finer details of human actions. Therefore, SBN also learns spatio-temporal features inherited from lower, middle, and higher layers to obtain the output tensor $\mathbf{M}$ using lateral connections (LCs). The output feature tensor of the lower layer (conv_2b) is 2× downsampled by strided convolution and higher layer (conv_4b) is 2× upsampled through deconvolution. Mid-level feature tensor extracted from conv_3b is temporally downsampled using temporal pooling with the kernel of size $2 \times 1 \times 1$ and merged with the other two tensors of same size $T/2 \times 28 \times 28 \times \zeta$. Hence, SBN preserves low-level, mid-level, and high-level spatio-temporal information. The output feature of the proposed shallow network is $\mathbf{M} \in \mathbb{R}^{T/2 \times 28 \times 28 \times \zeta}$. 3D-batch normalization [126] is stacked immediately over each convolution layer to improve the performance and optimize our backbone network. The backbone network architecture is ablated in Table 4.1. SBN learns spatio-temporal features from a given video segment. However, it is incapable of emphasizing action-specific semantic information of a video segment. Hence, to capture visual saliency information from a video segment, we have incorporated a visual attention network.

4.4.2 Visual Attention Network (VAN)

We humans consider various spatial features, like textures, color, and shape, along with our cognitive visual mechanism that provides selective attention to relevant regions for correctly recognizing actions in a video segment. The selectively attended regions could correspond to either interesting activities or prominent objects in a video segment that are most attractive to viewers. Visual attention mechanism significantly narrows the search term by giving the hierarchical priority within a target video segment to perform further activity analysis on these regions. The visual attention mechanism is incorporated in HAANet to extract relevant information from various parts of the video segments.

In this section, we incorporate a visual attention network, denoted by VAN, to focus on action-specific visual saliency information, such as, gesture and facial expression. The main aim of VAN is emphasizing action-specific semantic information of a video segment to recognize human action more precisely. For instance, the ambiguous actions, like “chopping“ can be distinguished from “slicing“ in terms of gesture. These two action videos must be classified into two different classes. Gesture are different, but have similar facial expression. Our visual attention network takes feature tensor $\mathbf{M}$ of backbone network as an input. This tensor is initially supervised separately in two parallel branches, *i.e.,* facial expression and gesture with learned weights, as shown in Figure 4.2. First, we have squeezed channel information of $\mathbf{M}$ globally with respect to spatial resolution to reduce the number of parameters. Formally, $\mathbf{R}$ is computed by shrinking $\mathbf{M}$ as follows:

$$\mathbf{R} = \frac{1}{\zeta} \sum_{j \in \zeta} \mathbf{m}_{i,j,k}, \tag{4.1}$$

where $\mathbf{m}_{i,j,k}$ is an element of $\mathbf{M}$. $\mathcal{T} = T/2$ and $H \times W$ is spatial resolution with ζ channels. Next, we have formulated facial expression and gesture supervised attention maps $\mathbf{P}^{epr}$ and $\mathbf{P}^{gst}$, which are given by:

$$[\mathbf{P}^{epr},\ \mathbf{P}^{gst}] = [\frac{\widehat{\mathbf{M}^{epr}}}{\Phi(\widehat{\mathbf{M}^{epr}})}, \frac{\widehat{\mathbf{M}^{gst}}}{\Phi(\widehat{\mathbf{M}^{gst}})}], \tag{4.2}$$

$$[\widehat{\mathbf{M}^{epr}}, \widehat{\mathbf{M}^{gst}}] = [\sigma(\omega_2 \mathbf{ReLU}\{\omega_1 \mathbf{R}\}), \sigma(\omega_4 \mathbf{ReLU}\{\omega_3 \mathbf{R}\})], \tag{4.3}$$

where $\{\sigma(\cdot), \Phi(\cdot)\}$ are sigmoid and standard deviation functions. The sigmoid function is used as a gating mechanism and the standard deviation function is used for normalization. $\{\omega_1,\ \omega_2, \omega_3, \omega_4\}$ are the weights for facial expression (epr) and gesture (gst) branches, respectively. **ReLU** represents a ReLU activation function [177], which is applied to the feature tensor $\mathbf{M}$ for generating output with positive region predictions in both cases. $\mathbf{P}^{epr}$ represents the activation maps with respect to facial expression, whereas $\mathbf{P}^{gst}$ has important score values of gesture. The output tensors for expression and gesture branches are aggregated, reshaped, and then recalibrated with the original feature tensor $\mathbf{M}$ through element-wise product to obtain $\mathfrak{R} \in \mathbb{R}^{T/2 \times 28 \times 28 \times \zeta}$, which is computed by:

$$\mathbf{l}_i = \mathsf{W}_i^{epr} \times \mathbf{p}_i^{epr} +\ \mathsf{W}_i^{gst} \times \mathbf{p}_i^{gst}, \tag{4.4}$$

$$\mathfrak{R} = [\mathbf{l}_1\, \mathbf{m}_1, \cdots, \mathbf{l}_{\mathcal{T} \times H \times W}\, \mathbf{m}_{\mathcal{T} \times H \times W}], \tag{4.5}$$

where $\{\mathbf{l}_i, \mathbf{m}_i\}$ are $i-th$ elements of L and $\mathbf{M}$, respectively. L is the feature tensor obtain by element-wise addition of facial expression and gesture branches. $\{\mathsf{W}^{epr}, \mathsf{W}^{gst}\}$ are the learnable parameters. The output of VAN module is the feature tensor $\boldsymbol{\mathcal{A}}$, which is obtained by element-wise addition of tensors $\mathfrak{R}$ and $\mathbf{M}$. SBN and VAN focus on spatio-temporal features with detailed visual understanding of the actions but still incapable of capturing long temporal context which lasts for several seconds. The cues of evaluating actions in long-duration helps in obtaining "what is happening in the present" and "what's gonna happen next" based on "what has happened earlier. In such a scenario, sophisticated mechanism like LSTM can be introduced.

4.4.3 Long-term Attention Network (LAN)

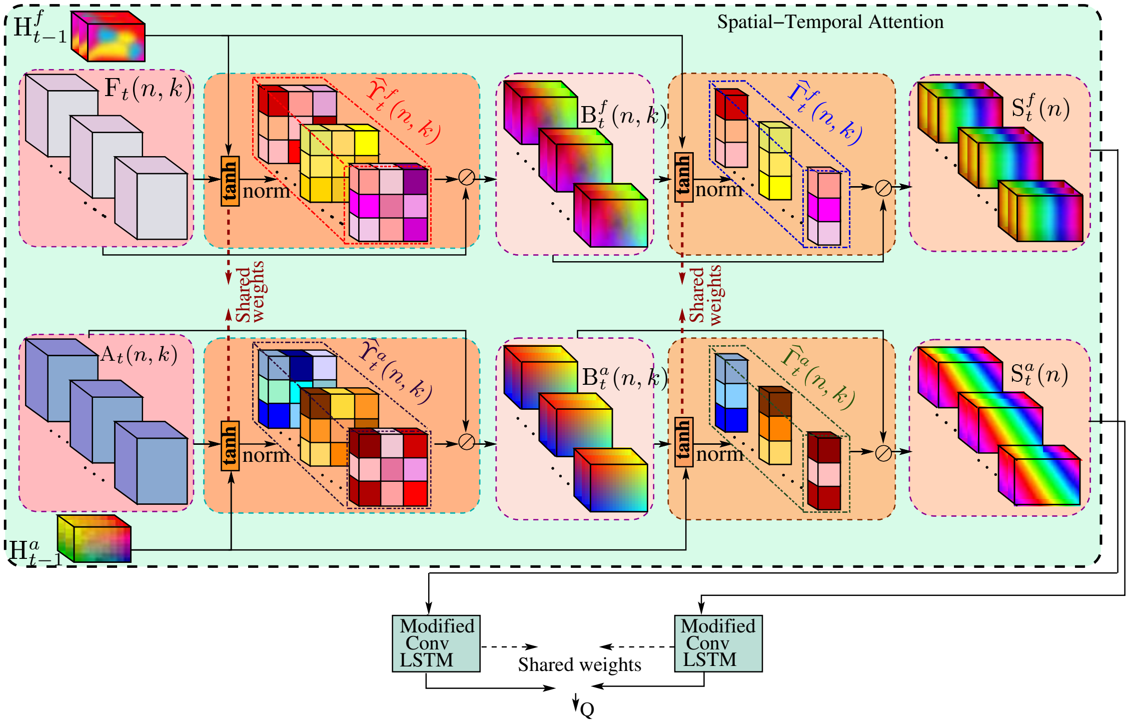


Figure 4.3: Architecture of Long-Range Attention Network (LAN).

The importance of short- to long-term temporal dynamics varies in classifying different actions. Actions like talk or quarrel can be recognized by short-term temporal dependencies in only a few frames. However, other actions like happy tear or cry, long-term temporal dependencies provide more discriminative cues for classification, as these actions last long in time. Nevertheless, such variations of temporal patterns have not been fully exploited in most existing methods. Therefore, we need to capture long-term temporal dynamics for improving the performance of recognizing actions accurately. In order to preserve long-term temporal and spatial dependencies of video-based actions, we have utilized ConvLSTMs in our network. ConvLSTM [130] has input tensor $\mathbf{X}_t$, future state of a cell $\mathbf{C}_t$, and hidden state $\mathbf{H}_{t-1}$ at time t and $t-1$, respectively. ConvLSTM is a variant of LSTM which learns the spatio-temporal features due to its inherent convolutional structure [130]. The hidden states, the inputs, and the cell outputs are

$\{\mathbf{H}_1, \cdots, \mathbf{H}_t\}$, $\{\mathbf{X}_1, \cdots, \mathbf{X}_t\}$, and $\{\mathbf{C}_1, \cdots, \mathbf{C}_t\}$, respectively. It also contains the forget gate $\mathbf{f}_t$, output gate $\mathbf{o}_t$, input gate $\mathbf{i}_t$, and memory cells $\mathbf{g}_t$. It has state-to-state and input-to-state transitions to extract correlation information of spatio-temporal features. The spatial and temporal resolution of output feature tensors $\mathbf{M}$ and $\boldsymbol{\mathcal{A}}$ are down-sampled by using pooling with a kernel of size $1 \times 2 \times 2$ to obtain $\mathbf{F} \in \mathbb{R}^{\mathcal{T} \times 14 \times 14 \times \zeta}$ and $\mathbf{A} \in \mathbb{R}^{\mathcal{T} \times 14 \times 14 \times \zeta}$, respectively, as shown in Fig 4.3. The inputs of ConvLSTM is reformulated to compute the discriminative features over several video segments. Feature tensor $\mathbf{FA}_t^{\phi}$ is utilize to encode the spatial and temporal cues of video segments, where ϕ represents either $\mathbf{F}_t$ or $\mathbf{A}_t$ for t-th video segment. This helps to capture underlying background context cues with respect to space and time. The input $\mathbf{FA}_t^{\phi}$ at each timestep t is given by:

$$\mathbf{FA}_t^{\phi} = \{\mathbf{FA}_t^{\phi}(1), \mathbf{FA}_t^{\phi}(2), \cdots, \mathbf{FA}_t^{\phi}(\mathcal{T} \times X \times Y)\}. \tag{4.6}$$

The feature tensor $\mathbf{FA}_t^{\phi}(k) \in \mathbb{R}^{\zeta}$ is a set of feature tensors at k-th spatial locations, where $k = \{1, 2, \cdots, \mathcal{T} \times X \times Y\}$. We introduce a novel attention mechanism to identify important long-term spatial and temporal contextual information. An attention is given to spatial features for capturing key regions followed by computation of the segment-score of each spatial feature to project attention on temporal features. $\mathbf{FA}_t^{\phi}(n, k)$ is a feature tensor at k-th location of n-th feature tensor which is used to estimate essential regions spatially. The current input $\mathbf{FA}_t^{\phi}(n, k)$ and previous hidden state $\mathbf{H}_{t-1}^{\phi}$ are convolved with kernel W_{fa}^{ϕ} and W_{h}^{ϕ}, respectively, to generate tensor with spatial attention. The salient spatial locations in the feature tensor $\mathbf{FA}_t^{\phi}(n, k)$ at t-th time step is calculated by:

$$\boldsymbol{\Upsilon}_t^{\phi}(n, k) = \mathsf{W}_{\Upsilon}^{\phi} * \mathbf{tanh}(\mathsf{W}_{h}^{\phi} * \mathbf{H}_{t-1}^{\phi} + \mathsf{W}_{fa}^{\phi} * \mathbf{FA}_{t-1}^{\phi}(n, k)), \tag{4.7}$$

where $\{\mathsf{W}_{\Upsilon}^{\phi}, \mathsf{W}_{h}^{\phi}, \mathsf{W}_{fa}^{\phi}\}$ are spatial attention parameters. $\boldsymbol{\Upsilon}_t^{\phi}(n, k)$ is un-normalized spatial location score of $\mathbf{FA}_t^{\phi}(n,k)$. The normalized $\widehat{\mathbf{\Upsilon}}_t^{\phi}(n,k)$ is computed by:

$$\widehat{\mathbf{\Upsilon}}_t^{\phi}(n,k) = \frac{\mathbf{exp}\{\mathbf{\Upsilon}_t^{\phi}(n,k)\}^{\lambda_{\widehat{\mathbf{\Upsilon}}}}}{\sum_{j=1}^{K^2} \mathbf{exp}\{\mathbf{\Upsilon}_t^{\phi}(n,j)\}^{\lambda_{\widehat{\mathbf{\Upsilon}}}}}, \tag{4.8}$$

where $\lambda_{\widehat{\mathbf{\Upsilon}}}$ is a parameter to control the sharpness of location-score map. The spatial attention score $\mathbf{B}_t^{\phi}(n,k)$ of n-th feature tensor at k-th location is formulated by:

$$\mathbf{B}_t^{\phi}(n,k) = \sum_{k=1}^{K^2} \widehat{\mathbf{\Upsilon}}_t^{\phi}(n,k) \circ \mathbf{FA}_t^{\phi}(n,k), \tag{4.9}$$

where $\circ$ is Hadamard product and K^2 is spatial resolution. The aforesaid analysis elaborates that the motion of a human in a video can be observed through spatial relationships within the successive segments of video. An attention method is further applied over the temporal features for selectively tracking the actions in video segment. The spatial tensor $\mathbf{B}_t^{\phi}(n,k)$ is given as input to temporal attention mechanism for estimating clip score $\Gamma_t^{\phi}(n)$. The n-th temporal tensor $\Gamma_t^{\phi}(n)$ at t-th time step can be formulated by:

$$\mathbf{\Gamma}_t^{\phi}(n) = \mathsf{W}_{\Gamma}^{\phi} * \mathbf{tanh}(\mathsf{W}_h^{\phi} * \mathbf{H}_{t-1}^{\phi} + \mathsf{W}_b^{\phi} * \mathbf{B}_{t-1}^{\phi}(n,k)), \tag{4.10}$$

where temporal attention parameters are $\{\mathsf{W}_{\Gamma}^{\phi}, \mathsf{W}_h^{\phi}, \mathsf{W}_b^{\phi}\}$ and $\mathbf{B}_t^{\phi}(n)$ is the n-th spatial attention feature tensor. We normalize $\mathbf{\Gamma}_t^{\phi}(n)$ to obtain the temporal score $\widehat{\mathbf{\Gamma}}_t^{\phi}(n)$, which is calculated as follows:

$$\widehat{\mathbf{\Gamma}}_t^{\phi}(n) = \frac{\mathbf{exp}\{\mathbf{\Gamma}_t^{\phi}(n)\}^{\lambda_{\widehat{\mathbf{\Gamma}}}}}{\sum_{j=1}^{T} \mathbf{exp}\{\mathbf{\Gamma}_t^{\phi}(j)\}^{\lambda_{\widehat{\mathbf{\Gamma}}}}}, \tag{4.11}$$

where $\lambda_{\widehat{\mathbf{\Gamma}}}$ is a parameter to control sharpness for normalization. The temporal score $\widehat{\mathbf{\Gamma}}_t^{\phi}(n)$ reflects the temporal importance of n-th temporal feature tensor at t-th time step. The spatial and temporal score use $\widehat{\mathbf{\Upsilon}}_t^{\phi}(n,k)$ and $\widehat{\mathbf{\Gamma}}_t^{\phi}(n)$ as the shared weights to reduce the computations and model complexity. Further, it is summarized to obtain spatial-temporal tensor $\mathbf{S}^{\phi}_t(n)$, which is defined by:

$$\mathbf{S}^{\phi}_t(n) = \sum_{n=1}^{\mathcal{T}} \mathbf{B}^{\phi}_t(n) \circ \widehat{\mathbf{\Gamma}}^{\phi}_t(n). \tag{4.12}$$

The dimensionality of $\mathbf{S}^{\phi}_t(n)$ is same as of $\mathbf{FA}^{\phi}_t$, where ζ is 512. We therefore conclude that $\mathbf{S}^{\phi}_t$ is a spatio-temporal feature tensor that capture the salient spatial and temporal cues. Our spatial-temporal attention method captures the essential context at spatial and temporal feature tensors that help in the prediction of human action significantly. $\mathbf{X}_t$ is an input to ConvLSTM, which is the feature tensor obtained from $\mathbf{FA}^{\phi}_t$. To preserve spatial and temporal correlation with respect to time, $\mathbf{S}^{\phi}_t$ is fed as an extra input into ConvLSTM. Therefore, our variant of ConvLSTM, known as modified ConvLSTM, is reformulated as:

$$\mathbf{f_t}^{\phi} = \sigma(\mathbf{Z}^{\phi}_{xf} * \mathbf{X}^{\phi}_t + \mathbf{Z}^{\phi}_{hf} * \mathbf{H}^{\phi}_{t-\mathit{1}} + \mathbf{Z}^{\phi}_{sf} * \mathbf{S}^{\phi}_t + b^{\phi}_{f}), \tag{4.13}$$

$$\mathbf{i_t}^{\phi} = \sigma(\mathbf{Z}^{\phi}_{xi} * \mathbf{X}^{\phi}_t + \mathbf{Z}^{\phi}_{hi} * \mathbf{H}^{\phi}_{t-\mathit{1}} + \mathbf{Z}^{\phi}_{si} * \mathbf{S}^{\phi}_t + b^{\phi}_{i}), \tag{4.14}$$

$$\mathbf{o_t}^{\phi} = \sigma(\mathbf{Z}^{\phi}_{xo} * \mathbf{X}^{\phi}_t + \mathbf{Z}^{\phi}_{ho} * \mathbf{H}^{\phi}_{t-\mathit{1}} + \mathbf{Z}^{\phi}_{so} * \mathbf{S}^{\phi}_t + b^{\phi}_{o}), \tag{4.15}$$

$$\mathbf{g}^{\phi}_t = \mathbf{tanh}(\mathbf{Z}^{\phi}_{xc} * \mathbf{X}^{\phi}_t + \mathbf{Z}^{\phi}_{hc} * \mathbf{H}^{\phi}_{t-\mathit{1}} + \mathbf{Z}^{\phi}_{sg} * \mathbf{S}^{\phi}_t + b^{\phi}_{c}), \tag{4.16}$$

$$[\mathbf{C}^{\phi}_t,\ \mathbf{H}^{\phi}_t] = [\mathbf{f}^{\phi}_t \circ \mathbf{C}^{\phi}_{t-1} + \mathbf{i}^{\phi}_t \circ \mathbf{g}^{\phi}_t,\ \mathbf{o}^{\phi}_t \circ \mathbf{tanh}(\mathbf{C}^{\phi}_t)], \tag{4.17}$$

where $\mathbf{Z}^{\phi}_{\theta}$ and b^{ϕ}_{ϑ} are the shared parameters of our ConvLSTM network. The sigmoid activation, tanh function, forget, input, and output gates, memory cells, cell output state at t-th time step, the current input, previous the hidden state, spatio-temporal attention feature tensor are represented by σ, $\mathbf{tanh}$, $\{\mathbf{f}^{\phi}_t, \mathbf{i}^{\phi}_t, \mathbf{o}^{\phi}_t, \mathbf{g}^{\phi}_t\}$, $\mathbf{C}^{\phi}_t$, $\mathbf{X}^{\phi}_t$, $\mathbf{H}^{\phi}_{t-1}$, and $\mathbf{S}^{\phi}_t$, respectively. The combination of both inputs $\mathbf{X}^{\phi}_t$ and $\mathbf{S}^{\phi}_t$ allows ConvLSTM to determine discriminative actions at each time step. Finally, we obtain current hidden state in both SBN and VAN, *i.e.,* $\mathbf{H}^{f}_t$ and $\mathbf{H}^{a}_t$, respectively. The hidden states are combined using element-wise addition to get $\mathbf{Q} \in \mathbb{R}^{T/2\times 14\times 14\times \zeta}$ and is given as input in the temporal pooling layer. This will diminish the dimension of $\mathbf{Q}$.

4.4.4 Temporal Attention Pooling (TAP)

In recent literature, concept of feature pooling in temporal dimension has been widely used for reducing the time to process the long-range videos. It performs statistical operations within multiple regional windows with respect to time to capture motion information of videos. In HAANet over the time steps, the temporal pooling layer is stacked on above the LAN module, as depicted in Figure 4.2. We exploit attention pooling in temporal dimension that can capture discriminative semantic aspects. $\mathbf{Q}_t \in \mathbb{R}^{14\times 14\times \zeta}$ is a feature tensor of a video segment at time step $t \in \{1, \cdots, T/2\}$. These tensors are given as input to temporal pooling at t-th time step. Weight matrix W_t is incorporated to produce weighted feature tensors $\mathbf{\Omega}_t^{(att)}$ using temporal pooling operator.

We design a temporal attention pooling model that can capture class-aware spatio-temporal discriminative features. A video has the ability to distinguish complex human actions. It extracts important semantic information in t-th time step by computing attentive scores on tensor $\mathbf{Q}_t$. The obtained attention tensor $\mathbf{\Omega}_t^{(att)}(k)$ of k-th class is formulated by:

$$\mathbf{\Omega}_t^{(att)}(k) = [\mathbf{1}^\mathsf{T}\mathbf{E}_k]^\mathsf{T}, \tag{4.18}$$

$$\text{where } \mathbf{E}_k = \mathbf{N}_k \circ \mathsf{W}^\mathsf{T}, \tag{4.19}$$

$$\mathbf{N}_k = \mathbf{Q}_t\mathbf{ß}_k, \tag{4.20}$$

$$\mathsf{W} = \mathbf{Q}_t\mathbf{\Psi}, \tag{4.21}$$

where k indicates action classes. $\mathbf{1} \in \mathbb{R}^{\mathcal{T}\times\eta}$ is a tensor of all ones, where $\eta = 14 \times 14 \times \zeta$ and $\{\mathbf{ß}_k, \mathbf{\Psi}\} \in \mathbb{R}^{\eta\times 1}$ are class-specific attention features vectors. The class-specific tensor is extracted using a fully-connected layer followed by sigmoid activation. In most deep neural networks, vectorized feature maps are first fed to dense layers and then to a softmax layer. Since the dense layers are susceptible to overfitting, we thus use the global average pooling (GAP) layer to reduce overfitting and the number of model parameters.

4.4.5 Joint Training Mechanism

HAANet consist of SBN, VAN, and LAN for action recognition in a video. It is difficult to optimize the overall architecture. Therefore, we introduce a joint training mechanism to train overall architecture.

1) Computation of VAN learned weights: In case of VAN, we have used the method of pseudo-label to estimate the true labels from the prediction of classifier. We have used the pseudo labels as a ground-truth of facial expression and gesture, which are unavailable with action classes. The loss function for computing pseudo-labels with the given set of unlabeled instances $\mathcal{Z}'$ and labeled instances $\mathcal{Z}$ are formulated as:

$$\pounds_{epr} = \frac{1}{\mathcal{Z}} \sum_{i \in \mathcal{Z}} \sum_{f_j \in F} \pounds(\mathbf{p}^i_{f_j}, \mathbf{q}^i_{f_j}) + \mu_1 \frac{1}{\mathcal{Z}'} \sum_{i \in \mathcal{Z}'} \sum_{f_j \in F} \pounds(\hat{\mathbf{p}}^i_{f_j}, \hat{\mathbf{q}}^i_{f_j}), \tag{4.22}$$

$$\pounds_{gst} = \frac{1}{\mathcal{Z}} \sum_{i \in \mathcal{Z}} \sum_{g_j \in G} \pounds(\mathbf{p}^i_{g_j}, \mathbf{q}^i_{g_j}) + \mu_2 \frac{1}{\mathcal{Z}'} \sum_{i \in \mathcal{Z}'} \sum_{g_j \in G} \pounds(\hat{\mathbf{p}}^i_{g_j}, \hat{\mathbf{q}}^i_{g_j}), \tag{4.23}$$

where $\{\mu_1, \mu_2\}$ are weights to control the contribution of unlabeled instances to the overall loss and value are computed similar as [178]. $\{\mathbf{p}^i_{f_j}, \mathbf{p}^i_{g_j}\}$ are the predicted values of $\mathcal{Z}$ instances and $\{\mathbf{q}^i_{f_j}, \mathbf{q}^i_{g_j}\}$ are labels of facial expression and gesture in labeled data. The predicted values of $\mathcal{Z}'$ instances are $\{\hat{\mathbf{p}}^i_{f_j}, \hat{\mathbf{p}}^i_{g_j}\}$ and pseudo-label are $\{\hat{\mathbf{q}}^i_{f_j}, \hat{\mathbf{q}}^i_{g_j}\}$ of facial expression and gesture in unlabeled data. We have obtained the pseudo facial expression labels by running pre-trained 3D ResNet-18 on RAVDESS dataset [179]. Similarly, the pseudo gesture labels are obtained through pre-trained 3D ResNet-18 on

IsoGD dataset [180]. The overall loss of VAN is given by:

$$\pounds_{\mathbf{VAN}} = \beta_1 \pounds_{epr} + \beta_2 \pounds_{gst}, \tag{4.24}$$

where $\{\beta_1, \beta_2\}$ are regularization parameters that balance the contribution of both facial expression and gesture branch.

2) **Regularized objective function:** The backbone network along with VAN and LAN is trained to predict accurate probability of action classes. The main objective function $\pounds_{\mathbf{main}}$ of the overall architecture is formulated by:

$$\pounds_{\mathbf{main}} = \pounds_{\mathbf{SBN}} + \lambda_1 \pounds_{\mathbf{VAN}} + \lambda_{\mathbf{2}} \pounds_{\mathbf{LAN}}, \tag{4.25}$$

where $\{\pounds_{\mathbf{SBN}}, \pounds_{\mathbf{VAN}}, \pounds_{\mathbf{LAN}}\}$ are loss function of backbone, visual attention and long-term attention networks, respectively. $\{\lambda_{\mathbf{1}}, \lambda_{\mathbf{2}}\}$ are regularization parameters that balance the contribution of both VAN and LAN. The objective function for classification is given by:

$$\pounds_{\mathbf{SBN}} = -\frac{1}{N_b} \sum_{\mathbf{i}} \hat{\mathbf{a_{i,j}}} \ \mathbf{log} \ (\mathbf{a_{i,j}}), \tag{4.26}$$

where N_b stand for batch size and $\mathbf{i}$ is the class of a video segment in a batch. $\mathbf{a_{i,j}}$ is the predicted likelihood of an action on $\mathbf{j}$ video instance and $\hat{\mathbf{a_{i,j}}}$ is the ground truth label of an action. The second loss term consists learned weights $\pounds_{\mathbf{VAN}}$, which is computed using Eq. (4.24). The last loss function for the LAN is given by:

$$\pounds_{\mathbf{LAN}} = -\frac{1}{T_s}\frac{1}{C} \sum_{t=1}^{T_s} \sum_{c=1}^{C} (\hat{\mathbf{y}_t^c} \ \mathbf{log} \ \mathbf{y}_t^c) + \lambda_\Theta \left\| \Theta \right\|_2, \tag{4.27}$$

where T_s is total time steps, C is number of action classes, Θ denotes model parameters, λ_Θ is coefficient of weights, $\mathbf{y}_t^c$ denotes predicted class, and $\hat{\mathbf{y}}_t^c$ is ground truth of action classes.

3) **Joint training of overall architecture.** Optimization of SBN, VAN, and LAN is rather difficult due to mutual influence on each other. We incorporate a joint training model to train different modules of HAANet efficiently. To ensure the convergence of overall architecture, joint training procedure is elaborated in Algorithm 4.1.

```
Algorithm 4.1: Joint training of SBN, VAN, and LAN.
Input  : Model training parameters $\mathbf{N_1}$, $\mathbf{N_2}$, and $\mathbf{N_3}$.
Output: Converge overall architecture.
Initialize the network parameters using He.
//pretrain SBN: Fine-tune SBN by $\mathbf{N_1}$ iterations.
//pretrain VAN: With LAN weights being fixed as ones, jointly train SBN
    and VAN. Fine-tune SBN and VAN by $\mathbf{N_1}$ and $\mathbf{N_2}$ iterations.
//pretrain LAN: With VAN weights being fixed as ones, jointly train SBN and
    LAN. Fine-tune backbone network and long-term attention network by $\mathbf{N_1}$ and
    $\mathbf{N_3}$ iterations.
//jointly train HAANet: Fix both VAN and LAN learned in step 3 and 5.
    Jointly fine-tune the HAANet $\mathbf{N_1}, \mathbf{N_2}$, and $\mathbf{N_3}$ iterations.
```

4.5 Experimental Results

We discuss the experimental and ablation results in this section.

4.5.1 Datasets and Metrics

UCF101 [1] includes 13.3k video instances with 101 action classes that are clustered in 25 groups. In each group, there are 4-7 videos of an action. **HMDB51** [2] consist 6.8K videos collected from movies and Youtube with 51 action classes. It contains 3.7k training videos. **Kinetics400** [133] consists of 300k videos with 400 action categories, which is divided into 240k for training and 20k for validation. The rest 40k as test videos. **ActivityNet** [135] includes 203 action categories with 193 samples per category on an average. The trimmed videos present in the dataset is of 27.8k. An average duration of a video is 3-7 minutes. **Breakfast-Actions** [136] is a dataset of 10 cooking activities conducted by 52 individual actors in 18 distinct kitchen scenes. It consist of 1712 videos with 1357 as training data and 335 for testing purpose.

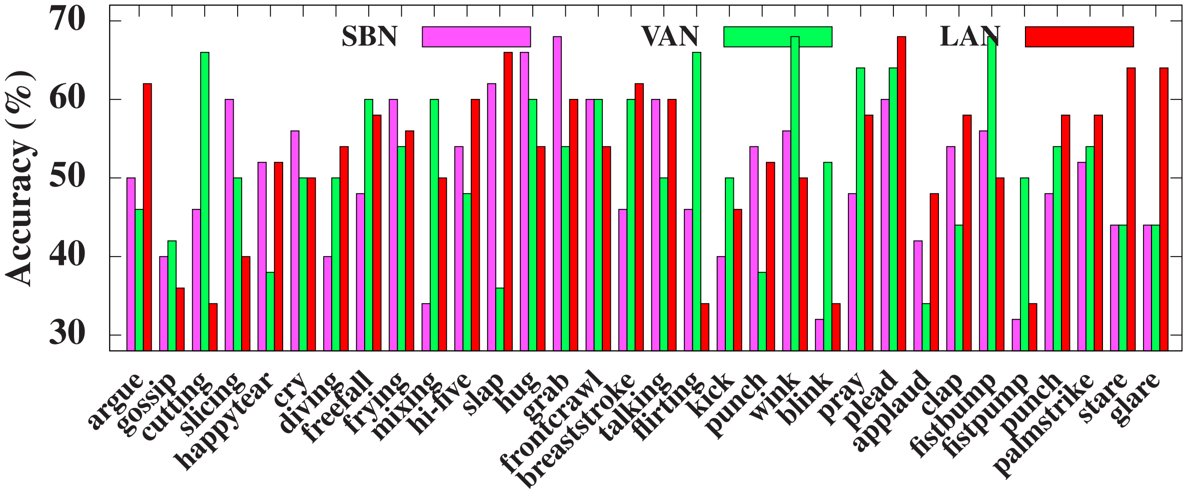


Figure 4.4: Mean accuracy of individual modules in VALC dataset.

Visual Attention with Long-term Context (VALC) is a dataset with 32 ambiguous action classes with 107 videos in each action class. We have addressed the ambiguous actions pairs and classify them based on important visual saliency cues. However, existing datasets such unambiguous class pairs. This motivated us to create a new dataset, named as *VALC* dataset. The ambiguous videos are collected from movies, YouTube, and web series, which are recorded in the presence of large variations in viewpoint, camera motion, illumination conditions, cluttered background, human appearance and pose, and object scale. VALC dataset mainly addresses the action class pairs which are ambiguous in nature in terms of facial expression and/or gesture of the actor in an action or long-term temporal context of the action video. Crowdsourcing is utilized for annotating videos. The statistics of the dataset are shown in Table 4.2.

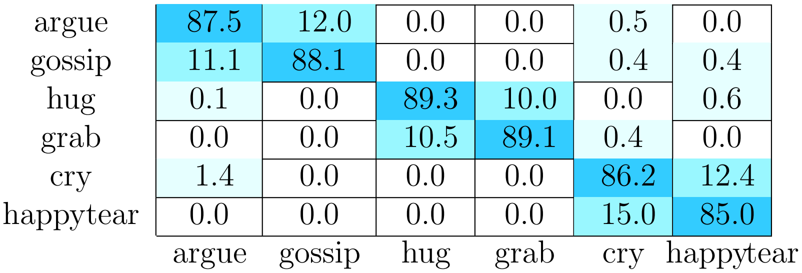
Table 4.2: Statistics of VALC dataset.

Attributes	**Values**	**Attributes**	**Values**
Actions	32	Min clip length	2 sec
Clips	3520	Max clip length	71 sec
Mean clip length	10 sec	Frame rate	30
Total duration	10 hour	Min. Resolution	320 × 240
Camera motion	Yes	Resources	YouTube and movies

We have shown the accuracy with respect to individual modules of HAANet in Figure 4.4. We observe that HAANet performs better in terms of accuracy but contains larger parameters as compared with I3D method as shown in Table 4.3 on VALC dataset. Moreover, overhead brought by HAANet is actually little larger *i.e.*, it occupies only +17.5 computation effort as compare to I3D. We report Top-1 and Top-5 metrics for Kinetics400 dataset, Top-3 for ActivityNet dataset, and mean accuracy (Acc.) metrics for rest of the benchmark datasets for performance evaluation of actions. The computational complexity of the HAANet is computed in respect to flop counts and number of training parameters, denoted by *params*. Figure 4.5 depicts confusion matrix for few classes.

Table 4.3: Performance of current SOTA methods on VALC dataset with computational complexity. Here parameters of HAANet is calculated after freezing SBN.

Methods	**VALC**	**params**
	(**Acc.**)	(in millions (**M**))
C3D [12]	70.2	78.41
I3D [140]	73.85	**12.70**
3D-ResNet-34 [15]	75.1	65.50
3D-ResNext-101 [15]	77.0	48.34
LGD [160]	79.35	46.40
HAANet	**80.9**	30.20



	argue	gossip	hug	grab	cry	happytear
argue	87.5	12.0	0.0	0.0	0.5	0.0
gossip	11.1	88.1	0.0	0.0	0.4	0.4
hug	0.1	0.0	89.3	10.0	0.0	0.6
grab	0.0	0.0	10.5	89.1	0.4	0.0
cry	1.4	0.0	0.0	0.0	86.2	12.4
happytear	0.0	0.0	0.0	0.0	15.0	85.0

Figure 4.5: The confusion matrix of VALC dataset.

4.5.2 Implementation Details

RGB inputs. We divide videos into K segments and randomly choose 1 frame from each segment. This selection furnishes robustness to variations and allows HAANet to exploit all RGB fully. HAANet directly extracts features and performs predictions on 16-frame clips, which are simultaneously processed by stacking 3D ConvNet. **VAN.** VAN consists of two branches, *i.e.*, facial expression and gesture. We independently train facial expression and gesture branches on RAVDESS and IsoGD datasets, respectively, to produce weighted scores accordingly after channel-wise global squeezed operation. **Hyper-parameters.** We start with learning rate of 0.01 for UCF101 and HMDB51, which is reduced by factor of 10 for every 30 epochs. Similarly, we train our model on Kinetic400, Breakfast-Actions, and VALC for 240 epochs except ActivityNet at 180 epochs. We use Adam optimizer and momentum of 0.9. We adopted weight decay of 5 $\times 10^{-4}$. We have utilized dropout layer and set dropout to 0.5. **Training.** HAANet is implemented based on Pytorch. We have trained model and computed all parameters with 8GTX 1080TI GPUs. Data augmentation is performed at the time of training to avoid overfitting through corner cropping and scale-jittering. Input size of video is $N \times T \times 112 \times 112$, where N is batch size and T is number of sample frames per video segment. In our experiment, we have set value of T to 16 and N as 30. **Inference.** We re-scale short side to 112 pixels of RGB frames. Alike [138], we randomly sample 10 times from full video and estimate individual softmax scores. We averaged softmax scores of all segments to obtain final prediction.

4.5.3 Ablation Study

Temporal term of video segment. We investigate the consequence of temporal length in an input video segment in Table 4.4. We have experimented with the number of frames from 8 to 32 on UCF101 and HMDB51 datasets. As number of frames increases, the precision and number of parameters increase. Therefore, there is a tradeoff between accuracy and computational complexity. Although, the parameters in case of temporal length of 8 frames are less, but it is unable to capture the motion information accurately. Thus, we have chosen a temporal term of 16 frames as it provides high accuracy. However, we observe that the performance of HAANet also degrades at larger temporal length since it may lead to a misclassification problem.

Table 4.4: Effect of temporal length on UCF101, HMDB51, and VALC.

#frames	**UCF101 (Acc.)**	**HMDB51 (Acc.)**	**VALC (Acc.)**	**FLOPs (/video)**
8	96.9	83.2	78.9	**0.5**×
16	**98.5**	**86.6**	80.9	1×
32	98.0	85.8	**81.1**	2×

Impact of backbone network. SBN is kept shallow to limit training parameters. Table 4.5 shows effect of variation in SBN keeping rest of the proposed HAANet same. As per Table 4.5 SBN is better than other variants with fewer parameters. It is observed that parameters in 3D ResNet, 3D ResNet34, 3D ResNet101 are too high that leads to overfit the model and drop test accuracy. We also investigate effect of use of LC on performance of HAANet. We observe that performance drops after removing LC from SBN.

Table 4.5: Effect of 3DResNet and C3D network as SBN on UCF101 and Kinetics400 datasets.

Backbone	**UCF101**		**Kinetics400**	
	#params	**Acc.**	**#params**	**Top-1**
3DResNet18	33.23M	95.4	33.38M	77.1
3DResNet34	63.50M	96.6	64.00M	80.2
3DResNet101	86.06M	94.6	86.20M	80.2
C3D (1 net)	78.41M	95.1	79.63M	81.3
SBN (with LC)	**15.71M**	**98.5**	**15.75M**	**83.0**
SBN (w/o LC)	15.71M	97.4	15.75M	81.0

Impact of different branches in VAN module. We have proposed VAN to focus on action-specific visual saliency information, such as, gestures and facial expression.

Table 4.6 shows effect of individual and fusion of visual saliency information. It is observed that fusion model that includes features of facial expression and gesture performs better than other individual models. VAN module performs an essential part in recognizing action classes with ambiguous pairs.

Table 4.6: Comparison of importance of FE, gesture, and FE + gesture on Breakfast-Actions and VALC datasets. FE is facial-expression.

Visual Attention	**Breakfast-Actions (Acc.)**	**ActivityNet (Top-3)**	**VALC (Acc.)**
FE	89.6	86.95	79.5
gesture	93.3	85.4	78.2
FE + gesture	**94.8**	**87.7**	**80.9**

Study on pooling mechanisms. We have exploited different types of pooling mechanisms: max, average, standard deviation, and attention pooling in Table 4.7. It is clear from our experimentation that temporal attention pooling provides better results for different datasets.

Table 4.7: Impact of temporal pooling mechanisms on HMDB51, BreakFast-Actions, and VALC datasets.

Temporal Pooling	**HMDB51 (Acc.)**	**BreakFast-Actions (Acc.)**	**VALC (Acc.)**
max	84.0	92.9	78.7
average	82.4	92.1	76.7
standard deviation	85.5	93.5	79.2
attention	**86.6**	**94.8**	**80.9**

Impact of individual modules and Visual Results. In Table 4.8, we examine how well an individual module can perform in recognition task. First, we eliminate visual attention and long-term attention networks from HAANet and name the rest as *SBN only*. Similarly, VAN module is only initialized with the backbone network and named as *VAN only*. Furthermore, only long-term attention network is initialized with the backbone network and denoted by *LAN only*. We also have evaluated impact of standalone backbone network with the visual attention network, which is represented

Table 4.8: Effect of backbone network, visual attention, and long-range context modules on Kinetics400, VALC, and ActivityNet. SBA+LAN is model without using visual saliency cues.

Models	**Kinetics400 (Top-1)**	**ActivityNet (Top-3)**	**VALC (Acc.)**
SBA only	78.2	82.5	71.0
VAN only	78.8	82.9	73.6
LAN only	79.1	83.2	74.9
SBA+VAN	79.5	83.8	78.4
SBA+LAN	80.8	85.2	78.6
VAN +LAN	81.8	86.5	79.7
SBA+VAN+LAN	**83.0**	**87.7**	**80.9**

by *SBN+VAN*. Next, we exclude VAN from HAANet and name the rest as *SBN+LAN*. Likewise, we study the impact of visual attention with a long-term attention network, which is indicated by *VAN+LAN*. All modules have notable impact on the accuracy of HAANet but our proposed variant outperforms other variants. We show the visual results in Figure 4.6.

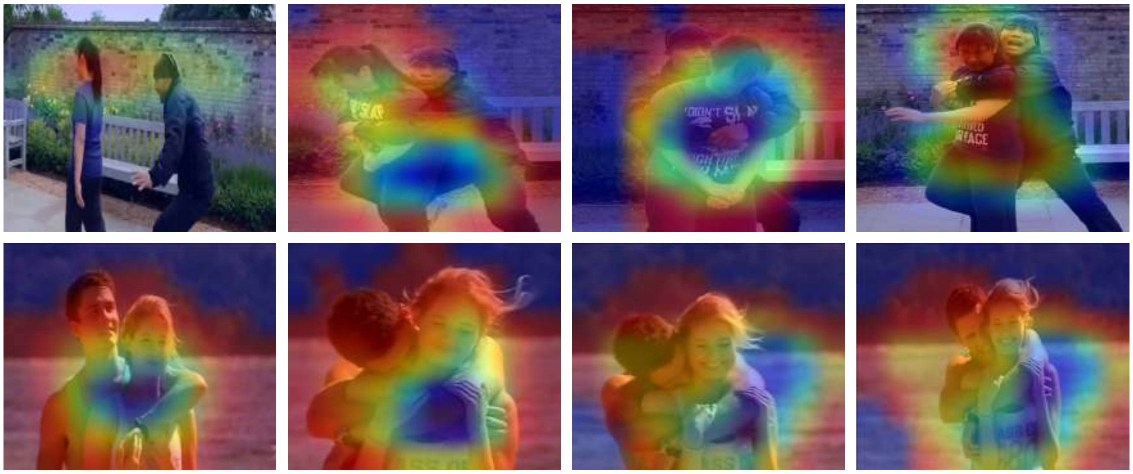


Figure 4.6: Attention maps of two videos of VALC dataset, which indicate a pair of ambiguous actions, like 'grab' and 'hug'.

4.5.4 Comparison with SOTA on benchmark datasets

We compare HAANet with recent literature on five benchmark datasets. By default, we consider RGB modality, if not mentioned. Table 4.9 shows the performance comparison of HAANet with SOTA methods on UCF101 and HMDB51 datasets. Our model achieves comparable performance on UCF101. On HMDB51, our model achieves the highest improvement of 1.5% as compared to R(2+1)D BERT. We observe that HMDB51 datasets have discriminative features of facial expression as well as gesture due to which our model better than others. Table 4.10 shows the performance score on the Kinetics400 dataset in terms of Top-1 and Top-5 accuracy. Furthermore, we provide performance comparison of our model on Breakfast-Actions and ActivityNet datasets considering long-term dependencies in Table 4.11.

Table 4.9: Comparison with SOTA methods for short-term videos on UCF101 and HMDB51 datasets.

Methods	**UCF101 (Acc.)**	**HMDB51 (Acc.)**
C3D [12]	85.2	-
I3D [140]	84.5	49.8
ECO [143]	94.8	72.4
TSM [159]	94.5	70.7
STM [138]	96.2	72.2
MARS [146]	95.6	73.1
LGD [160]	97.0	75.7
Quan *et al.* [161]	91.84	48.81
Early fusion + I3D [163]	98.2	81.1
R(2+1)D BERT (32f) [164]	98.65	83.99
R(2+1)D BERT (64f) [164]	**98.69**	85.1
Ours	98.5	**86.6**

4.6 Conclusion of the Chapter

Emotions and gesture are essential elements in improving social intelligence and predicting real human action. In recent years, recognition of human visual actions using

Table 4.10: Comparison with SOTA methods on Kinetics400 dataset and extra cues. Here OF is optical flow and FE is facial expression

Methods	**Top-1**	**Top-5**	**extra cues**
I3D [140]	71.1	89.3	RGB+OF
ECO [143]	70.7	89.4	RGB
Non-Local Network [137]	77.7	93.3	RGB
TSM [159]	72.5	90.7	RGB
STM [138]	73.7	91.6	RGB
Slow Fast + NL [150]	79.8	93.9	slow-fast RGB
ir-CSN-152 [151]	82.6	95.3	RGB
MARS [146]	72.8	-	RGB
LGD [160]	79.4	94.4	RGB
ViViT [166]	81.7	93.8	RGB
Ours	**83.0**	96.6	RGB + FE + gestures

Table 4.11: Comparison with SOTA methods for long-term videos on Breakfast-Actions and ActivityNet datasets.

Methods	**Breakfast-Actions (Acc.)**	**ActivityNet (Top-3)**
I3D [140]	80.64	-
Non-local [137]	83.79	-
Timeception [121]	86.93	-
C3D [12]	-	81.16
Ours	**94.8**	**87.7**

deep neural networks has gained wide popularity in multimedia and computer vision. However, ambiguous action classes like “praying“ and “pleading“ are still challenging to classify due to similar visual cues of action. We need to focus on attentive associated features of facial expression and gestures, including the long-term context of a video for correct classification of ambiguous actions. We have devised an effective supervised DNNs for action recognition that captures long-term dependencies of video. HAANet efficiently distinguishes ambiguous action classes based on emotion and gesture. We utilize lateral connections in shallow backbone network for learning fine-detail enrich spatio-temporal features. Emotions and gestures features are learn in visual attention network. We also extract class-aware spatio-temporal cues in temporal pooling to capture informative semantic features over the time. We deduce joint training model to train the HAANet efficiently. HAANet outperforms SOTA literature on ActivityNet, Breakfast-Actions, and UCF101 datasets.

4.7 Publication related to the Chapter

1. Nitika Nigam and Tanima Dutta, “Emotion and Gesture Guided Action Recognition in Videos Using Supervised Deep Networks,” in IEEE Transactions on Computational Social Systems, 2022 (Early Access), doi: 10.1109/TCSS.2022.3187198.