IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 3 February 2023, accepted 23 March 2023, date of publication 31 March 2023, date of current version 7 April 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3263812

RESEARCH ARTICLE

Convolutional Transformer Fusion Blocks for Multi-Modal Gesture Recognition

BASAVARAJ HAMPIHOLI[1,2], CHRISTIAN JARVERS[2], WOLFGANG MADER[1], AND HEIKO NEUMANN[2]

[1]BMW Car IT GmbH, 89081 Ulm, Germany
[2]Institute of Neural Information Processing, Ulm University, 89081 Ulm, Germany

Corresponding author: Basavaraj Hampiholi (basavaraj.hampiholi@uni-ulm.de)

ABSTRACT Gesture recognition defines an important information channel in human-computer interaction. Intuitively, combining inputs from multiple modalities improves the recognition rate. In this work, we explore multi-modal video-based gesture recognition tasks by fusing spatio-temporal representation of relevant distinguishing features from different modalities. We present a self-attention based transformer fusion architecture to distill the knowledge from different modalities in two-stream convolutional neural networks (CNNs). For this, we introduce convolutions into the self-attention function and design the Convolutional Transformer Fusion Blocks (CTFB) for multi-modal data fusion. These fusion blocks can be easily added at different abstraction levels of the feature hierarchy in existing two-stream CNNs. In addition, the information exchange between two-stream CNNs along the feature hierarchy has so far been barely explored. We propose and evaluate different architectures for multi-level fusion pathways using CTFB to gain insights into the information flow between both streams. Our method achieves state-of-the-art or competitive performance on three benchmark gesture recognition datasets: a) IsoGD, b) NVGesture, and c) IPN hand. Extensive evaluation demonstrates the effectiveness of the proposed CTFB both in terms of recognition rate as well as resource efficiency.

INDEX TERMS Self-attention, transformer, gesture recognition, multi-modal fusion.

I. INTRODUCTION

Multiple sensors provide diverse and complementary information about the same object, scene or action. RGB images are rich in color and texture information, depth maps contain geometric shape cues and infrared images can be captured under different illumination conditions. Integration of multiple such modalities each with high-level semantic features results in a descriptor that leads to robust and reliable recognition. Multi-modal learning has proven to be successful in many machine learning tasks, such as object recognition, scene understanding, action recognition, and audio-visual interaction. Our work focuses on gesture recognition using multiple modalities such as RGB, depth, and optical flow.

Gesture recognition has several applications in entertainment, autonomous vehicles, gaming, etc. For example, in the context of intermediate levels of autonomous driving, recognizing the gestures of a driver can be important to enable seamless interaction with the vehicle and increase safety. The interaction has to be robust and reliable and needs to function under different conditions independent of weather, illumination, and varying cluttered background. Multi-modal learning can help to achieve this robustness, by fusing complementary features from each modality.

Multi-modal fusion predominantly focuses on (a) feature extraction and representation for each modality, (b) how the information from each sensor stream needs to be fused, and (c) determining which level of the model is suitable for fusion. Earlier methods used traditional (handcrafted) image processing algorithms for feature extraction. In more recent work, deep neural networks are utilized as common feature extractors for different modalities. In general, two-stream or multi-stream networks are employed to learn multi-modal fusion where each stream computes the features for one modality. In this work, we use two-stream networks to extract the features from gesture videos of two different modalities.

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

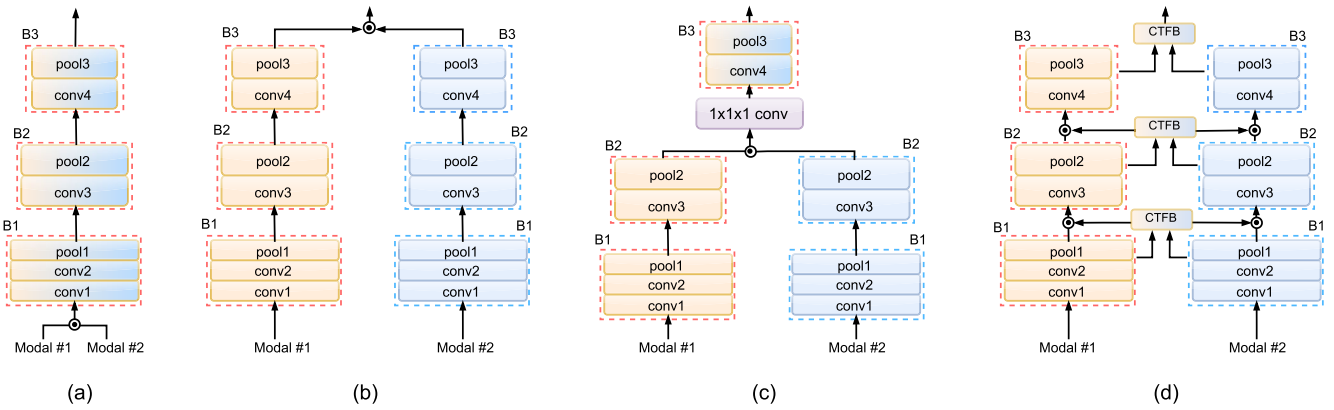

FIGURE 1. Illustration of different fusion techniques: (a) early fusion (b) late fusion (c) mid fusion (d) multi-level fusion. Each stream represents a CNN model with a hierarchical structure. The yellow stream takes modality1 as input while the blue stream takes modality2 as input. B1, B2, and B3 blocks contain a series of convolution (conv) layers with an activation function followed by pooling (pool) layers. Here, the $1 \times 1 \times 1$ conv layer is used for mid fusion. Our proposed CTFB is used for multi-level fusion. ◉ denotes the fusion operations such as addition, multiplication, or concatenation.

The primary challenge of multi-modal deep learning is the fusion of different modalities to exploit complementary cues to generate robust representations. Many existing methods fuse information through various operations, such as addition, averaging, multiplication [9], concatenation [10] or gating mechanisms [15]. Recently, transformer-based fusion networks [5], [17], [24] have been proposed to fuse multiple modalities such as lidar-images, video-text, video-speech, text-speech, etc. The self-attention operation used in these fusion architectures can learn and exploit global dependencies but is less sensitive to the local neighborhood structure that exists in images or videos. However, Convolution operations are specialized in capturing local neighborhood pixels from a grid structure in images or videos. It is important to encode both the local and global spatio-temporal features for various tasks, for example, video-based multi-modal hand gesture recognition tasks.

We design an Efficient Convolutional Self-Attention (ECSA) module that aims to compute attention maps on the local features. For this, we modify the standard self-attention block in [43] by replacing fully connected layers in it with 3D depthwise separable convolutions to capture the local spatio-temporal structure in videos. A multi-layered perceptron (MLP) captures the global dependencies between the attention maps generated by the ECSA module. We propose a new block called Convolutional Transformer Fusion Block (CTFB) that assembles ECSA and MLP modules in a specific order to fuse multiple modalities and captures both local and global representations from the fused information. CTFBs can be added straightforwardly at different hierarchical levels of existing unimodal CNN architectures. In this work, we use CTFB to study the interaction between two-stream CNNs that take two different modalities as inputs.

Another challenging question is to identify which level of CNN models is suitable to fuse the information. There have been several attempts in fusing multi-modal information at different levels: (a) early fusion [16], (b) late fusion, (c) mid fusion [25], and (d) multi-level fusion [24], [38]. Figure 1 schematically depicts all these fusion schemes. Early fusion combines features at the input level and feeds them to a common predictor. But it assumes that a single predictor is sufficient to capture the representation from all modalities. Late fusion [4], [27] integrates the predictions from individual models. Although late fusion uses individual predictors for each modality, it neglects the interaction between mid-level features of those predictors. Some methods have shown that mid fusion [25] is more effective because it captures the useful early correlations between the features of different modalities. However, it is difficult to choose one mid fusion point. Also, mid fusion neglects the interaction between features with varying coarseness. Despite several works [24], [30], the possibilities of information exchange between intermediate layers of two-stream CNNs have been barely investigated. In this work, we propose and evaluate different architectures using CFTB to explore multi-level fusion along the feature hierarchy (with varying coarseness) to explore the interaction between multiple modalities. The key contributions of this paper are as follows:

- We propose CTFB, a novel Convolutional Transformer Fusion Block, to integrate salient features from each modality.
- We design three different architectures for multi-level fusion pathways using CTFB and compare their performances. We refer them as 1) Shared Pathway, 2) Bidirectional Pathway, and 3) Central Pathway.

Our experiments demonstrate state-of-the-art or competitive results on benchmark gesture recognition datasets: a) IsoGD, b) NVGesture, and c) IPN hand.

II. RELATED WORK

A. TRANSFORMERS FOR VISION

The success of self-attention [43] in the natural language processing (NLP) community motivated researchers in the computer vision community to explore transformer architectures for vision tasks like object detection and recognition in images, segmentation, video action recognition, and many

more. Vision Transformer (ViT) [7] leads the way in this direction and proposes a pure transformer model without any convolutional layers to perform image classification. In this, an image is split into 2D patches, each of which is flattened into a 1D vector and fed to the transformer model. Hence it disregards the grid structure with neighbourhood pixels and only focuses on the global representation of image. ViT adopts the same self-attention function from [43]. Local ViT [18] focuses on learning local features by replacing an MLP module in transformers with the locality feedforward network. Some recent works [26], [33] focus on exploiting the benefits of both transformer nets and convolutional nets. Convolutional vision Transformer (CvT) [33] introduces convolutions into self-attention block [7], [43]. CvT [33] is a hierarchical structure that consists of a token embedding convolution layer and convolutional transformer block. But computational and memory costs of CvT [33] grow with the input size as it uses dot-product attention. An efficient transformer [26] addresses this issue by replacing dot-product attention with an efficient attention mechanism that has linear computational complexities and achieves performance similar to ViT [7] on image recognition tasks. Our CTFB designed using ECSA modules contains 3D depthwise separable convolution layers that allow capturing of salient local spatio-temporal representation.

B. MULTI-MODAL FUSION

This section provides insights into different multi-modal fusion techniques proposed previously. The bilinear learning framework [12] proposes the Bilinear block that consists of a modality pooling layer and a temporal pooling layer and aims to learn the combined representation of them. The GIF [15] network combines the intermediate RGB-D feature maps at all levels using fusion gates and weight generation networks. C3D-Stitch [25] proposed cross-stitch fusion units to exchange information between two C3D streams at various levels of feature hierarchy for multi-modal gesture recognition tasks. MMTM [30] adopts a Squeeze-Excitation (SE) module to recalibrate and fuse the channel-wise features from multiple modalities. The Channel Exchange Network (CEN) [34] dynamically exchanges the feature maps of each modality between the sub-networks. The Cross-Modality Attention (CMA) [6] block combines the features from RGB and flow networks using non-local attention operation. Recently transformer-based fusion architectures are in demand for multi-modal learning. HAMLET [14] is a multi-head attention-based fusion technique where features at different hierarchical levels are combined. MM-ViT [5] extracts the features of RGB frames, flows, and audio waveforms and computes self-attention across spatial, temporal, and different modalities. Transfuser [24] combines the output RGB and LiDAR features from each CNN block using a standard transformer module. Trear [17] proposes (a) an inter-frame transformer encoder to extract the attention-based features for each modality (RGB and depth), (b) mutual-attentional feature fusion combines these features and applies the cross-modality attention to learn the mutual interaction between modalities. Our fusion strategy captures the salient local representation of individual modality by computing attention maps using convolutions and also the global representation of fused features using a multi-layered perceptron (MLP) module.

III. MULTI-MODAL CONVOLUTIONAL TRANSFORMER FUSION

This section provides the detailed architectural design of the transformer-based fusion block for multi-modal learning. Our fusion network is built on a baseline two-stream I3D network [4] to train a pair of modalities. We choose different combinations of two modalities among RGB videos, optical flows, depth videos, and segmentation masks for our experiments. The overall architectural details of our two-stream fusion network are provided in the supplementary material.

A. PRELIMINARY: EFFICIENT ATTENTION

Dot-product attention in standard self-attention block [43] is computationally resource intensive and grows quadratically with the input size. The efficient attention mechanism proposed in [26] exploits the associative property of the matrix multiplication in dot-product attention and reorders the multiplication operations based on the observation that $(QK^T)V = Q(K^TV)$. This technique reduces the computational complexity from $O(n^2)$ to $O(d_k \times d_v)$ where 'n' is the size of the input sequence and d_k, d_v are the feature dimensions of the key (K) and values (V). Query (Q) also has the same feature dimension as key, i.e., d_k. In a CNN architecture, the feature dimensions d_k, d_v generally are of small size and already known. Therefore, the time complexity of the operation of the order $d_k \times d_v$ is less expensive compared to the order of n^2. Dot-product attention is computed between pairwise positions of all input elements (n^2), whereas efficient attention interprets a number of feature maps d_k as global attention maps that do not correspond to any position, instead, they represent the semantic aspect of the entire input. Experiments conducted with the efficient attention mechanism in [26] show that it yields similar/better performance than dot-product attention on action localization, object detection, and instance segmentation tasks.

B. EFFICIENT CONVOLUTIONAL SELF-ATTENTION

Initial convolutional layers of the I3D [4] model output high-resolution spatio-temporal features and hence produce very long sequences. For an input video with 64 frames and 224×224 pixels of spatial resolution, the B1 block in I3D produces the sequence size 'n' of $[32 \times 56 \times 56$ $(T \times H \times W) = 100, 352]$ (see Figure S2 in the supplementary material). This is a very long sequence and hence it is resource intensive to compute the attention maps using dot-product attention for low-level features of the I3D model. On the other hand, the standard self-attention block contains fully connected layers that compute features for queries, keys, and values. Vision transformers (ViT) [7] contain a series of such self-attention

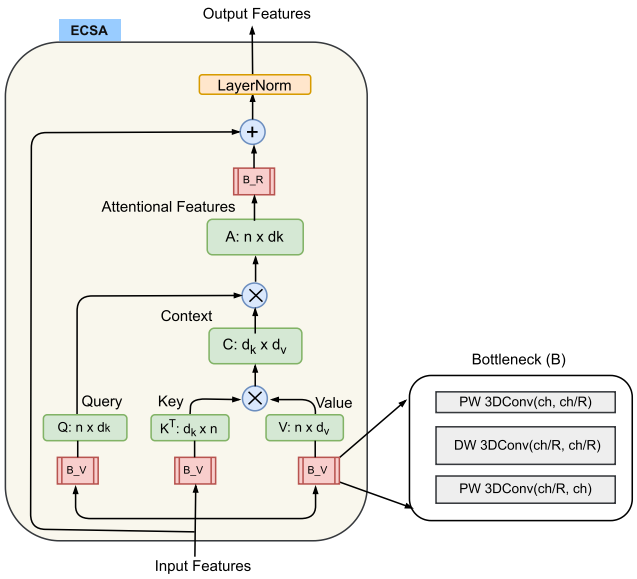

FIGURE 2. Efficient Convolutional Self-Attention (ECSA). B_Q, B_K, B_V, B_R (red boxes) are bottleneck blocks each composed of three layers of pointwise (PW) and depthwise (DW) convolutions (as shown in zoomed inset). These layers are parameterized to steer their operation. Here, *ch* is the number of channels, *R* is the reduction ratio, *n* is the input sequence length, and d_k, d_v are the dimensionalities of key (K) and value (V). All the operations required in the computation of self-attention are represented using green boxes. Each box is labeled with the name of a matrix, and inside the box are listed the names of the variables and their corresponding matrix sizes. In the end, we use layer normalization operation. ⊗ denotes matrix multiplication and ⊕ is a residualship connection.

blocks with the intent to capture the global representation of images but disregard local features. Furthermore, ViTs lack the inherent properties of convolutional architectures such as local receptive fields, translational equivariance, and shared weights.

We design the Efficient Convolutional Self-Attention (ECSA) module by making two changes to the standard self-attention block from [7] and [43]. First, to address the limitation of dot-product attention, we use the above-mentioned efficient attention [26] mechanism to compute attention maps. Second, we replace fully connected layers in the self-attention block with convolution-based bottlenecks. The bottleneck blocks B_Q, B_K, B_V (similar to ResNet [11] block) compute features for Q, K, and V. Each bottleneck block consists of three convolution layers. First, a pointwise convolution layer (kernel size $= 1 \times 1 \times 1$) reduces the number of channels with reduction ratio R. Normally, we use R=4. Next, we use a 3D depthwise (DW) convolution layer that applies a single filter per input channel. Finally, another pointwise (PW) convolution layer combines the outputs from depthwise convolution and expands the number of channels to their original size. The design choice of using depthwise separable convolutions in a bottleneck type of architecture helps to reduce the number of model parameters. Additionally, the bottleneck blocks help ECSA to capture salient local features. In this way, ECSA exploits the benefits of both attention mechanisms and convolutional nets. The ECSA module is depicted in Figure 2.

The ECSA module computes attention maps for spatio-temporal features along the I3D [4] feature hierarchy. First, we provide feature maps from I3D layers as a common input to three different 3D bottleneck blocks B_Q, B_K, B_V. It is then followed by an efficient attention mechanism [26] to compute attention maps. Similar to ViT [7], we use multi-head attention where each head weighs the importance of different parts of input features. The output of multi-head attention is reprojected using another 3D depthwise separable convolution layer. This stage is followed by the residual connection and layer normalization.

C. CONVOLUTIONAL TRANSFORMER FUSION BLOCK

We propose a Convolutional Transformer Fusion Block (CTFB) which is designed using an ECSA module, a fusion operation (fuse_op), and an MLP module. We discussed ECSA in the above section. Fusion operations can be addition, multiplication, concatenation, etc. The MLP module consists of two fully connected layers with ReLU activation in between and a dropout layer. The basic idea behind the multi-modal fusion using our proposed CTFB is to extract the salient features from each modality along multiple levels of the feature hierarchy in I3D and then fuse them to have useful discriminative representation for recognition tasks.

D. MULTI-LEVEL FUSION SCHEMES

The feature interaction between two-stream networks using multi-level fusion has been barely studied in the past. In this section, we focus on the architectural designs of three different schemes for multi-level fusion. We refer to them as 1) Shared Pathway, 2) Bidirectional Pathway, and 3) Central Pathway. These schemes provide insights into the information flow between bimodal two-stream networks. Figure 3 illustrates the different fusion schemes using CTFB. For brevity, we choose only two blocks B1 and B2 that contain segregated layers of the I3D stream (see Figure S1 in the supplementary material to know how the layers are segregated into blocks). In the following we explain the details of each of the above-mentioned schemes.

1) SHARED PATHWAY

In the shared pathway network setup (Figure 3 (a)), there are two ECSA modules each responsible for computing attention maps for both modalities separately. The output attention maps from each ECSA (specific to each modality) are then fused using an element-wise addition operation. The resultant features are passed to an MLP module that computes the position-wise relationship among the fused features. Therefore, the output features of the MLP module have a shared representation of both modalities. The fused representations from MLP are added back to each modal stream. The output of the final CTFB is passed to a fully connected layer for classification. In this scheme, the information is flowing back and forth between the fusion block and each backbone network.

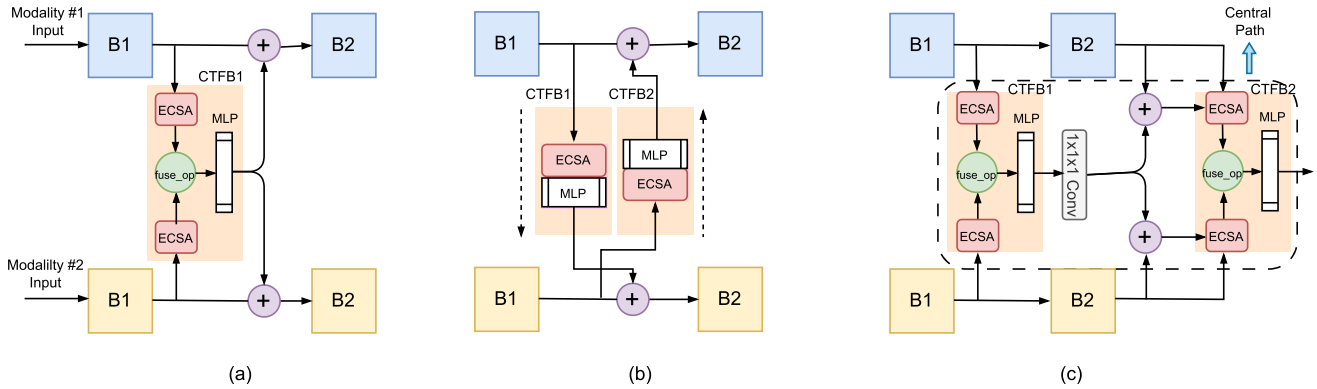

FIGURE 3. Design choices of proposed multi-level fusion pathways: (a) Shared Pathway (b) Bidirectional Pathway (c) Central Pathway. For brevity, here, we use only two blocks B1 and B2 that contain segregated layers of I3D. A stream with blue boxes learns modality1 and another stream with yellow boxes learns modality2. CTFB is our proposed fusion block. `fuse_op` denotes fusion operations such as addition, multiplication, and concatenation to combine the different information streams. $\oplus$ denotes an elementwise addition operation. $1 \times 1 \times 1$ is a strided convolution with stride=2.

2) BIDIRECTIONAL PATHWAY

Figure 3 (b) displays the bidirectional pathway scheme. It contains two individual CTFBs along each direction. Each block consists of an ECSA module followed by an MLP module and there is no fusion operation (fuse_op) in between. First, CTFBs compute the local salient features of each modality separately using ECSA followed by an MLP module to capture global dependencies among those modality-specific features. The computed features are then added to the stream of the opposite modality. For example, the salient features of RGB modality using CTFB1 are merged with the depth stream and vice-versa. In the end, we take the average of the output features of both streams and feed it to a fully connected layer for recognition tasks.

3) CENTRAL PATHWAY

This strategy is similar to the shared pathway network except that the information is not added back to the backbone stream. Instead, features coming from both modalities are projected on to a common space. In the end, this joint representation is used for classification. Figure 3 (c) presents the central pathway strategy. In this, the output attention maps of CTFB1 are fed into $1 \times 1 \times 1$ convolution layer with stride $= 2$ to adjust their dimensions in order to match the feature (channel) and spatio-temporal dimensions of the subsequent layers of I3D. The output features of B2 are added with the output features from the $1 \times 1 \times 1$ convolution layer and passed as input to CTFB2. This structure repeats multiple times along I3D hierarchy and the outputs of last CTFB is fed into fully connected layer for classification.

IV. EXPERIMENTAL RESULTS

A. GESTURE RECOGNITION DATASETS

In this section, we report about experiments conducted on three benchmark gesture recognition datasets: (a) NVGesture [21], (b) IsoGD [31], and (c) IPN hand gestures [3]. We compare our results with the state-of-the-art (SoTA) methods. Figure 4 shows frames of a sample gesture video frames of different modalities from each dataset. Gestures recognition can be either continuous or isolated. In continuous gesture recognition tasks, an input video clip contains a series of various gestures. On the other hand, in isolated gesture recognition tasks, an input video contains a single gesture. This work mainly focuses on the latter part.

1) IsoGD DATASET

Isolated gestures (IsoGD) [31] is a large-scale multi-modal gesture recognition dataset derived from the Chalearn gesture dataset (CGD 2011). IsoGD contains gesture videos of two different modalities viz. RGB and depth with the resolution of 320×240 pixels and a frame rate of 10 fps. The dataset is user independent, which means the participants in the training set are not repeated in the test set. In total, the dataset includes 47,933 RGB-D gesture videos with 249 gesture labels. The gestures were performed by 21 different individuals. The dataset is split into a training set with 35,878 samples from 17 participants, a validation set with 5,784 samples from 2 participants, and a test set with 6,271 samples from 2 participants.

2) NVGesture DATASET

NVGesture [21] is an in-car dynamic hand gesture recognition dataset captured from multiple viewpoints using multiple sensors. NVGesture contains gesture videos of three different modalities viz. RGB, depth, and infrared (IR). The videos are recorded at the rate of 30fps and with a resolution of 320×240 pixels. The dataset consists of 1,532 gesture videos with 25 different classes of hand gestures. The dataset is split into a training set with 1,050 samples and a test set with 482 samples. The gestures were captured from 20 different individuals. IR videos do not have the same viewpoint as RGB and depth videos. Therefore, we use only RGB and depth modalities for our experiments.

3) IPN HAND DATASET

The IPN hand dataset [3] focuses on gestures that are relevant to interaction with touchless screens. It contains RGB videos with a resolution of 640×480 pixels recorded at 30fps using

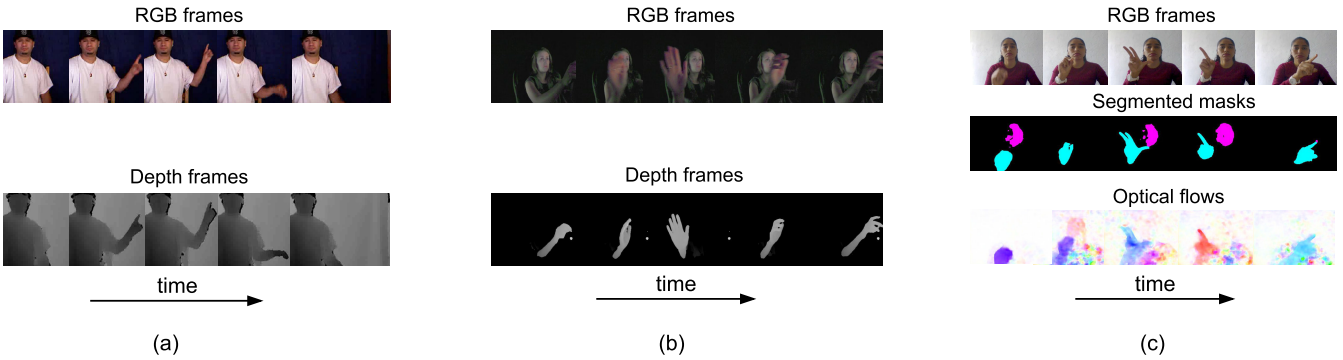

FIGURE 4. A sequence of frames (video) depicting the gesture of a person from benchmark datasets: (a) IsoGD [31] (b) NVGesture [21] (c) IPN Hand [3]. Videos are represented in different modalities such as RGB, depth maps, optical flows, and segmentation masks.

PC or laptop cameras. The videos are captured from 28 different scenes with 50 participants. These scenes include cluttered backgrounds and varying illumination. In total, there are 4,218 gesture instances with 13 different gesture classes. The dataset is slightly imbalanced with a majority of the samples belonging to only 2 classes. It is intended for both isolated and continuous gesture recognition tasks. We focus only on isolated gesture recognition. The dataset is randomly split into training (74%) and testing (26%) sets. For isolated gestures, the training set contains 3,117 gesture instances from 37 subjects and the test set contains 1,101 gesture instances from 13 subjects. This dataset provides RGB videos, optical flows, and segmented masks for each gesture instance. We use a combination of these modalities to train the two-stream I3D with the fusion module.

B. IMPLEMENTATION DETAILS

We adopt I3D [4] as a backbone model for our video-based multi-modal gesture recognition. The I3D model is the inflated version of inception v1 [13], [28] architecture. The main idea behind inception v1 architecture is that inception modules can capture multi-scale information by using parallel convolution layers with different kernel sizes. We group the I3D inception modules into blocks at every max pool layer to mark the fusion points at different levels of the hierarchy (see Figure S1 in the supplementary material for details). We use the publicly available pre-trained weights on Imagenet [36] + Kinetics [4] to initialize the weights of I3D in our experiments. We implement and train our models using the PyTorch framework and a V100 GPU.

During the training phase, we resize the input video clips to 256×256 pixels and then spatially crop them randomly with a patch size 224×224. We randomly select 64 consecutive frame snippets containing gestures from the input video. For shorter videos, we loop the video to have uniform inputs. In the end, the shape of all input video clips is $batchsize \times 3 \times 64 \times 224 \times 224$ where batchsize is 4. During the testing phase, we follow similar preprocessing steps as training phase, except that a center-crop technique is applied instead of random crop.

The training involves a two-stage process. In the first stage, the individual models are trained for each modality (RGB, depth, or flows) with the base learning rate $\eta = 0.05$. In the second stage, the fusion module is inserted at different levels of I3D with pre-trained weights of each modality and trained in an end-to-end fashion with the base learning rate $\eta = 0.005$. We employ a learning schedule to change the learning rate over time. We train the network for a total of 30 epochs and downscale the learning rate by 10% (multiplied by 0.1) after the 15^{th} and 25^{th} epoch. We use the stochastic gradient descent (SGD) optimizer with the Nesterov momentum 0.9 and weight decay of 10^{-4}. We optimize the cross-entropy loss for the multi-label classification of gestures and use accuracy as an evaluation metric. In all of the following experiments, we employed the same above-mentioned configurations for the training and evaluation phases.

C. COMPARISON BETWEEN EARLY, MID, LATE, OR MULTI-LEVEL FUSION

We conduct experiments to investigate which part of the CNN model hierarchy is suitable to fuse the multi-modal features using the NVGesture dataset [21]. Figure 1 illustrates different levels of fusion. Table 1 presents the results of individual modalities trained with the I3D network [4] separately and also of multi-modal fusion. We observe that the accuracy of early fusion with RGB+depth input is hardly better than single stream I3D trained on depth videos. Late fusion achieves a notable improvement over early fusion with a margin of 2%. Mid fusion can be performed at any level of the feature hierarchy. In our experiments, we use $1 \times 1 \times 1$ convolution as a mid fusion layer after the third block (B3) in two-stream I3D (see Figure S3 in the supplementary material). This setting improves the late fusion instances by 1%. These results suggest that capturing correlations among early features boosts the performance of the model. Finally, we perform multi-level fusion using only CTFB3 and CTFB4 in a two-stream I3D network (see Figure S2 in the supplementary material). Our results show that multi-level fusion surpasses the test accuracy of all other above-discussed fusion strategies. These outcomes indicate that feature interactions at multiple levels

TABLE 1. Where to Fuse? Early, Late, Mid, or Multi-level fusion. NVGesture dataset [21] is used for this experiment. First part contains average test accuracy of individual models and second part contains average test accuracy of different fusion techniques. R, D denote RGB, depth modalites.

Method	Modalities	Test Acc
I3D [4]	R	78.42
I3D [4]	D	82.28
Early	R+D	82.54
Late	R+D	84.43
Mid	R+D	85.27
Multi-level	R+D	**87.76**

TABLE 2. Performance of different fusion pathways: Average test accuracy on IsoGD [31] RGB (R) and depth (D) fusion. Number of model parameters are in millions (M).

Method	Modalities	Test Acc	#Params
I3D [4]	R	63.81	12.3M
I3D [4]	D	63.56	12.3M
Bidirectional	R+D	65.44	41.1M
Central	R+D	66.94	37.7M
Shared	R+D	**69.02**	**37.1M**

(from fine to coarse-grained) between streams can encode additional contextual information and hence improve the final performance of the model. Our findings are in accordance with the MMTM [30] method which shows the advantages of intermediate fusion over early and late fusion.

D. COMPARISON BETWEEN MULTI-LEVEL FUSION SCHEMES

In this section, we conduct experiments on the different fusion schemes and attempt to gain insights into the information flow between the two streams. The setup for these experiments follows three different architectural designs shown in Figure 3. For all these schemes, we perform fusion at four different levels of the two-stream I3D (see Figure S2 in the supplementary material). In the end, we feed the fused information to a fully connected layer to perform gesture classification. We evaluate and compare the performances of all three schemes: (a) shared pathway, (b) bidirectional pathway, and (c) central pathway. Table 2 presents the results of all these different fusion architectures. We observe that the shared pathway variant outperforms the other two. Also, it has fewer model parameters compared to the others. These results indicate that the shared pathway is the preferred choice among all three different architectures. We use this shared pathway scheme for the remaining set of our experiments.

E. RESULTS FOR DIFFERENT BENCHMARK DATASETS

1) IsoGD RESULTS

Table 3 presents the results of our fusion method on the IsoGD dataset and compares them with those of state-of-the-art (SoTA) methods. For this experiment, we use two-stream I3D with four fusion blocks (see Figure S2 in the supplementary material). The top section of the table shows the

TABLE 3. Results for IsoGD Dataset: Comparison with state-of-the-art methods. R, D denote RGB, Depth. The results show the average validation and test accuracy in %.

Methods	Modality	Val Acc	Test Acc
I3D [4]	R	61.33	63.81
I3D [4]	D	56.68	63.56
baseline [8]	R+D	49.17	67.26
ASU [20]	R+D	64.40	67.71
SYSU-ISEE	R+D	59.70	67.02
Lostoy	R+D	62.02	65.97
AMRL [32]	R+D	60.81	65.59
XDETVP [35]	R+D	58.00	60.47
CAPF [37]	R+D	-	66.79
Ours	R+D	**65.75**	**69.02**

performance of the backbone I3D model [4] trained on RGB and depth modalities separately. We use their pre-trained weights while training the fusion network. The middle section presents the results of various SoTA methods, including the baseline [8], for RGB+depth fusion. Our results at the bottom show that the proposed fusion architecture outperforms previous multi-modal gesture recognition approaches. For example, our method surpasses the earlier SoTA method ASU [20] by a margin of 1.3%. To our knowledge, the proposed fusion network sets new benchmark results on the IsoGD dataset. Although the FOANet [22] achieves a test accuracy of 82.07%, it is neither directly comparable to our method nor to the methods listed in the table. It uses the hand detection FOA network prior to gesture recognition, computes RGB flow and depth flow, and performs the fusion of four modalities. We, therefore, decided to not include the FOANet results in Table 3. Unlike this approach, our fusion method is using raw gesture videos of both RGB and depth modalities only.

2) NVGesture RESULTS

In this section, we compare our fusion network results with SoTA methods on the NVGesture dataset [21] and also analyze the misclassifications. Table 4 reports the results on the NVgesture dataset. First, we train the individual I3D models [4] on RGB and depth videos, respectively, and their results are shown at the top section of the table. NVGesture is a small dataset and hence large models with more parameters might lead to memorization (overfitting). Therefore, we use only two fusion blocks CTFB3 and CFTB4 in two-stream I3D for this experiment (see Figure S2 in the supplementary material). This adds a very small number of extra parameters but captures the fusion information efficiently. We can see the results of different fusion methods in subsequent sections of the table. Our method outperforms existing SoTA and improves SoTA on RGB+depth fusion by approximately 1 to 1.5% compared to recent MMTM [30] and MTUT [1] methods. It also surpasses the results of RGB+depth+flow fusion of MMTM [30] and MTUT [1] by approximately 0.8%. The classification performance of our fusion network

TABLE 4. Results for NVGesture Dataset: Comparison with state-of-the-art methods. The results show the average test accuracy in %. Here, R-RGB, D-Depth, F-Optical flow.

Method	Modality	Test Acc	#Params
I3D [4]	R	78.42	-
I3D [4]	D	82.28	-
HOG+HOG2 [23]	R+D	36.9	-
I3D late [4]	R+D	84.43	24.6M
MFF [16]	R+F	84.7	-
MTUT [1]	R+D	86.10	-
MMTM [30]	R+D	86.31	31M
PointLSTM [2]	point clouds	87.90	
CAPF [37]	R+D	**91.70**	-
Ours	R+D	87.76	27.2M
Human [21]	R	88.4	

TABLE 5. Results for IPN Hand Dataset: Comparison with state-of-the-art methods. The results show the average test accuracy (Acc) in %. Parameters in millions (M). R-RGB, F-optical flows, S-segmentation masks. We finetune I3D on the IPN dataset (indicated by *).

Method	Modality	Acc	#Params
C3D [29]	R	77.75	50.75M
ResNeXt-101 [42]	R	83.59	47.51M
ResNet-50 [42]	R	73.1	46.25M
I3D* [4]	R	**90.1**	12.3M
I3D* [4]	F	89.7	12.3M
I3D* [4]	S	89.3	12.3M
ResNeXt-101 Fusion [3]	R+F	86.32	47.56M
ResNeXt-101 Fusion [3]	R+S	84.77	47.56M
ResNeXt-50 Fusion [3]	R+F	74.65	46.27M
ResNeXt-50 Fusion [3]	R+S	75.11	46.27M
Ours	R+F	**91.4**	27.2M
Ours	R+S	90.8	27.2M

(RGB+depth) is close to human level (color) recognition rate [21]. We also compare the number of model parameters that need to be updated. Table 4 shows that our fusion network has 4 million fewer parameters than MMTM fusion [30] and still achieves better results. PointLSTM [2] is not directly comparable to our method as it only uses point clouds as input and is not a true multi-modal model. CAPF [37] performs better than our method. The performance of our baseline I3D (RGB-78.42%, Depth-82.28%) is lower than the baselines for CAPF (RGB-89.58%, Depth-90.62%). This indicates the gain in performance of CAPF is largely due to the influence of baseline models rather than the fusion strategy itself. The same is reflected in IsoGD results (refer to Table 3) where it is shown that our method outperforms CAPF.

The confusion matrix summarizes the performance of our fusion architecture on the NVGesture dataset (see Figure S4 in the supplementary material). The model demonstrates a high level of accuracy in its predictions of 25 different gesture classes. However, infrequently, the model produces inaccurate predictions for gesture classes such as *move_hand_down* with *push_hand_down* and vice-versa. This is due to the appearance and motion of both of these gestures resembling each other. The gestures like *show_two_fingers*/*push_two_fingers_away*, *click_index_finger*/*show_index_finger*, etc. encounter a similar pattern. These similar gestures are occasionally misclassified with our model.

3) IPN HAND RESULTS

Table 5 presents the results of processing IPN hand data with baseline models and their comparison with our fusion network. Part 1 of Table 5 contains the results of the IPN baseline models. IPN [3] uses models C3D [29], ResNet50, and ResNext-101 [42] as baselines that are pre-trained on the jesture dataset [19]. On the other hand, we use I3D [4] as a baseline model initialized with the publicly available pre-trained weights on ImageNet [36] + Kinetics [4]. We perform training in two stages. First, we finetune I3D on individual modalities using pre-trained weights of respective modalities. While finetuning the segmentation mask modality, we used pre-trained RGB weights. Part 2 of Table 5 shows the performance of I3D on individual modalities. For RGB modality, I3D [4] improves over the best IPN baseline model ResNeXt-101 by 6% with only $(1/4)^{th}$ the number of parameters (12.3M) compared to the number of ResNext-101 parameters (47.56M). Second, we train the fusion network for a different combinations of modalities using pre-trained I3D from first stage. While training our fusion network, we choose the fusion points CTFB3 and CTFB4 in the feature hierarchy (see Figure S2 in the supplementary material). Part 3 of Table 5 presents the performance of IPN fusion networks and the bottom part contains the results of our proposed fusion network. The performance of our fusion network surpasses the individual I3D trained on each modality (RGB, flow, depth) by 1 to 1.5%. Furthermore, our I3D fusion network, which employs CTFBs to merge RGB and optical flow data, surpasses the performance of the ResNeXt-101 fusion network [3] by 5%. To our knowledge, this is the new SoTA result on the IPN hand dataset for isolated gesture recognition.

F. IMPACT OF BACKBONE NETWORKS

In this section, we conduct experiments to understand the effect of different backbone networks on the performance of fusion networks. Apart from the I3D-based fusion model, we analyze the variations in the performance on different backbone networks such as 3D ResNet50 [42], 3D MobileNetV2 [41], 3D SqueezeNetV2 [40], and 3D ShuffleNet [39]. Each of these networks was initialized with pre-trained weights on Kinetics [4]. In Table 6, we report the results of experiments conducted using the above-mentioned networks on NVGesture dataset [21]. It reveals several interesting observations. The I3D-based fusion network outperforms the other backbone-based fusion architectures. The performance of the 3D ResNet50 alone achieves competitive accuracy compared to I3D on an individual modality, but the 3D ResNet50-based fusion architecture performs

TABLE 6. Comparison of the performance of fusion networks using different backbones. NVgesture dataset [21] is used for this experiment. Acc_R, Acc_D, and Acc_F are average test accuracies of RGB, Depth, and fusion network respectively. The number of model parameters is in millions (M).

Network	Acc_R	Acc_D	Acc_F	$\#Params_F$
I3D [4]	**78.42**	82.28	**87.76**	27.2M
3D ResNet50 [42]	77.38	**82.92**	84.44	105.2M
3D MobileNetV2 [41]	69.92	80.12	81.45	8.3M
3D SqueezeNetV2 [40]	65.43	73.65	80.08	4.74M
3D ShuffleNet [39]	69.05	79.87	80.49	3.02M

poorly compared to I3D-based fusion. This might be due to memorization caused by the huge amount of model parameters of 3D ResNet50. The fusion architectures based on resource-efficient models like 3D MobileNetV2, 3D SqueezeNetV2, and 3D ShuffleNet show low test accuracy compared to I3D and 3D ResNet50-based fusion networks. However, these models are extremely lightweight (given the order of magnitude, and lower number of parameters to train) and provide a good trade-off between accuracy and resource efficiency.

V. CONCLUSION AND FUTURE WORK

In this work, we explore video-based multi-modal gesture recognition. First, we introduce the Convolutional Transformer Fusion Blocks (CTFBs) to encode a discriminative multi-modal representation. A CTFB consists of an Efficient Convolutional Self-Attention (ECSA) mechanism, a fusion operation, and an MLP module. We designed ECSAs using 3D depthwise separable convolution layers to capture local key spatio-temporal features from each modality. We then perform elementwise addition operation to fuse output feature maps from two ECSA modules each for one modality. An MLP encodes the global features from the combined representation of two different modalities. The extensive evaluation shows that our proposed CTFB not only achieves competitive or SoTA performance on benchmark datasets but is also resource efficient. We believe that the gain in performance is due to couple of factors: (1) The CTFB generates salient local features that capture the fine-grained details and global representations that capture the overall context, and (2) The CTFBs in two-stream networks (baseline) placed at different abstraction levels are able to exploit the complementary information from two modalities at various degrees of granularity (from fine to coarse-grained).

Next, we also proposed three different multi-level fusion schemes: (a) Shared Pathway, (b) Bidirectional Pathway, and (c) Central Pathway to study the information flow between two-stream CNNs. Our experiments show that the shared pathway variant performs better than the other two schemes. It is also noteworthy that multi-level fusion outperforms early, mid, and late fusion strategies. Nevertheless, our model performs poorly on gestures of different categories having similar appearance and motion patterns.

There are opportunities for further improvements in several directions. First, in this work, we only used two-stream networks with bimodal fusion. Therefore, we would like to investigate if the incorporation of additional modalities leads to more discriminative representations that can be used to effectively classify resembling gestures. Second, we only focused on different modalities of videos that are having same dimensions. In the future, we intend to examine how well CTFBs perform when handling modalities of differing dimensions, such as skeletons, text, and audio, across a range of tasks.

ACKNOWLEDGMENT

The authors would like to thank Jochen Lang and Daniel Schmid for constructive feedback and fruitful discussions.

REFERENCES

[1] M. Abavisani, H. R. V. Joze, and V. M. Patel, "Improving the performance of unimodal dynamic hand-gesture recognition with multimodal training," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1165–1174.
[2] Y. Min, Y. Zhang, X. Chai, and X. Chen, "An efficient PointLSTM for point clouds based gesture recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 5760–5769.
[3] G. Benitez-Garcia, J. Olivares-Mercado, G. Sanchez-Perez, and K. Yanai, "IPN hand: A video dataset and benchmark for real-time continuous hand gesture recognition," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 4340–4347.
[4] 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.
[5] J. Chen and C. M. Ho, "MM-ViT: Multi-modal video transformer for compressed video action recognition," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2022, pp. 786–797.
[6] L. Chi, G. Tian, Y. Mu, and Q. Tian, "Two-stream video classification with cross-modality attention," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 4511–4520.
[7] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2021, pp. 1–22.
[8] J. Duan, J. Wan, S. Zhou, X. Guo, and S. Z. Li, "A unified framework for multi-modal isolated gesture recognition," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 14, no. 1s, pp. 1–16, Mar. 2018.
[9] C. Feichtenhofer, A. Pinz, and R. Wildes, "Spatiotemporal residual networks for video action recognition," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2016, pp. 3476–3484.
[10] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1933–1941.
[11] 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)*, Oct. 2017, pp. 3154–3160.
[12] J. Hu, W. Zheng, J. Pan, J. Lai, and J. Zhang, "Deep bilinear learning for RGB-D action recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 346–362.
[13] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Mach. Learn.*, 2015, pp. 448–456.
[14] M. M. Islam and T. Iqbal, "HAMLET: A hierarchical multimodal attention-based human activity recognition algorithm," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 10285–10292.
[15] J. Kim, J. Koh, Y. Kim, J. Choi, Y. Hwang, and J. Choi, "Robust deep multimodal learning based on gated information fusion network," in *Proc. Asian Conf. Comput. Vis.*, 2018, pp. 90–106.
[16] O. Kopuklu, N. Kose, and G. Rigoll, "Motion fused frames: Data level fusion strategy for hand gesture recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 2184–21848.

[17] X. Li, Y. Hou, P. Wang, Z. Gao, M. Xu, and W. Li, "Trear: Transformer-based RGB-D egocentric action recognition," *IEEE Trans. Cognit. Develop. Syst.*, vol. 14, no. 1, pp. 246–252, Mar. 2022.
[18] Y. Li, K. Zhang, J. Cao, R. Timofte, and L. Van Gool, "LocalViT: Bringing locality to vision transformers," 2021, *arXiv:2104.05707*.
[19] J. Materzynska, G. Berger, I. Bax, and R. Memisevic, "The jester dataset: A large-scale video dataset of human gestures," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 2874–2882.
[20] Q. Miao, Y. Li, W. Ouyang, Z. Ma, X. Xu, W. Shi, and X. Cao, "Multi-modal gesture recognition based on the ResC3D network," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2017, pp. 3047–3055.
[21] P. Molchanov, X. Yang, S. Gupta, K. Kim, S. Tyree, and J. Kautz, "Online detection and classification of dynamic hand gestures with recurrent 3D convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4207–4215.
[22] P. Narayana, J. R. Beveridge, and B. A. Draper, "Gesture recognition: Focus on the hands," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5235–5244.
[23] E. Ohn-Bar and M. M. Trivedi, "Hand gesture recognition in real time for automotive interfaces: A multimodal vision-based approach and evaluations," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 6, pp. 2368–2377, Dec. 2014.
[24] A. Prakash, K. Chitta, and A. Geiger, "Multi-modal fusion transformer for end-to-end autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 7073–7083.
[25] A. Roitberg, T. Pollert, M. Haurilet, M. Martin, and R. Stiefelhagen, "Analysis of deep fusion strategies for multi-modal gesture recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 198–206.
[26] S. Zhuoran, Z. Mingyuan, Z. Haiyu, Y. Shuai, and L. Hongsheng, "Efficient attention: Attention with linear complexities," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2021, pp. 3530–3538.
[27] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. Conf. Neural Inf. Process. Syst.*, 2014, pp. 568–576.
[28] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.
[29] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4489–4497.
[30] H. R. Vaezi Joze, A. Shaban, M. L. Iuzzolino, and K. Koishida, "MMTM: Multimodal transfer module for CNN fusion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13286–13296.
[31] J. Wan, S. Z. Li, Y. Zhao, S. Zhou, I. Guyon, and S. Escalera, "ChaLearn looking at people RGB-D isolated and continuous datasets for gesture recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2016, pp. 761–769.
[32] H. Wang, P. Wang, Z. Song, and W. Li, "Large-scale multimodal gesture recognition using heterogeneous networks," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2017, pp. 3129–3137.
[33] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, and L. Zhang, "CvT: Introducing convolutions to vision transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 22–31.
[34] J. Yang, Z. Ren, C. Gan, H. Zhu, and D. Parikh, "Cross-channel communication networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 1297–1306.
[35] L. Zhang, G. Zhu, P. Shen, J. Song, S. A. Shah, and M. Bennamoun, "Learning spatiotemporal features using 3DCNN and convolutional LSTM for gesture recognition," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2017, pp. 3120–3128.
[36] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
[37] B. Zhou, P. Wang, J. Wan, Y. Liang, F. Wang, D. Zhang, Z. Lei, H. Li, and R. Jin, "Decoupling and recoupling spatiotemporal representation for RGB-D-based motion recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 20154–20163.
[38] V. Vielzeuf, A. Lechervy, S. Pateux, and F. Jurie, "Multilevel sensor fusion with deep learning," *IEEE Sensors Lett.*, vol. 3, no. 1, pp. 1–4, Jan. 2019, doi: 10.1109/LSENS.2018.2878908.
[39] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6848–6856.
[40] F. Iandola, S. Han, M. Moskewicz, W. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size," 2016, *arXiv:1602.07360*.
[41] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520, doi: 10.1109/CVPR.2018.00474.
[42] K. Hara, H. Kataoka, and Y. Satoh, "Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and ImageNet?" in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6546–6555.
[43] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.

BASAVARAJ HAMPIHOLI received the bachelor's degree in computer science from Visvesvaraya Technological University, India, and the master's degree in intelligent systems from Technical University Kaiserslautern, Germany. He is currently pursuing the Ph.D. degree with Ulm University, Germany.

CHRISTIAN JARVERS received the bachelor's degree in cognitive science from Osnabr University, Germany, and the master's degree in experimental and clinical neuroscience from Regensburg University, Germany. He is currently pursuing the Ph.D. degree with Ulm University, Germany.

WOLFGANG MADER received the Ph.D. degree in physics from the University of Freiburg. Currently, he is with BMW Car ITGmbH, where he works in the field of sensor fusion enabling next-generation autonomous driving functions.

HEIKO NEUMANN received the Ph.D. and Habilitation degrees from the University of Hamburg, Germany. He leads the Vision and Perception Science Laboratory, Institute of Neural Information Processing. He is currently a Professor with the Faculty of Engineering, Computer Science and Psychology, Ulm University, Germany.