Taylor & Francis
Taylor & Francis Group

Applied Artificial Intelligence
An International Journal

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/uaai20

Multi-Cue Gate-Shift Networks for Mouse Behavior Recognition

Longfeng Shen, Yulei Jian, Debao Chen, Fangzheng Ge, Xiangjun Gao, Huaiyu Liu, Qianqian Meng, Yingjie Zhang & Chengzhen Xu

To cite this article: Longfeng Shen, Yulei Jian, Debao Chen, Fangzheng Ge, Xiangjun Gao, Huaiyu Liu, Qianqian Meng, Yingjie Zhang & Chengzhen Xu (2022) Multi-Cue Gate-Shift Networks for Mouse Behavior Recognition, Applied Artificial Intelligence, 36:1, 2151680, DOI: 10.1080/08839514.2022.2151680

To link to this article: https://doi.org/10.1080/08839514.2022.2151680

Published online: 02 Dec 2022.

APPLIED ARTIFICIAL INTELLIGENCE
2022, VOL. 36, NO. 01, e2151680 (3981 pages)
https://doi.org/10.1080/08839514.2022.2151680

Taylor & Francis
Taylor & Francis Group

OPEN ACCESS Check for updates

Multi-Cue Gate-Shift Networks for Mouse Behavior Recognition

Longfeng Shen[a,b,c], Yulei Jian[a,b,c], Debao Chen[a,b,c], Fangzheng Ge[a,c], Xiangjun Gao[a,c], Huaiyu Liu[a,c], Qianqian Meng[a,c], Yingjie Zhang[a,c], and Chengzhen Xu[a,b,c]

[a]School of Computer Science and Technology, Huaibei Normal Universit', Huaibei, China; [b]Anhui Big-Data Research Center on University Management, Huaibei, China; [c]Anhui Engineering Research Center for Intelligent Computing and Application on Cognitive Behavior (ICACB), Huaibei, China

ABSTRACT

Automatic identification of mouse behavior plays an important role in the study of disease or treatment, especially regarding the short-term action of mice. Existing three-dimensional (3D) convolutional neural networks (CNNs) and two-dimensional (2D) CNNs have different limitations when addressing the task of mouse behavior recognition. For instance, 3D CNNs require a large calculation cost, while 2D CNNs cannot capture motion information. To solve these problems, a low-computational and efficient multi-cue gate-shift network (MGSN) was developed. First, to capture motion information, a multi-cue feature switching module (MFSM) was designed to utilize RGB and motion information. Second, an adaptive feature fusion module (AFFM) was designed to adaptively fuse the features. Third, we used a 2D network to reduce the amount of computation. Finally, we performed an extensive evaluation of the proposed module to study its effectiveness in mouse behavior recognition, achieving state-of-the-art accuracy results using the Jiang database, and comparable results using the Jhuang database. An absolute improvement of +5.41% over the benchmark gate-shift module was achieved using the Jiang database.

ARTICLE HISTORY
Received 23 June 2022
Revised 15 November 2022
Accepted 18 November 2022

Introduction

Mice are widely used in biomedical science research and their responses to disease or treatment are often measured by recording their behavior patterns. In most cases, the recordings are manually tagged. Annotating mouse recordings manually can be challenging, so having a reliable and automated behavior recognition system to complete the task using computers would be beneficial. With a high-performance system, we can solve the problem of manual annotation and improve efficiency. Several animal motion recognition systems have been proposed as a result of the existing research. These systems are mainly

CONTACT Longfeng Shen, longfengshen521@126.com School of Computer Science and Technology, Huaibei Normal University, No.100, Dongshan Road, Huaibei, Anhui 235000, China

divided into two types: traditional methods based on manual feature extraction and deep learning methods using neural networks.

Studies on the mouse behavior recognition system using the traditional method of manual feature extraction have investigated the following. In 2005, Dollár et al. (2005) used a classification of sparse spatio-temporal features to identify mouse behavior. In 2010, Jhuang et al. (2010) proposed a system for automatically analyzing the behaviors of caged mice. This system combined motion information between adjacent frames with mouse speed and position information, then used this input to support vector machine hidden Markov models (SVMHMM) to obtain the classification results. In 2012, another study created an application of AdaBoost with spatio-temporal and trajectory features to classify mouse behavior (Burgos-Artizzu et al. 2012).

The method of constructing a complex model based on manual feature expression can no longer meet the requirements of high precision and speed, but the introduction of deep learning brings a new development direction for animal behavior recognition. For example, in 2016, Kramida et al. (2016) proposed the use of VGG features and LSTM networks to identify mouse movement. In 2019, embedded networks were used to extract features for rats and scene contexts participating in social behavior events. These LSTM networks were then used for behavior recognition (Zhang, Yang, and Wu 2019). In 2019, Nguyen et al. (2019) proposed using I3D and R(2 + 1D) models to address challenges with mouse behavior recognition. This produced one of the most advanced deep learning models for human action recognition at that time, which played a significant role in mouse behavior recognition.

Deep neural networks have made significant progress in human action recognition (Feichtenhofer 2020; Feichtenhofer et al. 2019; Tran et al. 2015a; Wang et al. 2015, 2016a; Zhu et al. 2017). Time modeling is also important for capturing motion information in video for action recognition. Currently, mainstream action recognition methods are realized through two mechanisms. The common method learns motion features from RGB frames using either 3D-CNN (Hara, Kataoka, and Satoh 2018; Karpathy et al. 2014; Stroud et al. 2020; Tran et al. 2015a, 2015b) or time convolution implicitly (Li et al. 2021; Qiu, Yao, and Mei 2017; Tran et al. 2018; Wu et al. 2020; Xie et al. 2018). However, 3D-CNN often has a large amount of computation and poor performance because of the lack of sufficiently large datasets. The other method uses a two-stream convolution network (Carreira and Zisserman 2017; Feichtenhofer, Pinz, and Zisserman 2016; Shi et al. 2019; Simonyan and Zisserman 2014), in which one stream extracts spatial information from RGB frames, while the other stream extracts motion information from optical flow. This method can effectively improve the accuracy of action recognition and performs well on small datasets.

Inspired by the human action recognition method, we applied a human action recognition deep learning model to mouse behavior recognition. This study uses a human action recognition model with the Gate-Shift Module (GSM) (Sudhakaran, Escalera, and Lanz 2020) as the baseline model. The GSM is a lightweight module that can transform a 2D-CNN into an efficient extractor of spatiotemporal features. Our network is two-stream, consisting of two modules: multi-cue feature switching module (MFSM) and adaptive feature fusion module (AFFM). MFSM is a feature-switching module for RGB and optical flow, its purpose is to replace useless features with features of other cues. AFFM is capable of adaptive fusion of features after feature switching. After fusion, the features change from two-stream to single-stream; thus, the two-stream convolution network changes back to a single-stream convolution network. Therefore, the accuracy of behavior recognition can be effectively improved with only a small increase in the calculation.

The contributions of the proposed method are summarized as follows:

(1) We propose a new MFSM that can replace feature maps with other cues that have a better effect on the final result for mouse behavior recognition;
(2) We propose an AFFM that can make the features perform adaptive fusion after feature switching;
(3) We perform extensive ablation experiments on the proposed module to study its effectiveness in mouse motion recognition;
(4) We achieve improved results using the Jiang database and competitive results using the Jhuang Database, but only show a small increase in parameters and floating-point operations per second (FLOPs).

Related Work

Two-Stream Networks

The basic principle of the two-stream model structure is to first calculate the dense optical flow every two frames in the video sequence to obtain temporal information. Then the convolutional neural networks (CNN) model is trained based on video image, spatial, and temporal, and the two branches of the network are used to judge each of the action categories. Finally, the training results from the two networks were directly fused to obtain the final classification results. The advantage of a two-stream convolution network architecture is its high precision, but slow speed.

Feichtenhofer, Pinz, and Zisserman (2016) followed the architecture of two-stream convolution network fusion for video action recognition. To make better use of the spatiotemporal information from the two-stream model, the author improved the fusion strategy of spatiotemporal networks. They proposed five different fusion schemes for the fusion of spatial and temporal networks and three methods for the fusion of temporal networks. Wang, Qiao, and Tang (2015) listed the accuracy of a two-stream network using several of the latest CNN network architectures. Wang et al. (2016b) found that previous research results only accounted for short-term actions with an insufficient understanding of the time structure for long-term actions and small training samples. Therefore, a sparse time-sampling strategy and a video supervision strategy were used. The video was segmented by time domain and randomly selected segments were used to compensate for the first deficiency, while cross training, regularization, and data expansion were used to compensate for the second deficiency. This network structure is called a Temporal Segment Network (TSN). Due to the recent successful application of residual networks (ResNet) (He et al. 2016) in deep learning, Feichtenhofer et al. proposed a novel spatio-temporal residual network model, which combines ResNet and a two-stream model (Christoph and Pinz 2017). The temporal and spatial characteristics of behavior are hierarchically learned through residual connections between spatial and temporal flows.

In the early stage of feature extraction, we use a two-stream network that combines an RGB image and an optical flow image. The purpose is to use the complementary advantages of the two cues to conduct mouse behavior recognition, which further improves the accuracy of prediction. After feature fusion, the two-stream network is transformed into a single-stream network, which can effectively control computing cost and the number of parameters, and improve recognition performance.

Feature Fusions

In many studies, the fusion of different modal features is an important method for improving accuracy. Combining the features of different modalities can achieve a better recognition effect by using their complementarity. Chaaraoui, Padilla-Lopez, and Florez-Revuelta (2013) proposed a method combining 2D shape human pose estimation with bone features. Integrating effective 2D contours and 3D bone features can yield visual features with high discrimination value, and the additional discrimination data provided by the contour can be utilized to improve the robustness of human action recognition errors. Sanchez-Riera et al. (2016) combined RGB features with depth features for gesture recognition and general object recognition, then evaluated the two schemes of early and late fusion. Li, Leung, and Shum (2016) proposed a multi-feature sparse fusion model that extracts multiple features of human body parts from skeleton and depth data. When using sparse regularization technology to automatically identify the feature structure of key parts, the learned weighted features are more discriminative for multi-task classification. Chen, Jafari, and Kehtarnavaz (2014) extracted depth image features and RGB video features of human actions using a depth camera and inertial body sensor to evaluate two recognition frameworks: feature-level and decision-level fusion.

In the current paper, we propose an effective AFFM that enables the network to directly learn how to filter the features of different modals to retain only useful information for combination. At each spatial location, the features of different modals are adaptively fused, and some features may be filtered out because they have conflicting information at that location, while others may dominate.

Methods

In this section, we present Multi-cue Gate-Shift Networks (MGSN) for mouse behavior recognition, which includes the two modules: MFSM and AFFM. We first introduce the two submodules and then outline how they are integrated into MGSNs.

Multi-Cue Feature Switching Module

The MFSM requires the use of a BN layer, we first introduce a batch normalization (BN) layer (Ioffe and Szegedy 2015). The function of the BN layer is to enhance generalization and speed up network training and convergence. We used $x_{m,c}$ to represent the c-th feature map in the m-th feature network. After normalization of the BN layer, $x_{m,c}$ performs an affine transformation,

$$x'_{m,c} = \gamma_{m,c} \frac{x_{m,c} - \mu_{m,c}}{\sqrt{\sigma^2_{m,c} + \varepsilon}} + \beta_{m,c} \tag{1}$$

where $\mu_{m,c}$ and $\sigma_{m,c}$ represent the mean and mean standard deviation of all pixel positions of the feature map c in the m-th feature network respectively. $\gamma_{m,c}$ and $\beta_{m,c}$ are trainable scaling factors and offsets, respectively; ε is a small constant that prevents division by zero. The function of factor $\gamma_{m,c}$ is to evaluate the correlation between $x'_{m,c}$ and $x_{m,c}$ in training. If $\gamma_{m,c} \rightarrow 0$, the loss gradient $x'_{m,c}$ will approach zero, which means that $x'_{m,c}$ will lose its influence on the result, and therefore $x'_{m,c}$ will become redundant.

We got inspired by Wang et al. (2020) to replace the feature maps with smaller $\gamma_{m,c}$ with those of other feature networks, because these feature maps would lose their influence on the result and become redundant feature maps. To solve this problem, we propose the following formula:

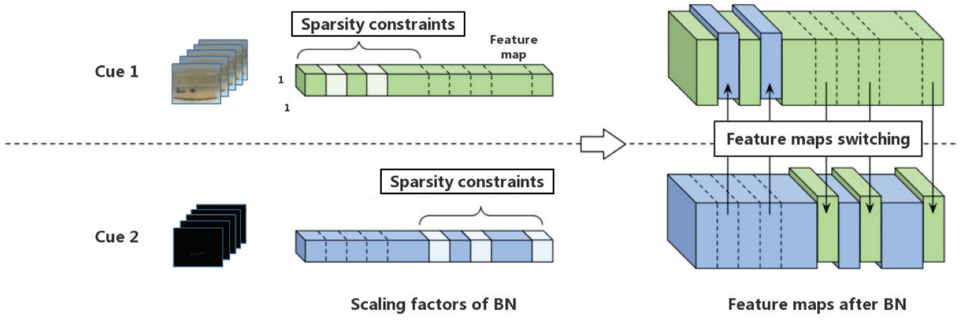


Figure 1. An illustration of our multi-cue fusion strategy. The sparsity constraints on scaling factors are applied to disjoint regions of different cues. If a feature map's scaling factor is lower than the specified threshold, the feature map will be replaced by that of other cues at the same position.

$$x'_{m,c} = \begin{cases} \gamma_{m,c} \frac{x_{m,c}-\mu_{m,c}}{\sqrt{\sigma^2_{m,c}+\varepsilon}} + \beta_{m,c} & \text{if} \gamma_{m,c} > \theta; \\ \frac{1}{M-1} \sum_{m' \neq m}^{M} \gamma_{m',c} \frac{x_{m',c}-\mu_{m',c}}{\sqrt{\delta^2_{m',c}+\varepsilon}} + \beta_{m',c}, & \text{else}; \end{cases} \tag{2}$$

In Equation (2), if the scaling factor $\gamma_{m,c}$ of feature map c is less than the threshold θ (θ we set in the experiment is 1e-2), the current feature map c is replaced with the average of the feature map c of other feature networks. In other words, if one feature map of a cue loses its influence on the result, it is replaced by the average value of other characteristic network feature maps. In our implementation, we applied the above formula to the process of feature extraction and each cue switch feature maps after convolution and nonlinear activation. We represent the scaling factor that must be switched as $\gamma_{m',c}$, and apply a sparse constraint on $\gamma_{m',c}$ to avoid unnecessary switching. This not only enables the replacement of useless feature maps, but also avoids the occurrence of useless switching.

We divide the entire feature map into M equal sub-parts, and only perform the feature switching for different cues in each different sub-part. We denote the scaling factors that can be replaced by $\hat{\gamma}_m$. Contrary to $\hat{\gamma}_m$ the switching in Equation (2) is a directed process within only one sub-part of the feature maps, which ideally will not only retain cue-specific propagation in the other $M - 1$ sub-parts, but also avoid unavailing switching since $\hat{\gamma}_m$. Figure 1 illustrates the feature-switching process.

Adaptive Feature Fusion Module

We refer to different feature fusion methods and finally use the adaptive feature fusion method (Liu, Huang, and Wang 2019) to design our AFFM. In contrast to previous MFFMs based on element summation or splicing,

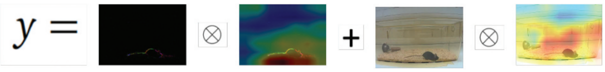


Figure 2. The feature maps of two cues are fused according to the learned weight map.

in our AFFM, for each pixel (i, j) on the fused feature map, the weights of the features from each clue at the pixel position are adaptively learned.

Let x_{ij}^{m} be the value of the feature vector at the pixel (i, j) position of the feature map of the m-th clue. The feature fusion method proposed in this study is as follows:

$$y_{ij} = \alpha_{ij} \cdot x_{ij}^{1} + \beta_{ij} \cdot x_{ij}^{2} \quad , \tag{3}$$

where α_{ij} and β_{ij} represent the normalized weight coefficients of the two different cues when the features are adaptively fused at the pixel (i, j) position of the fused feature map. The value y of the feature vector after feature fusion at the pixel (i, j) position can be calculated using Eq.(3).

In our method, a 1×1 convolution layer was added after the feature maps of the two cues, and two convolution maps were obtained. The values a and b at the pixel (i, j) positions in the two convolution maps were taken as the weight coefficients of the features of the two cues and then normalized by the SoftMax function. Inspired by Wang, Wang, and Lin (2019), we force the value of each weight coefficient to be normalized to the interval $[0, 1]$, and the sum of each weight coefficient to be normalized to one, that is, $\alpha_{ij} + \beta_{ij} = 1$ and $\alpha_{ij}, \beta_{ij} \in [0, 1]$ were defined. Finally, the normalized weight coefficients α_{ij} and β_{ij} of the feature fusion for the two cues were obtained. This was represented by the following formula:

$$\alpha_{ij} = \frac{e^{\lambda_{a_{ij}}}}{e^{\lambda_{a_{ij}}} + e^{\beta_{a_{ij}}}} \tag{4}$$

$$\beta_{ij} = \frac{e^{\beta_{a_{ij}}}}{e^{\lambda_{a_{ij}}} + e^{\beta_{a_{ij}}}} \tag{5}$$

With this method, α_{ij} and β_{ij} can thus be learned through standard back-propagation with the features are adaptively aggregated at each cue. AFFM is shown in the Figure 2.

Multi-Cue Gate-Shift Networks

Overview

MGSNs use MFSM and AFFM to feature map switching and adaptive fusion of the features for the two cues, so that the complementary advantages of the

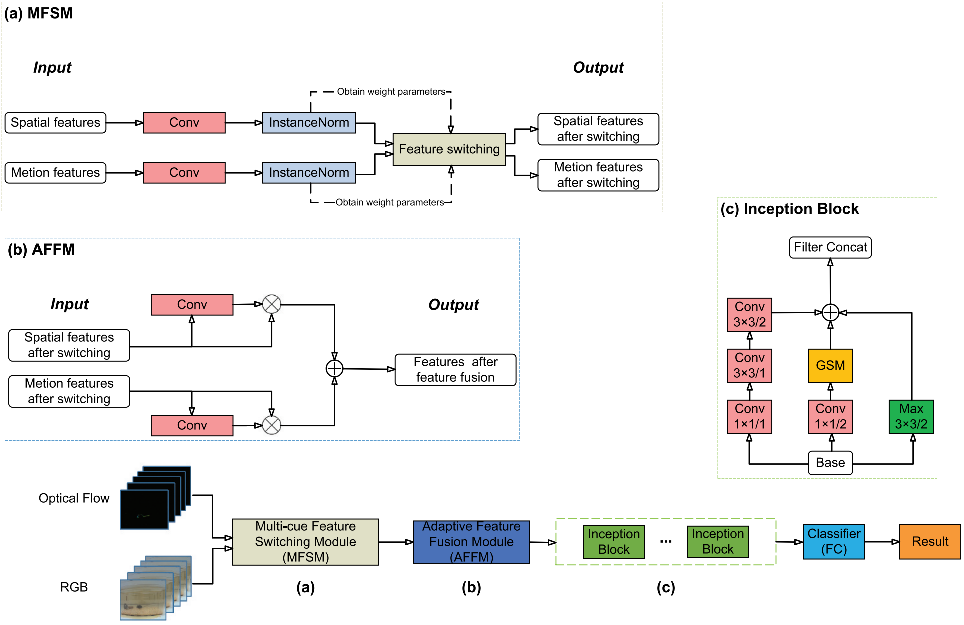


Figure 3. The overall architecture for InceptionV3. An illustration of MFSM, AFFM and inception block.

two cues can be fully utilized. The output features are identified using Gate-Shift Networks. Thus, the output of the GSM can be viewed from a spatiotemporal perspective, channel interdependence and motion-sensing information. Figure 3 shows the MGSN architecture for InceptionV3.

The Architecture of Multi-Cue Gate-Shift Networks

We used TSN as the reference architecture for behavior recognition, which uses the C2D backbone to perform the time pool of frame-level features. We choose to use BN-Inception and InceptionV3 as the backbone options for TSN, but we made a few modifications to the feature extraction part in the front of the backbone. We changed the input to RGB and optical flow two-stream input and insert for MFSM and AFFM. Subsequently, we inserted GSM into the backbone.

Algorithm Pseudocode

Algorithm 1 Multi-cue Feature Switching

```
Require: The whole feature maps of a cue $F$, the weight parameter $\gamma$ of the BN layer and the threshold $\theta$ of
    whether to switch or not
while Traverse $F$ do
```

(*Continued*)

Algorithm 1 Multi-cue Feature Switching

```
    if $\gamma > \theta$ then
        Perform affine transformation for the part of $F$ that conforms to $\gamma > \theta$ based on Eq.2
    else
        Perform multi-cue feature switching for the part of $F$ that conforms to else based on Eq.2
    end if
end while
return $F$
```

Algorithm 2 Adaptive Feature Fusion

```
Require: The whole feature maps of two cues $F_1$, $F_2$
After the two cues pass through convolution layer, batch normalization layer and activation function, $weight_1$ and $weight_2$ are obtained.
Concatenate $weight_1$ and $weight_2$ on channel dimension to obtain $weight_v$
Reduce the number of channels $weight_v$ through $1 \times 1$ convolution to obtain $weight$
Apply softmax function on the second dimension based on Equation (4) and (5) to get $\alpha$ and $\beta$ in the Equation (3)
Perform adaptive feature fusion to obtain $F'$ based on Equation (3)
return $F'$
```

Experiments and Results

Datasets

Our paper uses two datasets, namely, the Jhuang et al. (2010) and Jiang et al. (2018) datasets. The Jhuang dataset includes eight behavioral categories: drink (drink from the water supply), eat (take food from the feeding door), groom (the mouse combs its fur), hang (the mouse hangs on the top of the cage), head (slight movement of the limbs or head), rear (standing position, forelimb off the ground), rest (the mouse stays stable or sleeps), and walk (the mouse walks or runs in the cage). In addition to the above public data set, the Jiang dataset was also used, which included six behavior categories: dig (lift wood chips with forelimbs or head), eat (the rat gets food from the food box), groom (forelimbs sweep across the face or torso), rear (standing position, forelimbs off the ground), head (slight movement of limbs or head), and walk (movement). Sample video frames from the Jiang and Jhuang databases are shown in Figure 4. The number of frames in the two datasets is shown in Figure 5.

Implementation Details

In our experiments, BN-Inception and InceptionV3 were chose as the CNN backbones. MFSM and AFFM were added to BN-Inception and InceptionV3. All models used for the comparisons were initialized using ImageNet pretrained weights. We trained the entire network end-to-end using Stochastic Gradient Descent (SGD) with an initial learning rate of 0.01 and momentum of 0.9. A cosine learning rate schedule was used to

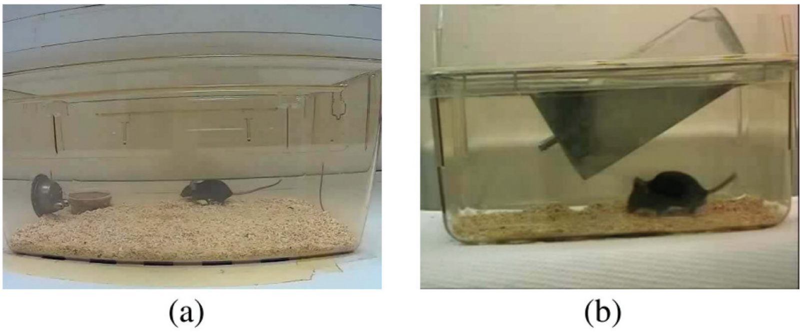
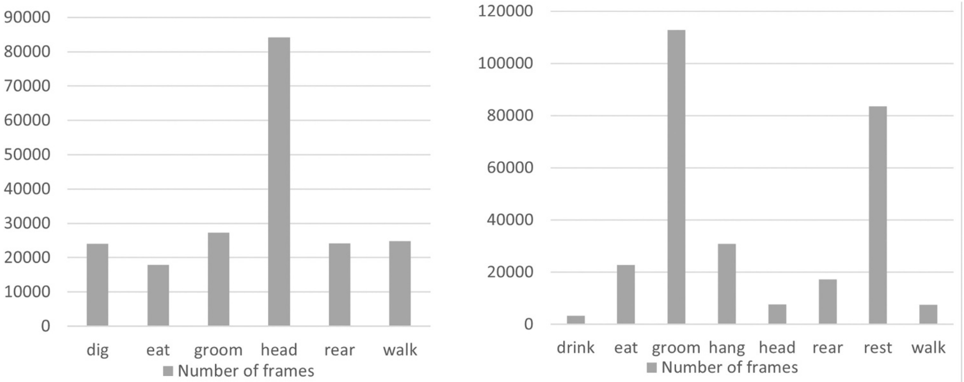


Figure 4. Dataset used in our experiment. (a) The Jiang dataset. (b) The Jhuang dataset.



(a) The Jiang dataset

(b) The Jhuang dataset

Figure 5. Distribution of number of frames for each behavior in the Jiang and the Jhuang dataset.

adjust the learning rate. The network was trained for 100 epochs using the Jiang and Jhuang databases. The first 10 epochs were used for gradual warm-up. The batch size was 16 for these databases. The classification layer for both databases apply dropout at a rate of 0.5. We applied random scaling, cropping, and flipping to augment data during training. The dimensions of the input were 224×224 for BNInception and InceptionV3. We used the center crop during inference.

Comparison with the State of the Art

Descriptions of Existing Methods Used for Comparison

We compare our proposed MGSN with other methods using motion information or temporal modeling methods. Results are shown in Tables 1 and 2. These methods all use ResNet as the backbone and 16 frames as input. A TDN (Wang et al. 2021) was proposed to extract multi-scale temporal information. ACTION-Net (Wang et al. 2021) proposed a plug-and-play ACTION module

Table 1. Comparison to state-of-the-art accuracy in the Jiang database (Red test denotes the best accuracy, blue is the second best accuracy, green is the third best accuracy).

Method	Backbone	FLOPs(G)	Accuracy
TDN (CVPR2021) (Wang et al. 2021)	ResNet-50	36.00	80.41%
PAN (TIP2020) (Zhang et al. 2020)	ResNet-50	35.70	77.51%
ACTION-Net (CVPR2021) (Wang et al. 2021)	ResNet-50	34.75	78.38%
TAM (ICCV2021) (Liu et al. 2021)	ResNet-50	82.00	77.85%
TDN (CVPR2021) (Wang et al. 2021)	ResNet-101	66.00	81.31%
TAM (ICCV2021) (Liu et al. 2021)	ResNet-101	82.00	78.55%
TEA (CVPR2020) (Li et al. 2020)	Res2Net-50	35.00	76.01%
GSM (CVPR2020) (Sudhakaran, Escalera, and Lanz 2020)	InceptionV3	26.82	77.70%
GSM (CVPR2020) (Sudhakaran, Escalera, and Lanz 2020)	BN-Inception	16.56	80.79%
SFV-SAN pipeline (Jiang et al. 2018)	N/A	N/A	72.40%
SFV-SAN+HMM pipeline (Jiang et al. 2018)	N/A	N/A	74.70%
MGSN (Ours)	InceptionV3	28.21	83.11%
MGSN (Ours)	BN-Inception	17.72	82.70%

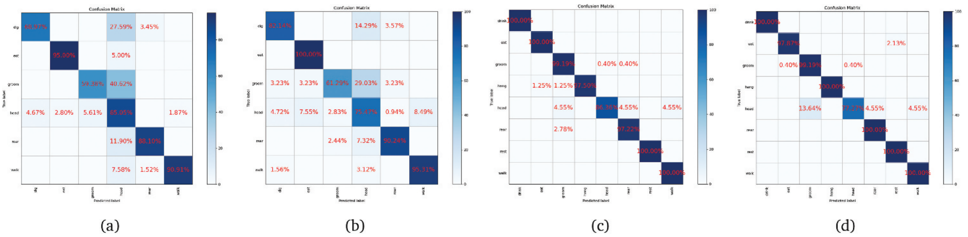
Table 2. Comparison to state-of-the-art accuracy in the Jhuang database (Red text denotes the best accuracy, blue is the second best accuracy, green is the third best accuracy).

Method	Backbone	FLOPs(G)	Accuracy
TDN (CVPR2021) (Wang et al. 2021)	ResNet-50	36.00	98.90%
PAN (TIP2020) (Zhang et al. 2020)	ResNet-50	35.70	95.62%
ACTION-Net (CVPR2021) (Wang et al. 2021)	ResNet-50	34.75	98.91%
TAM (ICCV2021) (Liu et al. 2021)	ResNet-50	82.00	97.18%
TDN (CVPR2021) (Wang et al. 2021)	ResNet-101	66.00	98.03%
TAM (ICCV2021) (Liu et al. 2021)	ResNet-101	82.00	97.81%
TEA (CVPR2020) (Li et al. 2020)	Res2Net-50	35.00	98.12%
GSM (CVPR2020) (Sudhakaran, Escalera, and Lanz 2020)	InceptionV3	26.82	98.44%
GSM (CVPR2020) (Sudhakaran, Escalera, and Lanz 2020)	BN-Inception	16.56	98.28%
SFV-SAN pipeline (Jiang et al. 2018)	N/A	N/A	96.50%
JHuang (Jhuang et al. 2010)	N/A	N/A	93.00%
MGSN (Ours)	InceptionV3	28.21	98.75%
MGSN (Ours)	BN-Inception	17.72	98.75%

that can extract appropriate spatio-temporal patterns, channel-wise features, and motion information to recognize actions. A Temporal Adaptive Module (TAM) (Liu et al. 2021) proposes an adaptive temporal modeling method, while the Temporal Excitation and Aggregation (TEA) (Li et al. 2020) block proposes to use both short- and long-range information. The above methods use motion information and are the most advanced methods available, which are of great significance.

Comparison with the Jiang Database

Table 1 shows the performance comparison between MGSNs and the most advanced methods from the Jiang database. Eight frames were used as input in the experiment. We used the various behavior recognition methods shown in Table 1 to conduct the experiments on the Jiang database and compared them with the methods used in the current study. Table 1 lists the comparison between the most advanced methods and our methods and the accuracy of using different backbones. As can be seen in the confusion matrix in



(a) (b) (c) (d)

Figure 6. Confusion matrixes for our system using different backbones with the Jiang and the Jhuang database. The diagonal cells contain the number and percentage of correct classified behaviors. The non-diagonal cells show the number and percentage of incorrectly classifications. (a) The Jiang dataset (InceptionV3). (b) The Jiang dataset (BN-Inception). (c) The Jhuang dataset (InceptionV3). (d) The Jhuang dataset (BN-Inception).

Figure 6(b), BN-Inception is more accurate in all categories except for the main category.

As shown in Table 1, MGSNs have a maximum absolute gain of 5.41% (83.11% vs. 77.70%) on the baseline GSM. In the same case, the backbone networks have different degrees of gain. The top three recognition accuracies in the table are MGSNs that use different modules. In addition, a state-of-the-art recognition accuracy of 83.11% was achieved by using InceptionV3, which is larger than BN-Inception. The TDN, PAN, ACTION-Net, and TAM methods use resnet50 as the backbone. The TDN has achieved good recognition accuracy. It was used by ResNet-101 to reach the previous highest accuracy of 81.31%, which is higher than the accuracy of GSM based on BN-Inception (80.79%). However, the accuracy of our MGSN based on BN-Inception exceeded that of all previous methods, with an accuracy of 82.70%. The recognition accuracy of our MGSN based on InceptionV3 was further improved, exceeding that of all the current methods by 83.11%. Our method is much lower than the aforementioned methods in terms of the amount of calculation, and can also obtain good results in the case of ground calculation.

Comparison with the Jhuang Database

Table 2 shows the performance comparison between MGSNs and the most advanced methods used in the Jhuang database. We trained the network using eight frames and sampled two clips. We use the various behavior recognition methods shown in Table 2 to conduct the experiments on the Jhuang database and to compare them with our methods. Among the methods in Table 2, our method attained a high degree of accuracy (the one marked in green is the third accuracy). However, because all of the methods showed very high accuracy in the Jhuang database, our method has no obvious advantage, but it also exceeds many of the most advanced methods available. Our method is only lower than that of TDN and ACTIONNet. The reason our method is lower is that the FLOPs of our method are much lower than those of TDN and ACTION-Net. Our method has a significant advantage in terms of computation and can achieve good results with low computation. The confusion matrices for the two backbones are shown in Figure 6(c,d).

Ablation Studies

In this section, we summarize the ablation analysis performed on the Jiang database. Exploration studies were performed on the Jiang database to investigate whether MFSM and AFFM showed performance improvement from the baseline GSM. The specific implementation details are described in Section 5.2.

Study on the Impact of Different Backbones

Different backbones were used to explore its impact. For overall accuracy, InceptionV3 performed best when the two modules were inserted together, resulting in a 0.41% higher accuracy than BN-Inception and 5.41% absolute gain over the baseline GSM. In Table 4, the accuracy, parameters, and FLOPs of the two backbones are presented. Inceptionv3 has better accuracy and is accompanied by a larger model.

Exploring Whether to Insert MFSM and AFFM

We then compared performance improvement by inserting the MFSM into InceptionV3 and BN-Inception. Table 4 shows the ablation results. Baseline was the standard GSM architecture, with accuracies of 77.70% and 80.79%. We then inserted MFSM. This improved the recognition performance by 2.37% and 0.18% for InceptionV3 and BNInception, respectively. Inserting AFFM also improved the recognition performance by 2.03% and 0.97%, respectively. The final model, in which MFSM and AFFM are inserted into InceptionV3 and BN-Inception, resulted in recognition accuracies of 83.11% and 82.70%, that is, a + 5.41% and +1.91% absolute improvement over the GSM baseline. Only 0.04% and 5.2% overhead in the parameters and complexity of InceptionV3, respectively. Similar to InceptionV3, only 0.1% and 7.0% overhead in parameters and complexity on BN-Inception, respectively.

Comparison of AFFM with Other Fusion Methods

Table 5 reports the comparison of our AFFM with three methods using the same backbone: addition, concatenation, and self-attention. For a more fair comparison, all experiments were conducted under the same experimental conditions, and the three methods were compared at the same location. The accuracy of our method outperformed the other fusion methods. While self-attention attains the closest performance to our method (82.56% vs. 83.11%), our method has fewer fusion parameters and calculations. The above conclusions can be drawn from the results in Table 5.

Table 3. Comparison of each category in the Jiang database (Red text denotes the best accuracy for IncetionV3 or BN-inception).

Behavior	InceptionV3				BN-Inception			
	GSM	MFSM +GSM	AFFM +GSM	MFSM+AFFM +GSM	GSM	MFSM +GSM	AFFM +GSM	MFSM+AFFM +GSM
dig	55.17%	58.62%	75.00%	68.97%	62.07%	57.15%	62.07%	82.14%
eat	90.00%	95.00%	100.00%	95.00%	95.00%	100.00%	90.00%	100.00%
groom	71.88%	56.25%	51.61%	59.38%	71.88%	61.29%	65.66%	61.29%
head	69.16%	78.50%	77.36%	85.05%	77.57%	81.13%	83.18%	75.47%
rear	83.33%	88.10%	92.68%	88.10%	90.48%	85.37%	88.10%	90.24%
walk	96.97%	93.94%	90.61%	90.91%	93.94%	92.19%	89.39%	95.31%
all	77.70%	80.07%	80.97%	83.11%	80.79%	80.97%	81.76%	82.70%

Table 4. Recognition accuracy for inserting modules in the Jiang database (Red text denotes the best accuracy for InceptionV3 or BN-inception).

Method	Accuracy	Params.(M)	FLOPs(G)
GSM (InceptionV3)	77.70%	21.86	26.82
GSM+MFSM (InceptionV3)	80.07%	21.86	28.07
GSM+AFFM (InceptionV3)	79.73%	21.87	28.21
GSM+MFSM+AFFM (InceptionV3)	83.11%	21.87	28.21
GSM (BN-Inception)	80.79%	10.32	16.56
GSM+MFSM (BN-Inception3)	80.97%	10.32	17.56
GSM+AFFM (BN-Inception)	81.76%	10.33	17.72
GSM+MFSM+AFFM (BN-Inception)	82.70%	10.33	17.72

Table 5. Comparison of AFFM with other fusion methods on Jiang database (the one marked in red is the best accuracy for InceptionV3).

Method	Accuracy	Params.(M)	FLOPs(G)
baseline	77.70%	21.86	26.82
baseline + concat	80.13%	21.87	29.32
baseline + sum	80.20%	21.86	28.11
baseline + self-attetion	82.56%	21.96	30.35
baseline + AFFM	83.11%	21.87	28.21

Exploring Accuracy of Different Categories

Table 3 shows the various classes and overall recognition accuracy of the Jiang database. As can be seen in the BN-Inception column in Table 3, dig, eat, and walk have the best precision with the insertion of MFSM and AFFM simultaneously, while the head has the best precision when inserting only AFFM. This is because the MFSM is more sensitive to motion information, while the behavior of the head is relatively static, so AFFM alone is more accurate. As can be seen in the InceptionV3 column in Table 3, inserting our modules in addition to the groom and walk classes resulted in better precision. Our module improves the recognition of most categories of behavior. For overall accuracy, InceptionV3 performed best when the two modules were inserted together, with a 0.41% higher accuracy than BN-Inception and 5.41% absolute gain over the baseline GSM. The benchmark GSM was more accurate for the groom and walk classes, but with our module, it was less accurate for these two classes. We tallied the predictions and found that both were most likely to misidentify the head. By analyzing our modules and prediction results, we conclude that the motion information in the optical flow has a significant influence on the MFSM. Because the motion information of groom, walk, and head are very similar, the recognition of these two classes is not as good as that of the baseline GSM. Using the Jiang database, the recognition performance of most types was improved, and the recognition performance of joint addition was better than that of single-module addition, which again proves the inference of a synergistic effect.

Conclusion

In this study, we proposed a MGSN for mouse behavior recognition. The core contribution of the MGSN was to include MFSM and AFFM to make full use of the complementary advantages of the two cues with little overhead. We performed an extensive evaluation to study the MGSN's effectiveness in mouse behavior recognition, achieving state-of-the-art accuracy results using the Jiang database, and obtaining competitive results using the Jhuang database. When MFSM and AFFM were inserted into the GSM baseline for InceptionV3, an absolute gain of +5.4% in recognition accuracy was obtained using the Jiang database with only 0.1% and 7.0% overhead in parameters and FLOPs, respectively.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This work is in part supported by the National Natural Science Foundation of China (Grant No: 61976101, 71801108), Anhui Provincial universities outstanding young backbone talents domestic visiting study and Research project (Grant No. gxgnfx2019006), the University Synergy Innovation Program of Anhui Province, China (GXXT-2021-002, GXXT-2022-033), the projects of Natural Science Foundation of Anhui Provincial Department of Education (Grant No: KJ2019A0603, KJ2019A0606, KJ2020B13), Anhui Provincial Department of Science and Technology fundamental RESEARCH and development program (Grant No. 201904a05020072), the Natural Science Foundation of Anhui Province, China (Grant No. 1808085MF174, 1908085QF286), Innovation Fund for Postgraduates of Huaibei Normal University (cx2022042), Open Laboratory project of Huaibei Normal University (Grant No: 2021sykf026, 2021sykf027), and 2022 National Innovation and Entrepreneurship Training Program for College Students (202210373005).

ORCID

Longfeng Shen http://orcid.org/0000-0003-1184-8552

Author contributions

Longfeng Shen: Conceptualization, Methodology, Software, Writing – original draft, Writing – review - editing. Yulei Jian: Methodology, Software, Validation, Data curation, Supervision, Resources. Debao Chen: Data curation, Software. Fangzheng Ge: Data curation, Software. Xiangjun Gao: Data curation, Software. Qianqian Meng: Data collection. Yinjie Zhang: Writing-review-editing. Chengzhen Xu: Writing-review-editing.

References

Burgos-Artizzu, X. P., P. Dollár, D. Lin, D. J. Anderson, and P. Perona (2012). Social behavior recognition in continuous video. In *2012 IEEE conference on computer vision and pattern recognition*, 1322–29. Providence, Rhode Island: IEEE.

Carreira, J., and A. Zisserman (2017). Quo vadis, action recognition? a new model and the kinetics dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, Honolulu, USA, 6299–308.

Chaaraoui, A., J. Padilla-Lopez, and F. Florez-Revuelta (2013). Fusion of skeletal and silhouette-based features for human action recognition with rgb-d devices. In *Proceedings of the IEEE international conference on computer vision workshops*, Sydney, Australia, 91–97.

Chen, C., R. Jafari, and N. Kehtarnavaz. 2014. Improving human action recognition using fusion of depth camera and inertial sensors. *IEEE Transactions on Human-Machine Systems* 45 (1):51–61. doi:10.1109/THMS.2014.2362520.

Dollár, P., V. Rabaud, G. Cottrell, and S. Belongie (2005). Behavior recognition via sparse spatio-temporal features. In *2005 IEEE international workshop on visual surveillance and performance evaluation of tracking and surveillance*, 65–72. Beijing, China: IEEE.

Feichtenhofer, C. (2020). X3d: Expanding architectures for efficient video recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, Seattle, USA, 203–13.

Feichtenhofer, C., H. Fan, J. Malik, and K. He (2019). Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, Seoul, Korea, 6202–11.

Feichtenhofer, C., A. Pinz, and R. P. Wildes. 2017. Spatiotemporal residual networks for video action recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 7445–54. doi:10.1109/CVPR.2017.787.

Feichtenhofer, C., A. Pinz, and A. Zisserman (2016). Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, Las Vegas, USA, 1933–41.

Hara, K., H. Kataoka, and Y. Satoh (2018). Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *Proceedings of the IEEE conference on computer vision and pattern recognition*, Salt Lake City, USA, 6546–55.

He, K., X. Zhang, S. Ren, and J. Sun (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, Las Vegas, USA, 770–78.

Ioffe, S., and C. Szegedy (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, 448–56. Lille, France: PMLR.

Jhuang, H., E. Garrote, X. Yu, V. Khilnani, T. Poggio, A. D. Steele, and T. Serre. 2010. Automated home-cage behavioural phenotyping of mice. *Nature Communications* 1 (1):1–10. doi:10.1038/ncomms1064.

Jiang, Z., D. Crookes, B. D. Green, Y. Zhao, H. Ma, L. Li, S. Zhang, D. Tao, and H. Zhou. 2018. Context-aware mouse behavior recognition using hidden Markov models. *IEEE Transactions on Image Processing* 28 (3):1133–48. doi:10.1109/TIP.2018.2875335.

Karpathy, A., G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei (2014). Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, Columbus, USA, 1725–32.

Kramida, G., Y. Aloimonos, C. M. Parameshwara, C. Fermüller, N. A. Francis, and P. Kanold (2016). Automated mouse behavior recognition using vgg features and lstm networks. In *Proceedings of the Visual Observation and Analysis of Vertebrate and Insect Behavior Workshop (VAIB)*, Cancun, MEXICO, 1–3.

Li, Y., B. Ji, X. Shi, J. Zhang, B. Kang, and L. Wang (2020). Tea: Temporal excitation and aggregation for action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, Seattle, USA, 909–18.

Li, M., H. Leung, and H. P. Shum (2016). Human action recognition via skeletal and depth based feature fusion. In *Proceedings of the 9th international conference on motion in Games*, Burlingame, California, 123–32.

Liu, S., D. Huang, and Y. Wang. 2019. Learning spatial fusion for single-shot object detection. *arXiv e-prints* arXiv:1911.09516.

Liu, Z., L. Wang, W. Wu, C. Qian, and T. Lu (2021). Tam: Temporal adaptive module for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, Montreal, Canada, 13708–18.

Li, H., Z. Wu, A. Shrivastava, and L. S. Davis (2021). 2d or not 2d? adaptive 3d convolution selection for efficient video recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, Virtual, 6155–64.

Nguyen, N. G., D. Phan, F. R. Lumbanraja, M. R. Faisal, B. Abapihi, B. Purnama, M. K. Delimayanti, K. R. Mahmudah, M. Kubo, and K. Satou. 2019. Applying deep learning models to mouse behavior recognition. *Journal of Biomedical Science and Engineering* 12 (02):183–96. doi:10.4236/jbise.2019.122012.

Qiu, Z., T. Yao, and T. Mei (2017). Learning spatio-temporal representation with pseudo-3d residual networks. In *Proceedings of the IEEE international conference on computer vision*, Venice, Italy, 5533–41.

Sanchez-Riera, J., K. -L. Hua, Y. -S. Hsiao, T. Lim, S. C. Hidayati, and W. -H. Cheng. 2016. A comparative study of data fusion for rgb-d based visual recognition. *Pattern Recognition Letters* 73:1–6. doi:10.1016/j.patrec.2015.12.006.

Shi, L., Y. Zhang, J. Cheng, and H. Lu (2019). Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, Long Beach, USA, pp. 12026–35.

Simonyan, K., and A. Zisserman. 2014. Two-stream convolutional networks for action recognition in videos. *Advances in Neural Information Processing Systems* 27.

Stroud, J., D. Ross, C. Sun, J. Deng, and R. Sukthankar (2020). D3d: Distilled 3d networks for video action recognition. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, Snowmass Village, United States, 625–34.

Sudhakaran, S., S. Escalera, and O. Lanz (2020). Gate-shift networks for video action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, Seattle, USA, 1102–11.

Tran, D., L. Bourdev, R. Fergus, L. Torresani, and M. Paluri (2015a). Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, Santiago, Chile, 4489–97.

Tran, D., L. Bourdev, R. Fergus, L. Torresani, and M. Paluri (2015b). Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, Santiago, Chile, 4489–97.

Tran, D., H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri (2018). A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, Salt Lake City, USA, 6450–59.

Wang, Y., W. Huang, F. Sun, T. Xu, Y. Rong, and J. Huang. 2020. Deep multimodal fusion by channel exchanging. *Advances in Neural Information Processing Systems* 33:4835–45.

Wang, L., Y. Qiao, and X. Tang (2015). Action recognition with trajectory-pooled deep-convolutional descriptors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, Las Vegas, USA, 4305–14.

Wang, Z., Q. She, and A. Smolic (2021). Action-net: Multipath excitation for action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, Virtual, 13214–23.

Wang, L., Z. Tong, B. Ji, and G. Wu (2021). Tdn: Temporal difference networks for efficient action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, Virtual, 1895–904.

Wang, G., K. Wang, and L. Lin (2019). Adaptively connected neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, Long Beach, USA, 1781–90.

Wang, L., Y. Xiong, Z. Wang, and Y. Qiao. 2015. Towards good practices for very deep two-stream convnets. *arXiv e-prints* arXiv:1507.02159.

Wang, L., Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool (2016a). Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*, 20–36. Amsterdam, Netherlands: Springer.

Wang, L., Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool (2016b). Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*, 20–36. Amsterdam, Netherlands: Springer.

Wu, Z., H. Li, C. Xiong, Y. -G. Jiang, and L. S. Davis (2020). A dynamic frame selection framework for fast video recognition. *IEEE Transactions on Software Engineering* PP (99).

Xie, S., C. Sun, J. Huang, Z. Tu, and K. Murphy (2018). Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European conference on computer vision (ECCV)*, Munich, Germany, 305–21.

Zhang, Z., Y. Yang, and Z. Wu (2019). Social behavior recognition in mouse video using agent embedding and lstm modelling. In *Chinese conference on pattern recognition and computer vision (PRCV)*, 530–41. Xi'an, China: Springer.

Zhang, C., Y. Zou, G. Chen, and L. Gan. 2020. Pan: Towards fast action recognition via learning persistence of appearance. *arXiv e-prints* arXiv:2008.03462.

Zhu, X., Y. Xiong, J. Dai, L. Yuan, and Y. Wei (2017). Deep feature flow for video recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, Honolulu, USA, 2349–58.