IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 25, 2020, accepted September 11, 2020, date of publication September 22, 2020, date of current version October 1, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3025931

Multi-Label Multi-Class Action Recognition With Deep Spatio-Temporal Layers Based on Temporal Gaussian Mixtures

YURI YUDHASWANA JOEFRIE[1,2] **AND MASAKI AONO**[1], **(Member, IEEE)**

[1]Department of Computer Science and Engineering, Toyohashi University of Technology, Toyohashi 441-8580, Japan
[2]Department of Information Technology, Universitas Tadulako (UNTAD), Palu 94118, Indonesia

Corresponding author: Yuri Yudhaswana Joefrie (yudhaswana@kde.cs.tut.ac.jp)

This work was partially supported by JSPS (Japan Society for the Promotion of Science), Grants-in-Aid for Scientific Research (B) Grant Number 17H01746, and was partially supported by JICA (Japan International Cooperation Agency).

ABSTRACT Current action recognition studies enjoy the benefits of two neural network branches, spatial and temporal. This work aims to extend the previous work by introducing a fusion of spatial and temporal branches to provide superior action recognition capability toward multi-label multi-class classification problems. In this paper, we propose three fusion models with different fusion strategies. We first build several efficient temporal Gaussian mixture (TGM) layers to form spatial and temporal branches to learn a set of features. In addition to these branches, we introduce a new deep spatio-temporal branch consisting of a series of TGM layers to learn the features that emerged from the existing branches. Each branch produces a temporal-aware feature that assists the model in understanding the underlying action in a video. To verify the performance of our proposed models, we performed extensive experiments using the well-known MultiTHUMOS benchmarking dataset. The results demonstrate the importance of our proposed deep fusion mechanism, contributing to the overall score while keeping the number of parameters small.

INDEX TERMS Action recognition, motion detection, multi-branch network, multi-layer neural network, spatio-temporal branch, videos.

I. INTRODUCTION

Unquestionably, action recognition is currently a topic of active research due to not only the challenges that researchers must be overcome in this field but also due to the importance of action recognition applications in our daily lives. Such applications include criminal detection, more timely alert systems for natural disasters, video summarization, and video recommendation systems. One of the challenges in classifying an action in a video is to capture accurately the hidden temporal relation between frames in addition to capture the spatial semantics. To take both aspects into account, the filters of the convolutional layer must operate on 3-dimensional data (3D), in contrast with the more general approach for image classification that uses only 2-dimensional (2D) convolution.

The use of 3D convolution has been shown to be effective in capturing both temporal and spatial dimensions. Although a video consists of spatial and temporal aspects, there have been several attempts to understand the variations in actions through the use of 2D convolution without sacrificing accuracy [1], [2]. Such 2D convolution helps the computer to train faster and reduce the inference times than would be the case using 3D convolution.

A different approach has been proposed to use optical flow fields as an additional input modality, while keeping the RGB images as the main input. This modality focuses only on pixel displacement over time, thereby removing unnecessary spatial information contained in frames. Optical flow is computed by taking the difference of two consecutive frames, hence providing useful information regarding the motion.

Because more than one modality is used, several studies have used at least two separate architectures to accommodate different modalities to learn the spatial and temporal information separately. Optical flow components from the RGB images are extracted and then fed together into separate architectures, that will be referred to as the ‘‘branches’’ for each separation in this paper. Each branch is expected to be complementary to the other branches. Previous works based

The associate editor coordinating the review of this manuscript and approving it for publication was Xianye Ben.

on this approach [3]–[5] have obtained impressive results. Different architectures have been introduced that use the recurrent model to understand data sequences. Considering frames in a video as a sequence of items of information, we can consider this technique as aiming to capture temporal relations contained in frames.

All of the methods mentioned above are suitable only for capturing a fixed length of video data, i.e., regardless of the length of the video, frames are sampled to determine the action category. A study by Piergiovanni and Ryoo [6] addresses this drawback by attempting to learn temporal structure using notably fewer parameters of a custom convolutional layer. Their work successfully implements a temporal Gaussian mixture (TGM) layer on top of two-stream inflated 3D ConvNets (I3D) [7] and InceptionV3 [8] to reveal the temporal relations in videos. Specifically, [6] employs a set of Gaussian-controlled filters/kernels responsible for detecting frames to which the model should pay more attention. In addition to its ability to capture the temporal structure of an activity, their system also features a kernel that is independent of the duration of a video due to its fully convolutional design. This design also provides the ability to handle very long videos. Moreover, they also conduct experiments using a favored ''two-stream'' configuration design to classify actions in a video.

One would question the possibility of mixing both RGB and optical flow feature representations and then processing the resulting representations further using a TGM layer. Previous work [6], [7] appears to use only two streams, neglecting the potential contribution of mixing these two representations. This is understandable because several research efforts reported in the literature, as described later, have demonstrated the significance of mingled modalities.

Based on this intuition, we propose three new fusion models that are considerable improvements on the previous work. We utilize different fusion mechanisms between the different branches of the TGM layers to enrich a model with a refined feature set. Concretely, we propose combinations of spatial and temporal feature set on a temporal branch and on a new spatio-temporal branch at different levels. The main contributions of our work are summarized as follows:

1) **A series of TGMs as a deep spatio-temporal branch.**
We propose a deep spatio-temporal branch comprised by several TGM layers to boost the accuracy in estimating the multi-label, multi-class problem for a given action video.
2) **Unified multi-branch action recognition architecture.**
We combine the Gaussian-based spatial and temporal branches with a deep spatio-temporal branch in a unified architecture. These branches are named based on their input modalities.
3) **Custom TGM layer on a temporal branch.**
We replace the 1 × 1 convolution with the maximum function inside the TGM layer, i.e., taking the maximum value on the input-channel axis, in order to make the TGM layer of the temporal branch more aware of the distinctive action features. This replacement results in finer action features for the subsequent layer. We confirm that this approach is superior to that of the original work.

The remainder of this paper is organized as follows. **Section 2** describes related studies and explains their principal concepts. In **Section 3**, we introduce our proposed TGM-based multi branch network. **Section 4** includes the details of our experiments and evaluation and describes the comparison with state-of-art methods. It also includes an ablation study to determine the optimum result. **Section 5** provides concluding remarks regarding our work and points out future directions.

II. RELATED WORK

We describe some work related to hand-engineered features and deep neural networks for action recognition.

An action can be described as a sequence of primitive movements [9], e.g., kicking a ball and closing something. Such action sometimes consists of several simple actions, e.g., cooking involves taking ingredients and pouring them into a pan. Prior to the development of deep neural networks, several traditional algorithms existed to quantify inputs for action recognition in a video. Those algorithms use two modalities to produce a quantified vector. The RGB images are known to contribute significantly to action recognition systems. A study in [10] explained very well the progress in action recognition based on the use of the RGB data. Other classic techniques have been developed by Bobick and Davis *et al.* [11], who created motion templates using methods called motion energy image (MEI) and motion history image (MHI). Then, a work by Klaeser *et al.* [12] introduced a 3D version of histogram of gradients (3DHOG), an extended version of 2DHOG, to project human action in a space-time dimension. Another work by Scovanner *et al.* [13] extended the scale invariant feature transform to 3DSIFT and used a bag of words to represent videos. Schuldt *et al.* [14] used a support vector machine (SVM) to classify space-time 3D descriptors into action categories. Ben *et al.* [15], [16] introduced a technique called coupled patch alignment and a general tensor representation framework for gait recognition. In addition, an optical flow field plays an essential part in action recognition to increase the robustness of a system, as can be seen in [17]–[19]. Generally, handcrafted spatial and temporal video features have been used to describe spatial and temporal aspects that will be used in classic machine learning to determine an underlying action.

Since convolutional neural networks have emerged to outperform any existing traditional machine learning techniques, researchers have competed by using them to build very deep neural networks. In one preliminary attempt, researchers used 2D convolution to learn an action frame by frame, as described in [20], [21]. This 2D convolution operated on one frame of a video at a time. It was found that convolution

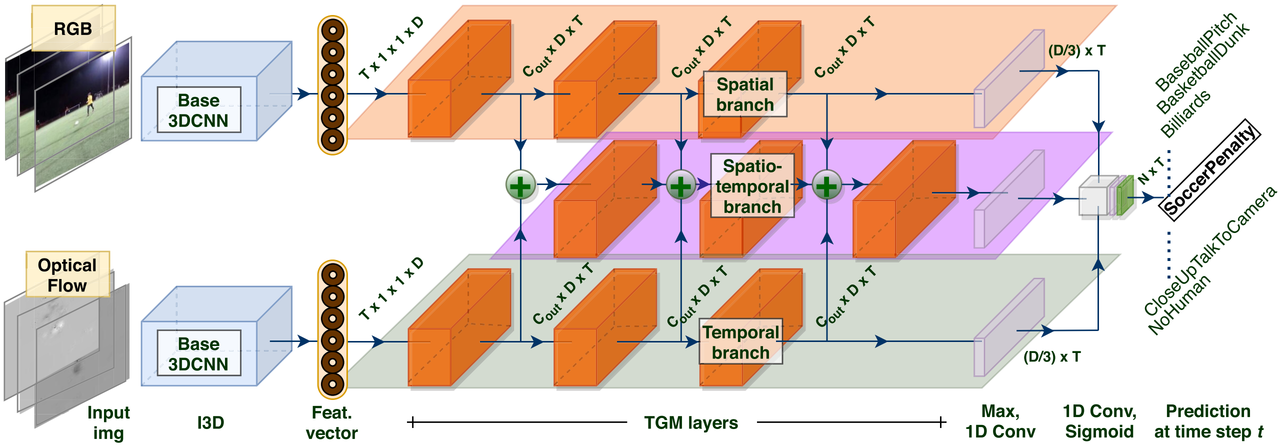

FIGURE 1. Overview of our best proposed model. For each branch, we add 3 layers of TGM to learn complex temporal structures. On top of each branch, we reduce the output dimension ($C \times D \times T \rightarrow D \times T$) and adjust the feature maps dimension (D) to be three times lower than the original dimension by using max function and 1D convolution, respectively. The outputs from three branches are then concatenated on the dimension axis and passed to another 1D convolution to map from D size (i.e., 1,024) to N size (the number of classes, i.e., 65). Last, the sigmoid function is used to obtain the prediction for each time step. For simplicity, we omit a shortcut connection, dropout, and ReLU layer.

in 2D space still performs well on some datasets although it does not model temporal patterns. Since the video naturally contains motion, 3D convolution has been proposed in recent studies and is expected to show improved performance because it learns spatial and temporal dependencies simultaneously. The work in [22]–[24] utilized a 3D convolution kernel to include motion cues. Other work that uses 3D convolution includes a 3D version of Inception-V1 by Carreira and Zisserman [7] and 3D ResNet by Hara *et al.* [25]. The achievements of image classification inspires such work; the researchers inflated all of the 2D filters and the pooling windows to enable the network to accept a stack of frames. In other words, they replaced all $N \times N$ filters with a cuboid version $-N \times N \times N$. Many researchers have been inspired by 3D ResNet and have adopted it in their studies [26], [27]. Several studies have also investigated the use of a recurrent network, such as long short-term memory (LSTM) or bi-directional long short-term memory (Bi-LSTM) to classify an action, as described in [28]–[31]. These authors argued that implementing a recurrent network on top of a convolutional neural network (CNN) backbone will enable the capture of sequential information of a video.

Insightful and innovative multi-stream methods have also been reported [4], [6], [32], [33]. As mentioned earlier, each stream conveys a different type of data. The types of data can be partitioned into several categories such as RGB, optical flow, RGB difference, and audio. Several fusion mechanisms have been demonstrated. Chi *et al.* [34] adopted a self-attention mechanism for which the input sources are RGB and optical flow. Feichtenhofer *et al.* [3] investigated several techniques for fusing different modalities at specific layers and proposed a fusion scheme containing two fusion strategies, namely, spatial and temporal. In the current work, we consider only spatial fusion, leaving temporal fusion for future research. Among spatial fusion strategies (Sum, Max, Concatenation and Conv fusion) on the UCF101 benchmarking dataset [35], these authors reported that the Sum fusion outperforms other functions except for Conv fusion. Of course, learning randomly initialized weights when using the Conv fusion consumes more training time compared to Sum fusion because the latter operates on two feature maps by simply summing them. We use the Sum fusion function in our work because it is simple and yet quite robust.

III. OUR PROPOSED MULTI GAUSSIAN-BASED BRANCH

A. MULTI TGM LAYERS

We introduce our new architecture that consists of several Gaussian kernel-based branches. The overall structure of our best model is shown in Figure 1. As seen in the figure, our model uses two modalities, RGB data and optical flow. Thus, we have two branches based on their input types, namely, spatial and temporal, to classify the actions. Given the outputs from the base CNN, represented as $F \in \mathbb{R}^{T \times 1 \times 1 \times D}$, we first propagate F to the spatial and temporal branches. T and D are the temporal length and the feature maps dimension, respectively. We set the number of the TGM layers to three for each branch, identical to that of the referenced work [6]. Inside the TGM layer, we learn a set of the Gaussian mixture kernels. For each input channel $j \in [1, C_{in}]$ and each output channel $i \in [1, C_{out}]$, the associated filters will convolve on the temporal input feature x and map the resulted channels into a single channel to produce s_i:

$$s_i = (x * K_{i,j}) * w_i \tag{1}$$

where $\mathrm{K} = [K_1, K_2, \ldots, K_{c_{out}}]$ denotes the Gaussian mixture kernels corresponding to each input and output channel and w_i is a 2D convolution with 1×1 kernel size and one as the output channel, followed by the rectified linear unit (ReLU) activation function. We include more details of the

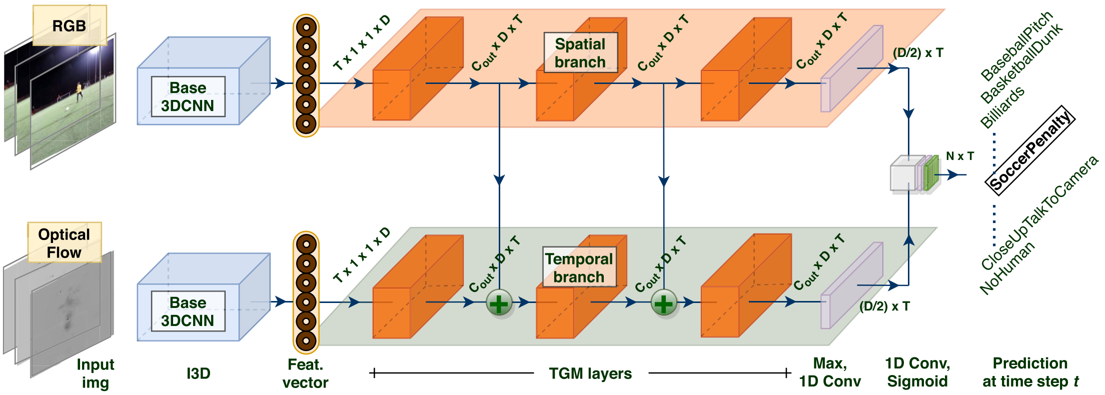

FIGURE 2. The upper branch indicates the spatial branch and the other branch is the temporal branch. Fusion occurs in the temporal branch. The output ($C_{out} \times D \times T$) from the earlier TGM layers of the upper branch is fed not only to the subsequent layer but also to the temporal branch. A simple addition operation is performed in the temporal branch to guarantee one-way information sharing. An explanation for the remaining parts of the diagram is identical to that of Figure 1.

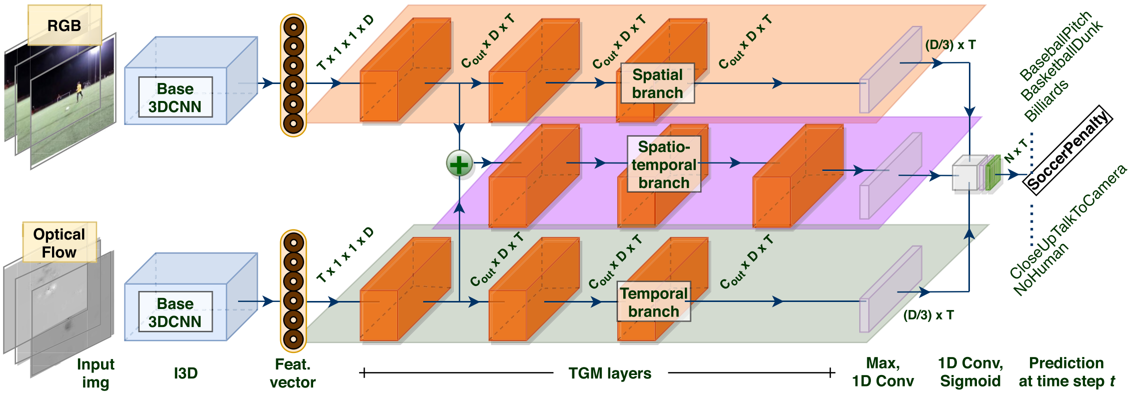

FIGURE 3. Different from Figure 2, a middle branch is created to learn representations simultaneously using blended features rather than passing the representation from one branch to another existing branch. The name "spatio-temporal" branch is defined to this new branch. Notice that the addition operation only occurs in the beginning of the spatio-temporal branch. With only a few extra parameters, this configuration surpasses Proposed #1 (see Table 1 for this comparison). The remaining parts of the diagram are identical to that of Figure 1.

TGM kernel/layer in the subsequent section. Notation for channel-wise operation and ReLU activation function is ignored for simplicity. The obtained s_i is then stacked on the channel axis to produce $S_{rgb} = [s_1, s_2 \cdots, s_{C_{out}}]$ with a dimensionality of $C_{out} \times D \times T$, where C_{out} is the number of the output channels. C_{in} and C_{out} can be considered as hyperparameters. We set C_{out} to four in this work. The value of D is consistent throughout all of the TGM layers (i.e., 1,024).

Going beyond the previous work, we argue that the maximum function accentuates the distinctive, important aspect of temporal features. Thus, for the temporal branch (S_{flow}), we replace 1×1 convolution + ReLU (Equation (1)) with the aggregate function. Mathematically, this can be formulated as:

$$s_i = \max(x * K_{i,j}) \tag{2}$$

In Equation (2), we replace w with the max function operating on the channel axis. The output s_i is then appended along the channel axis to obtain S_{flow} which is the $C_{out} \times D \times T$ representation, identical to that of S_{rgb}. Then, S_{rgb} and S_{flow} are fed forward to the next layer. Figure 7 illustrates the overall process inside the TGM layer in a single branch. In addition, the output of the base model CNN and the last TGM layer are concatenated (see illustration in Figure 4).

B. ROLE OF TGM KERNEL

The Gaussian mixture kernel used in this paper is introduced in [6]. In essence, this kernel is a constrained kernel governed by the variance of the Gaussian, i.e., a center μ and a width σ for which the values are in the positive range. As observed in Figure 5, this layer also includes an attention mechanism widely used in computer vision and language processing. A soft attention is applied to each Gaussian distribution to enable the layer to focus on the relevant parts in a temporal sequence.

We would like to highlight the implementation of the TGM layer in our proposed spatio-temporal branch. In the original

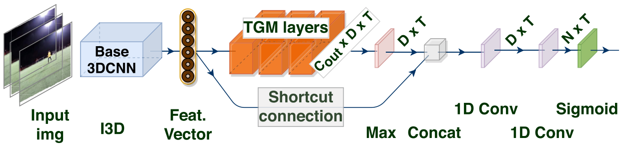

FIGURE 4. The output from 3DCNN is concatenated with the output from the last TGM layer. This shortcut connection prevents the information loss during the learning phase. Squeeze and permutation are necessary for the output of base 3DCNN ($T \times 1 \times 1 \times D \rightarrow D \times T$) prior to the concatenation. This figure illustrates only one branch and thus no fusion is occurred.

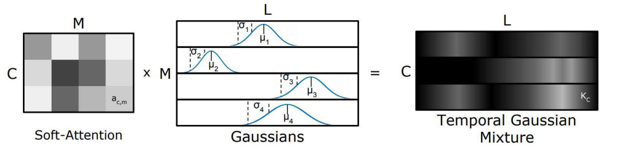

FIGURE 5. A temporal Gaussian mixture (TGM) kernel with multiple temporal convolution C and length L is computed from learnable attentive parameters and multiple Gaussian distribution. M denotes the number of Gaussian distributions. For each Gaussian distribution, there are 2 constrained variables: a center μ and a width σ. The image is taken from [6].

paper, the authors implement the TGM layer only on the RGB and optical flow modality, whereas the TGM layer is utilized on mixed modalities in our implementation.

C. OUR PROPOSED FUSION MODELS

We propose three fusion mechanisms to enhance the performance of the model.

1) Spatial and temporal fusion model (**Proposed #1**)
As observed from Figure 2, a fusion is introduced between the TGM layers. This type of combination is similar to those in work by Feichtenhofer *et al.* [36]. The two lateral connections are established from the spatial branch to the temporal branch in order to merge a meaningful representation of the RGB data. In contrast to their work where the outputs are transformed prior to fusing, we merely add the RGB and optical flow features because the shape and length of those two type of features already match. Given S^{rgb} and S^{flow} that are the outputs of the previous TGM layers in each branch, a new S^{flow} is determined according to the following equation:

$$S^{flow}_{i+1} = S^{flow}_{i} \oplus S^{rgb}_{i}, \quad i = [0, 1] \tag{3}$$

where i is the index where fusion occurs.

2) Early spatio-temporal fusion model (**Proposed #2**)
In this approach, we construct a new pathway from the existing branches. We introduce a spatio-temporal branch with a fusion occurring at the beginning of the branch (see Figure 3). We want our model to learn not only spatial and temporal information separately but also spatio-temporal information simultaneously. We are confident that the model will benefit from this fusion. Formally, a new spatio-temporal branch (S^{st}) results from the element-wise addition of S^{rgb} and S^{flow}.

$$S^{st} = S^{rgb}_{i} \oplus S^{flow}_{i}, \quad i = 0 \tag{4}$$

3) Multi-level spatio-temporal fusion model (**Proposed #3**)
In contrast to Proposed #2, we carry out a fusion strategy at several levels. We argue that each level of the TGM layers produces different temporal activity patterns. As described in Equation (5), given features S_{rgb} and S_{flow}, we perform element-wise addition at different levels to form the spatio-temporal branch, denoted as S_{st}.

$$S^{st}_{i} = S^{rgb}_{i} \oplus S^{flow}_{i}, \quad i = [0, 1, 2] \tag{5}$$

We assign the term "spatio-temporal" to our new branch since this branch operates on mingled modalities, namely, the RGB and optical flow components. The term "spatio-temporal" is also used to describe the role of 3D ConvNets; i.e., a 3D kernel convolves not only on the surface of the feature maps but also on the depth of the temporal axis of the feature maps.

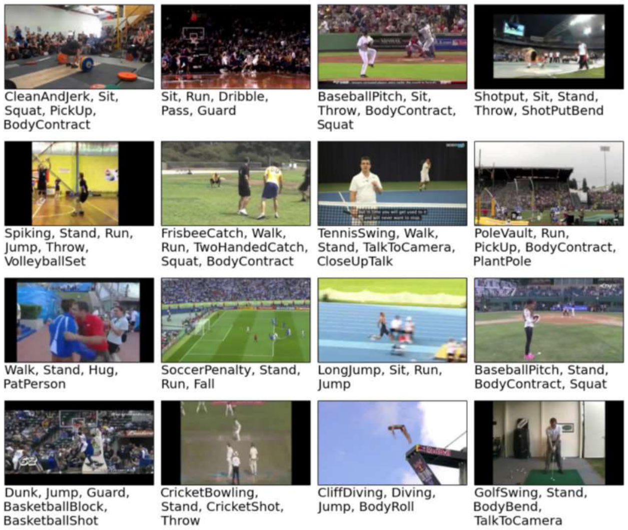
CleanAndJerk, Sit, Squat, PickUp, BodyContract
Sit, Run, Dribble, Pass, Guard
BaseballPitch, Sit, Throw, BodyContract, Squat
Shotput, Sit, Stand, Throw, ShotPutBend
Spiking, Stand, Run, Jump, Throw, VolleyballSet
FrisbeeCatch, Walk, Run, TwoHandedCatch, Squat, BodyContract
TennisSwing, Walk, Stand, TalkToCamera, CloseUpTalk
PoleVault, Run, PickUp, BodyContract, PlantPole
Walk, Stand, Hug, PatPerson
SoccerPenalty, Stand, Run, Fall
LongJump, Sit, Run, Jump
BaseballPitch, Stand, BodyContract, Squat
Dunk, Jump, Guard, BasketballBlock, BasketballShot
CricketBowling, Stand, CricketShot, Throw
CliffDiving, Diving, Jump, BodyRoll
GolfSwing, Stand, BodyBend, TalkToCamera

FIGURE 6. Some videos of the MultiTHUMOS dataset. Each video may contain multiple activities; hence, the task is categorized as a multi-label multi-class classification problem. The image is taken from [37].

IV. EXPERIMENT

In this section, we describe our experiment in detail. We demonstrate the implementation of the TGM layer in each branch and simultaneously fuse the spatial and temporal branches to achieve a positive result. Moreover, we conduct some ablation studies to emphasize the benefit of the *Max* function over 1 × 1 convolution inside the TGM layer and to investigate the benefits of the weighted branch scheme.

A. DATASET

We conducted some experiments on the untrimmed, multi-label MultiTHUMOS dataset [37] to investigate the

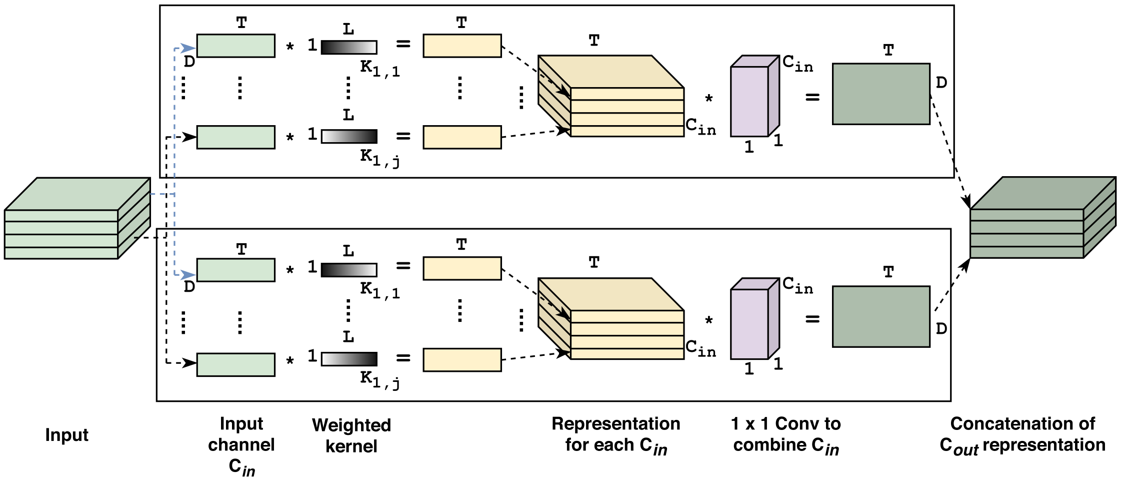
T
D
L
1
K1,1
K1,j
T
Cin
1
1
Cin
D
Input
Input channel Cin
Weighted kernel
Representation for each Cin
1 x 1 Conv to combine Cin
Concatenation of Cout representation

FIGURE 7. Inside the TGM layer. A Gaussian weighted kernel with length L is multiplied by each input channel C_{in} to produce a tensor with the shape of $C_{in} \times D \times T$. We apply a slight alteration to this layer: we substitute 1 × 1.2-dimensional convolution (indicated by the purple box) with the max function to combine the input channels. We note that the figure illustrated above that was taken from [6] represents the original TGM layer.

effectiveness of the spatio-temporal branch. This dataset is an extension of the well-known THUMOS dataset, with its number of action classes extended from 20 classes to 65 classes. Verbs are taken from the original THUMOS with several additions of various activities.

In total, we examined 30 hours of duration from 400 videos. This contains 38,690 annotations with every frame having 1.5 labels on average, notably higher than THUMOS (0.3 per frame). Each video has 10.5 action classes, increasing sharply from the 1.1 of THUMOS. Examples of the videos are illustrated in Figure 6.

B. FEATURE EXTRACTION

Prior to video feature extraction, we first extract all of its RGB frames and optical flow. We crop all of the extracted frames to a window size of 224 × 224 at the center to match the base model's input dimensions. We also normalize all pixels to have values with a range of $[-1\ldots 1]$. In addition, we use the well-known TVL1 algorithm [38] to compute the optical flow. The next step is to extract the video features from the image data. Prior to extracting, we load our model with the weights pre-trained on the ImageNet and the Kinetics dataset. This is a common transfer learning technique. We propagate onward our collection of frames via the I3D network to obtain the activations. We select the last average pooling *AvgPool3d* of I3D to serve as an endpoint logit for feature extraction, same as the referenced work. The extracted shape of each video is $T \times 1 \times 1 \times D$. The value of T can be varied depending on the length of a video whereas D is consistent for all videos (i.e., 1,024). We note that the TGM layer can accept arbitrary temporal lengths. The output features are then formatted as NumPy arrays and saved to a disk. If the number of frames is excessively high (i.e., a video has a long duration), then we divide it by the threshold (in our case, we set it to 100) and forward propagate the current chunk to the I3D network. The value of 1,024 represents the number of channels. The term ''channel'' here can be ambiguous. We typically use this word to describe the number of feature maps, whereas ''channel'' in the TGM layer, represented by C_{in} and C_{out}, refers to the number of Gaussian mixtures.

C. TRAINING AND TESTING

We conduct all experiments using the PyTorch framework. The Adam optimizer [39] is applied to optimize the model's parameters. We choose 5e−3 as our starting learning rate and decrease it if the loss becomes saturated. The training routine is finished at 60 epochs. To minimize the loss, we use the binary cross-entropy loss function, estimated as follows:

$$\mathcal{L} = -\sum_{i=1}^{N} y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i)) \quad (6)$$

where $\mathcal{L}$ is the loss score, y_i is the true label for a specific class occurring at a specific time, and $p(y_i)$ is our model's prediction. To measure the performance, we define the mean average precision (mAP) as follows:

$$mAP = \frac{1}{N} \sum_{c=1}^{N} AP_c \quad (7)$$

According to Equation (7), the mAP is calculated by summing each AP (AP_c) and then dividing by N (the number of queries), whereas the AP for a specific class is computed using *precision* at each relevant position n:

$$AP = \frac{\sum_{n=1}^{M} \text{Precision}(n)}{M} \quad (8)$$

where M is the total number of actions predicted.

TABLE 1. Comparison with other state-of-the-art methods on a popular benchmark MultiTHUMOS dataset. Even with fewer output channels (C_{out}) for each TGM layer, we consistently outperform the baseline and other methods.

Method	# of Output Channels	mAP (%)
Two-stream by Yeung *et al.* [37]	—	27.6
Two-stream + LSTM by Yeung *et al.* [37]	—	28.1
Multi-LSTM by Yeung *et al.* [37]	—	29.7
Predictive-corrective by Dave *et al.* [40]	—	29.7
SSN by Zao *et al.* [41]	—	30.3
TGM by Piergiovanni *et al.* [6] (baseline)	[4, 4, 4]	34.2
Proposed #1 (Ours)	[4, 4, 4]	**35.1**
Proposed #2 (Ours)	[4, 4, 4]	**35.7**
Proposed #3 (Ours)	[4, 4, 4]	**37.1**

D. RESULTS

Clearly stated, our aim is to improve previous on the work [6] by implementing a variety of fusion schemes to increase the mAP. We postulate that a fusing mechanism is important for accuracy. In this section, we compare our three proposed models with the baseline and state-of-the-art methods. We evaluate three type of fusions and conduct short analyses based on the results.

An examination of the results presented in Table 1 shows that by integrating the information of the spatial branch into the temporal branch, the improvement of Proposed #1 over the baseline is marginal (0.9% higher). This confirms the advantage of the one-way information sharing of two branches. In the next approach, the model Proposed #2 does benefit from having a newly created branch in the form of a 1.5% improvement over the baseline. This certifies that inserting a new branch and simultaneously optimizing the weights of all of the branches will improve the performance to some extent. Our last and best model, Proposed #3, outperforms the baseline method by 2.9%. We believe that this model captures more complex, nonlinear temporal features, and thus contribute significantly to the performance of the model.

We can conclude that a fusion mechanism demonstrates consistent performance over the baseline model and other existing models on the MultiTHUMOS dataset.

E. ABLATION STUDY

This section describes experimental studies on MultiTHUMOS. We conducted several comparisons to identify the advantageous strategies. We experiment with various channel combination strategies. We also conduct ablation experiments of weighted scheme for spatial and temporal branches.

1) CUSTOM CHANNEL UNIFICATION

Referring to Figure 7, we observe that the original version applies a 2D convolution to combine temporal reasoning features on the channel axis (see Equation (1)). This 1 × 1.2D convolution is designed to combine all of the C_{in} into one channel.

TABLE 2. Results for different channel combinations. Interestingly, different modalities yield different results for the same operation. This table also emphasizes that the RGB images is more critical modality than the optical flow in term of accuracy. Note that "3" in 3TGM refers to the number of TGM layers used for this ablation experiment.

Channel Combination Mode	mAP (%)	
	Spatial (RGB)	Temporal (Optical Flow)
3TGM with 1 × 1 Conv	**32.5**	16.9
3TGM with summation	29.7	16.8
3TGM with average	32.1	17.0
3TGM with max	31.5	**18.1**

To the best of our knowledge, there are simpler, less expensive yet effective techniques for achieving channel combination: summation, average, and maximum. These functions have no parameters to optimize and thus have a lower computational cost. We performed several experiments to determine the effectiveness of each function compared to those obtained in the original work. The results are presented in Table 2. We observed that 1 × 1 convolution is more beneficial to the spatial branch and, conversely, is not useful for the temporal branch. Surprisingly, the TGM layer with the maximum function performs better on the temporal branch. We hypothesize that the reason for this function showing little improvement is that each pixel in the temporal branch corresponds to small changes; thus, the maximum function ensures that the model takes only distinctive pieces of representation in each channel. Consequently, the subsequent layer processes more fine-grained temporal features.

2) WEIGHTED BRANCH

An examination of the data presented in Table 2 shows that each branch makes different contributions to the performance. Thus, it is natural to ask whether the use of a weighted scheme per branch can lead to an increase of model performance. We performed several ablation experiments to determine the optimal combination of weights per branch. We weight each input of the first TGM layer, in both the spatial and temporal branches. We changed the values randomly and found the optimal combination of weights. The results are described in Table 3. Even though the difference is marginal, the results confirm that each branch contributes to the performance unequally, in agreement with the results presented in Table 2.

F. ANALYSIS

In this section, we discuss how well the proposed models classify the frames into the predefined action classes. To accomplish this, we plot the prediction into the temporal region with the X-axis being the time axis (see Figure 8).

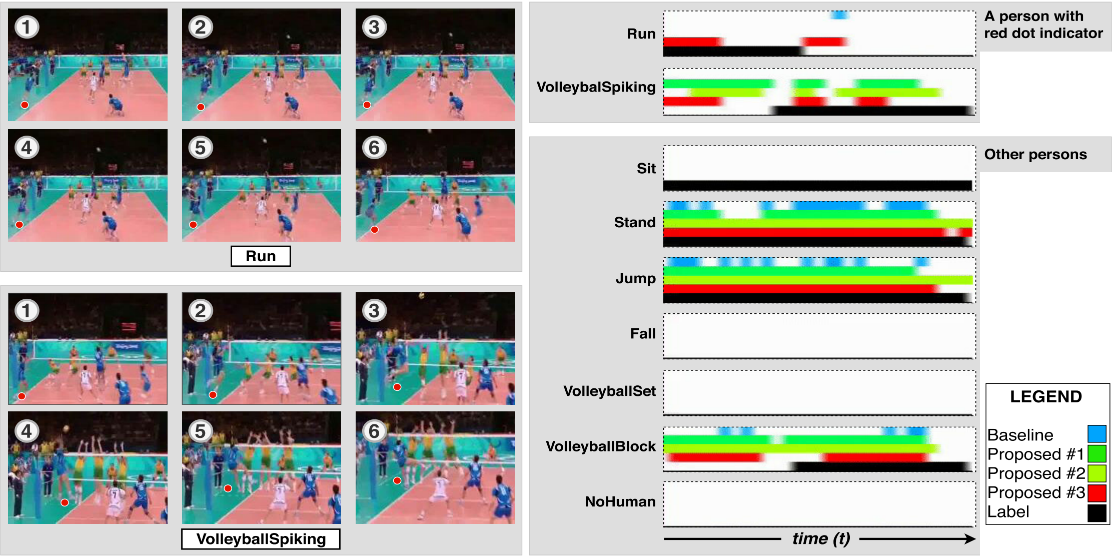

FIGURE 8. An example of various activities in a video of a volleyball match. There are five temporal regions to illustrate the lifespan of an activity. Temporal regions in black and blue show the ground truth (label) and the baseline model, respectively. The other colors show the proposed models. We show video images that describe the two actions "Run" and "VolleyballSpiking". A red dot indicator is placed below a person performing a specific activity at time *t*. It is observed that interpolating semantic information between two branches is beneficial for improving the performance. Best viewed in color.

TABLE 3. Results for different weights for each branch. Proposed #3 is used to produce this result.

Method	mAP (%)
Equal distribution on spatial and temporal branches	37.09
Spatial: 1.25, temporal: 0.95	**37.13**
Spatial: 0.95, temporal: 1.25	37.00

TABLE 4. Number of parameters of our proposed models compared to other work.

Model	# of Parameters
LSTM [6]	10.50M
1 Temporal Conv. [6]	10.50M
3 Temporal Conv. [6]	31.50M
Proposed #1	2.17M
Proposed #2	2.17M
Proposed #3	2.17M

This figure enables us to determine the accuracy of each proposed model by comparing the nonblack regions (the proposed models) with the black region (the ground truth) in the video of vollyball game. It is observed that our proposed models exhibit good performance in some activities. We plot several video frames to describe the activities of ''Run'' and ''VolleyballSpiking''.

In the *Run* activity, surprisingly, our proposed models greatly improve the performance while the baseline model fails to predict the activity. Figure 8 shows that a person with a red dot is performing *Run* before jumping and hitting the ball. In the right-hand layout, our proposed models correctly predict this activity; the red temporal region stretches for some time whereas the baseline model misses the prediction.

We also show the advantage of fusing the TGM layers in *VolleyballSpiking*. While the baseline model fails to accomplish the prediction, our proposed models enjoy the benefits of the fusing mechanism that can locate and classify the spiking activity in video frames. Again, a man with a red dot indicator is carrying out a volleyball spike. At the same time, this man is also performing the *Jump* activity. Our models predict *Jump* activity with a higher confidence level compared to the base model: the temporal regions are drawn continuously for our proposed models.

Despite successful predictions, we also observed the occurrence of misprediction. As shown for the *Sit* activity, none of the models can classify this activity even when it is present in reality. We suspect that because no changes are occurring temporally (i.e., the object/person is motionless), our proposed models have difficulty in learning its temporal structure. Furthermore, it is possible that the object is relatively too small for detection purposes and thus that our proposed models fail to notice it. We note that a false-positive prediction also occurred, as was found for the *VolleyballSpiking* and *VolleyballBlock* activities. In earlier times of these activities, all of the models predict a non-existence action, in contrast with the prediction after some *t* time.

We also observe that all of the proposed models are accurate in *Fall*, *VolleyballSet*, and *NoHuman* class. Additionally, we are confident that the fusion of two modalities, regardless of how they are fused, exhibits very good performance compared to the baseline. This suggests that it will be beneficial to explore other fusion techniques (e.g., temporal fusion, as mentioned earlier).

Furthermore, we examined the number of learnable parameters and compared it with other works. Table 4 indicates that three branches (spatial, temporal, and spatio-temporal) with three stacked TGM layers have notably fewer parameters; therefore, it is a lightweight layer.

V. CONCLUSION

In this paper, we proposed new unified multi-branch neural network models consisting of a sequence of TGM layers that serve as a spatial, temporal, and spatio-temporal branches for performing multi-label multi-class classification tasks. Outputs from the spatial and temporal branches are interpolated to form the spatio-temporal branch, with only a few learnable parameters added. We also demonstrated the benefit of using the maximum function inside the TGM layer to combine the input channels. The experimental results have shown that our proposed fusion strategies with spatio-temporal model learn temporal structure effectively, ultimately improving the activity detection performance and outperforming the baseline and several other designs on the MultiTHUMOS dataset.

For future work, as discussed in section II. Related Work, we are interested in fusing the two branches temporally, as demonstrated in [3], where the authors claim that a temporal fusion strategy can achieve larger gains in the model performance.

REFERENCES

[1] B. Jiang, M. Wang, W. Gan, W. Wu, and J. Yan, "STM: SpatioTemporal and motion encoding for action recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*. Piscataway, NJ, USA: Institute Electrical and Electronics Engineers Inc., Oct. 2019, pp. 2000–2009.

[2] D. He, Z. Zhou, C. Gan, F. Li, X. Liu, Y. Li, L. Wang, and S. Wen, "StNet: Local and global spatial-temporal modeling for action recognition," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, Jul. 2019, pp. 8401–8408.

[3] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," 2016, *arXiv:1604.06573*. [Online]. Available: http://arxiv.org/abs/1604.06573

[4] W. Dai, Y. Chen, C. Huang, M.-K. Gao, and X. Zhang, "Two-stream convolution neural network with video-stream for action recognition," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Piscataway, NJ, USA: Institute Electrical Electronics Engineers Inc., Jul. 2019, pp. 1–8.

[5] J. You, P. Shi, and X. Bao, "Multi-stream I3D network for fine-grained action recognition," in *Proc. IEEE 4th Inf. Technol. Mechatronics Eng. Conf. (ITOEC)*, Piscataway, NJ, USA: Institute Electrical Electronics Engineers Inc., Dec. 2018, pp. 611–614.

[6] A. J. Piergiovanni and M. S. Ryoo, "Temporal Gaussian mixture layer for videos," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2019, pp. 5152–5161.

[7] J. Carreira and A. Zisserman, "Quo Vadis, action recognition? A new model and the kinetics dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4724–4733. [Online]. Available: http://ieeexplore.ieee.org/document/8099985/

[8] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*. Washington, DC, USA: IEEE Computer Society, Jun. 2016, pp. 2818–2826.

[9] T. B. Moeslund, A. Hilton, and V. Krüger, "A survey of advances in vision-based human motion capture and analysis," *CVIU*, vol. 104, nos. 2–3, pp. 90–126, 2006.

[10] H.-B. Zhang, Y.-X. Zhang, B. Zhong, Q. Lei, L. Yang, J.-X. Du, and D.-S. Chen, "A comprehensive survey of vision-based human action recognition methods," *Sensors*, vol. 19, no. 5, p. 1005, Feb. 2019. [Online]. Available: http://www.mdpi.com/1424-8220/19/5/1005

[11] A. F. Bobick and J. W. Davis, "The recognition of human movement using temporal templates," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 3, pp. 257–267, Mar. 2001.

[12] A. Klaeser, M. Marszalek, and C. Schmid, "A spatio-temporal descriptor based on 3D-gradients," in *Proc. Brit. Mach. Vis. Conf.*, 2008, pp. 99.1–99.10. [Online]. Available: http://www.bmva.org/bmvc/2008/papers/275.html

[13] P. Scovanner, S. Ali, and M. Shah, "A 3-dimensional sift descriptor and its application to action recognition," in *Proc. 15th Int. Conf. Multimedia*. New York, NY, USA: ACM, 2007, pp. 357–360. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1291233.1291311

[14] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local SVM approach," in *Proc. 17th Int. Conf. Pattern Recognit.*, vol. 3, Aug. 2004, pp. 32–36.

[15] X. Ben, C. Gong, P. Zhang, X. Jia, Q. Wu, and W. Meng, "Coupled patch alignment for matching cross-view gaits," *IEEE Trans. Image Process.*, vol. 28, no. 6, pp. 3142–3157, Jun. 2019.

[16] X. Ben, P. Zhang, Z. Lai, R. Yan, X. Zhai, and W. Meng, "A general tensor representation framework for cross-view gait recognition," *Pattern Recognit.*, vol. 90, pp. 87–98, Jun. 2019.

[17] G. Zhu, C. Xu, Q. Huang, and W. Gao, "Action recognition in broadcast tennis video," in *Proc. 18th Int. Conf. Pattern Recognit. (ICPR)*, vol. 1, Aug. 2006, pp. 251–254.

[18] A. Yao, J. Gall, and L. Van Gool, "A Hough transform-based voting framework for action recognition," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2061–2068.

[19] K. Guo, P. Ishwar, and J. Konrad, "Action recognition using sparse representation on covariance manifolds of optical flow," in *Proc. 7th IEEE Int. Conf. Adv. Video Signal Based Surveill.* Washington, DC, USA: IEEE Computer Society, Aug. 2010, pp. 188–195.

[20] M. Gholamrezaii and S. M. T. Almodarresi, "Human activity recognition using 2D convolutional neural networks," in *Proc. 27th Iranian Conf. Electr. Eng. (ICEE)*. Piscataway, NJ, USA: Institute Electrical Electronics Engineers Inc., Apr. 2019, pp. 1682–1686.

[21] Z. Qiu, T. Yao, and T. Mei, "Learning spatio-temporal representation with pseudo-3d residual networks," in *Proc. IEEE Int. Conf. Comput. Vis.* Piscataway, NJ, USA: Institute Electrical Electronics Engineers Inc., Dec. 2017, pp. 5534–5542. [Online]. Available: http://arxiv.org/abs/1711.10305

[22] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Jan. 2013.

[23] J. Arunnehru, G. Chamundeeswari, and S. P. Bharathi, "Human action recognition using 3D convolutional neural networks with 3D motion cuboids in surveillance videos," *Procedia Comput. Sci.*, vol. 133, pp. 471–477, Jan. 2018.

[24] H. Yang, C. Yuan, B. Li, Y. Du, J. Xing, W. Hu, and S. J. Maybank, "Asymmetric 3D convolutional neural networks for action recognition," *Pattern Recognit.*, vol. 85, pp. 1–12, Jan. 2019.

[25] K. Hara, H. Kataoka, and Y. Satoh, "Learning spatio-temporal features with 3D residual networks for action recognition," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*. Piscataway, NJ, USA: Institute Electrical Electronics Engineers Inc., Oct. 2017, pp. 3154–3160.

[26] N. Dhingra and A. Kunz, "Res3ATN–deep 3D residual attention network for hand gesture recognition in videos," in *Proc. Int. Conf. 3D Vis. (3DV)*. Piscataway, NJ, USA: Institute Electrical Electronics Engineers Inc., Sep. 2019, pp. 491–501.

[27] A. Stergiou and R. Poppe, "Spatio-temporal FAST 3D convolutions for human action recognition," in *Proc. 18th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*. Piscataway, NJ, USA: Institute Electrical Electronics Engineers Inc., Dec. 2019, pp. 183–190.

[28] C.-Y. Ma, M.-H. Chen, Z. Kira, and G. AlRegib, "TS-LSTM and temporal-inception: Exploiting spatiotemporal dynamics for activity recognition," *Signal Process., Image Commun.*, vol. 71, pp. 76–87, Feb. 2019.

[29] A. Ullah, J. Ahmad, K. Muhammad, M. Sajjad, and S. W. Baik, "Action recognition in video sequences using deep bi-directional LSTM with CNN features," *IEEE Access*, vol. 6, pp. 1155–1166, 2018. [Online]. Available: http://ieeexplore.ieee.org/document/8121994/

[30] Z. Wu, Y.-G. Jiang, X. Wang, H. Ye, and X. Xue, "Multi-stream multi-class fusion of deep networks for video classification," in *Proc. ACM Multimedia Conf. (MM)*. New York, NY, USA: ACM, 2016, pp. 791–800. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2964284.2964328

[31] Y. Y. Joefrie and M. Aono, "Action recognition by composite deep learning architecture I3D-DenseLSTM," in *Proc. Int. Conf. Adv. Inform., Concepts, Theory Appl. (ICAICTA)*, Sep. 2019, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/8904245/

[32] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.*, vol. 1. Cambridge, MA, USA: MIT Press, 2014, pp. 568–576.

[33] Y. Cai, W. Lin, J. See, M.-M. Cheng, G. Liu, and H. Xiong, "Multi-scale spatiotemporal information fusion network for video action recognition," in *Proc. IEEE Vis. Commun. Image Process. (VCIP)*. Piscataway, NJ, USA: Institute Electrical Electronics Engineers Inc., Dec. 2018, pp. 1–4.

[34] L. Chi, G. Tian, Y. Mu, and Q. Tian, "Two-stream video classification with cross-modality attention," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*. Piscataway, NJ, USA: Institute Electrical Electronics Engineers (IEEE), Mar. 2020, pp. 4511–4520.

[35] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," 2012, *arXiv:1212.0402*. [Online]. Available: http://arxiv.org/abs/1212.0402

[36] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "SlowFast networks for video recognition," 2018, *arXiv:1812.03982*. [Online]. Available: http://arxiv.org/abs/1812.03982

[37] S. Yeung, O. Russakovsky, N. Jin, M. Andriluka, G. Mori, and L. Fei-Fei, "Every moment counts: Dense detailed labeling of actions in complex videos," *Int. J. Comput. Vis.*, vol. 126, nos. 2–4, pp. 375–389, Apr. 2018.

[38] C. Zach, T. Pock, and H. Bischof, "A duality based approach for realtime TV-L1 optical flow," in *Pattern Recognition* (Lecture Notes in Computer Science: Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 4713. Berlin, Germany: Springer, 2007, pp. 214–223.

[39] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent.*, Dec. 2015, pp. 1–15.

[40] A. Dave, O. Russakovsky, and D. Ramanan, "Predictive-corrective networks for action detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*. Piscataway, NJ, USA: Institute Electrical Electronics Engineers Inc., Jul. 2017, pp. 2067–2076.

[41] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, X. Tang, and D. Lin, "Temporal action detection with structured segment networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*. Piscataway, NJ, USA: Institute Electrical Electronics Engineers Inc., Oct. 2017, pp. 2933–2942.

YURI YUDHASWANA JOEFRIE received the B.Sc. degree from the Institut Teknologi Nasional (ITN), Malang, Indonesia, in 2002, and the M.Eng. degree from the Institut Teknologi Bandung (ITB), Bandung, Indonesia, in 2008. He is currently pursuing the Ph.D. degree with the Toyohashi University of Technology (TUT), Toyohashi, Aichi, Japan. He has also been a Lecturer with Tadulako University, Palu, Indonesia, since 2009. His research interests include computer vision, action recognition, and video understanding.

MASAKI AONO (Member, IEEE) received the B.S. and M.S. degrees from the Department of Information Science, The University of Tokyo, Tokyo, Japan, and the Ph.D. degree from the Department of Computer Science, Rensselaer Polytechnic Institute, New York. He was with IBM Tokyo Research Laboratory from 1984 to 2003. He is currently a Professor with the Computer Science and Engineering Department, Graduate School, Toyohashi University of Technology. His research interests include text and data mining for massive streaming data and information retrieval for multimedia, including 2D images, videos, and 3D shape models. He is a member of ACM and the IEEE Computer Society. He has been a Japanese Delegate with the ISO/IEC JTC1 SC24 Standard Committee since 1996.