BQN: Busy-Quiet Net Enabled by Motion Band-Pass Module for Action Recognition

Guoxi Huang and Adrian G. Bors, *Senior Member, IEEE*
Department of Computer Science, University of York, York YO10 5GH, UK

***Abstract*—A rich video data representation can be realized by means of spatio-temporal frequency analysis. In this research study we show that a video can be disentangled, following the learning of video characteristics according to their spatio-temporal properties, into two complementary information components, dubbed Busy and Quiet. The Busy information characterizes the boundaries of moving regions, moving objects, or regions of change in movement. Meanwhile, the Quiet information encodes global smooth spatio-temporal structures defined by substantial redundancy. We design a trainable Motion Band-Pass Module (MBPM) for separating Busy and Quiet-defined information, in raw video data. We model a Busy-Quiet Net (BQN) by embedding the MBPM into a two-pathway CNN architecture. The efficiency of BQN is determined by avoiding redundancy in the feature spaces defined by the two pathways. While one pathway processes the Busy features, the other processes Quiet features at lower spatio-temporal resolutions reducing both memory and computational costs. Through experiments we show that the proposed MBPM can be used as a plug-in module in various CNN backbone architectures, significantly boosting their performance. The proposed BQN is shown to outperform many recent video models on Something-Something V1, Kinetics400, UCF101 and HMDB51 datasets. The code for the implementation is available[1].**

***Index Terms*—Motion Band-Pass Module, Busy-Quiet Net, Spatio-temporal video processing, Action Recognition.**

I. Introduction

Video action recognition is a fundamental problem in video understanding, having many real-world applications, including autonomous driving technology, video surveillance, drone movement control, robotics, human-computer interaction, augmented reality and gaming, among others. Over the last decade, video action recognition has attracted significant research interest while its fast development was empowered by deep learning [1] and the availability of large-scale labeled video datasets [2], [3]. Downstream tasks such as video retrieval and robot control also benefit from advances in video action recognition.

The Convolution Neural Network (CNN) architectures [4], [5], [6], [7], provide excellent results in image classification, but cannot be directly applied for video processing and classification. A straight forward manner to use CNNs for processing videos was to expand the convolution kernels from 2D to 3D [8], [9], [10], [11]. Starting with the Inflated 3D ConvNet (I3D) [10], research efforts in video processing and classification have been directed towards designing new 3D architectures. However, 3D CNNs require significantly more computation resources than 2D CNNs. Some recent works [12], [13], [14], [15], [16], [17] would increase the efficiency of 3D CNNs by reducing the redundancy in the model parameters. Nevertheless, these works ignore the fact that videos contains substantial redundancy in the spatio-temporal space, which results in the inefficient processing by existing systems. Meanwhile, the 3D CNN performance [10] can be further improved by simply replacing raw time-series of RGB frame inputs with motion representations, such as for example the TV-L1 flow [18].

Video data processing can also benefit from being decomposed into different streams of information which would be allocated the appropriate computational resources for processing each data stream. Busy information describes fast-changing motion happening at the boundaries of moving regions. Such regions are crucial both for defining movement as well as for action recognition. Meanwhile, the Quiet information contains substantial redundancy, as for example when having continuous background textures. In order to efficiently process video data, we propose to disentangle a video stream, or just a clip, into Busy and Quiet components. Subsequently, we process efficiently the Busy and Quiet components separately, by allocating high-complexity processing for the Busy information while low-complexity processing would be used for the Quiet information.

This research study proposes a lightweight, end-to-end trainable motion feature extraction mechanism called the Motion Band-Pass Module (MBPM), which distills motion information conveyed within a specific spatio-temporal frequency bandwidth. The distillation is optimized with respect to relevant movements following the training with the videos from a given dataset. As illustrated in Figure 1, by applying MBPM to a video, selecting the representative frames whilst retaining and compressing the essential motion representation. Our experiments demonstrate that by simply replacing the RGB frame input with the motion representation extracted by MBPM, the performance of existing video models can be significantly boosted. Secondly, we design a two-pathway multi-scale architecture, called the Busy-Quiet Net (BQN), whose processing pipeline is shown in Figure 2. The Busy pathway is responsible for processing the information distilled by MBPM, representing fast changing spatio-temporal data. The other pathway, called Quiet, is devised for processing the information encoded by the global smoothing spatio-temporal network structures. In order to fuse the information from different pathways, we also employ Band-Pass Lateral Connection (BPLC) modules, facilitating the exchange of

[1] *https://github.com/guoxih/busy-quiet-net*

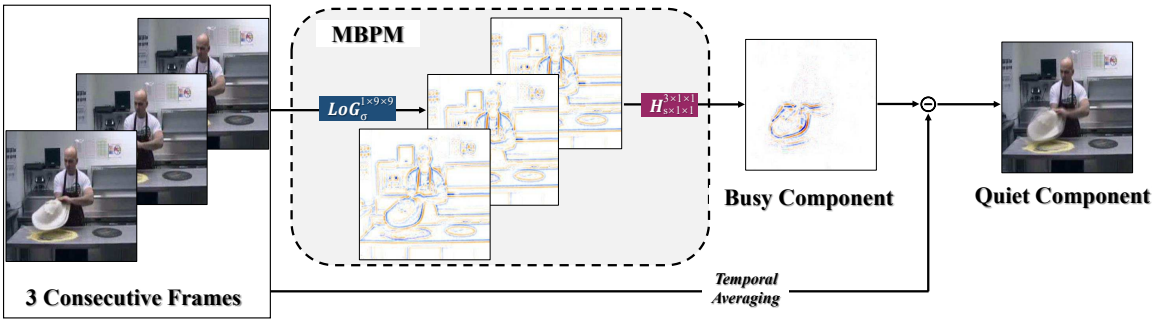


Fig. 1: The Motion Band-Pass Module (MBPM) disentangles a short frame sequence into Busy and Quiet components. For every three consecutive RGB frames, the MBPM generates a single-frame output, substantially reducing the redundancy.

information between the layers of Busy and Quiet pathways. Through experiments we demonstrate that BPLC modules represent the key factor to the overall model optimization success.

Compared with the frame summarization approaches [19], [20], MBPM retains the strict temporal order of the frame sequences, which is considered essential for long-term temporal relation modeling. Compared with optical flow-based motion representation methods [18], [21], [22], [23], [24], the motion representation captured by MBPM has a smaller temporal size and can be employed on the fly. Meanwhile, efficient video models such as Octave Convolution [25], bL-Net [26] and SlowFast networks [27] would only reduce the input redundancy along either the spatial or temporal dimensions. Instead, the proposed BQN reduces the redundancy in the joint spatio-temporal space.

The contributions of this paper can be summarized as follows:

- A novel Motion Band-Pass Module (MBPM) is proposed for Busy and Quiet motion information distillation. The motion cues extracted by the MBPM significantly reduces temporal redundancy.
- We design a two-pathway Busy-Quiet Net (BQN) that separately processes the Busy and Quiet information in videos. After separating the Busy information using MBPM, we further decrease redundancy by downsampling the Quiet information.
- Extensive experiments demonstrate the superiority of the proposed BQN over a wide range of models on four standard video benchmark datasets: Kinetics400 [10], Something-Something V1 [3], UCF101 [28] and HMDB51 [29].

The rest of the paper is organized as in the following. The related works are outlined in Section II. The proposed Motion Band-Pass Module (MBPM) and its training is described in Section III. The Busy-Quiet Net (BQN) is presented in Section IV. The experimental results are provided in Section V while the conclusions of this study are drawn in Section VI.

II. Related Work

A. Spatio-temporal Networks

Following the unprecedented breakthrough of 2D CNNs on image classification tasks [4], [5], [6], [30], [31], [32], early work [33] attempted to directly apply 2D CNNs to video action recognition tasks by simply fusing frame-wise prediction, while the temporal information of videos was not exploited. In order to enable 2D CNNs with spatio-temporal modeling abilities, Simonyan and Zisserman [34] proposed a representative two-stream architecture, in which the spatial stream processes raw RGB frames as input, while the temporal stream models motion-relevant features by taking the optical flow as input. Long Short-Term Memory (LSTM) [35] embedded with 2D CNNs was designed for learning temporal relations between frames in various studies [10], [36], [37], [38], [39], [40]. However, 2D CNN+LSTM [10] empirically shows lower performance than two-stream processing architectures. One possible explanation could be that the learning of temporal relationships by 2D CNN+LSTM [10] architecture is only being operated on the high-level 2D CNNs features, ignoring the importance of low-level temporal information. Given the progress in GPU performance, other methods [8], [9], [10], [11], tend to exploit the computationally intensive 3D convolution which allows simultaneous spatio-temporal data processing. Meanwhile, some studies focus on improving the efficiency of 3D CNN, such as the Pseudo-3D Residual Net (P3D) [14], R(2+1)D networks [16], Separable 3D (S3D) CNN [17], Temporal Shift Module (TSM) [13], Channel-Separated Convolutional Network (CSN) [15] and Expand 3D (X3D) [12]. Non-local Net [41] and its variants [42], [43] introduce self-attention mechanisms to CNNs in order to learn long-range spatio-temporal dependencies. This research study is complementary to these approaches and our Busy-Quiet Net (BQN) can benefit from the efficiency of these CNNs by simply adopting any of them as its backbone.

B. Motion Representation

Optical flow as a short-term motion representation has been widely used as an input modality in two-stream-based architectures [23], [34], [44] for boosting performance in action

recognition. Flow Net series [22], [45] and AutoFlow [46] improve the optical flow estimation by using deep learning. However, optical flow estimation is inefficient with respect to the memory storage and computation. To estimate the optical flow on the fly, ActionFlowNet [47] and Hidden Two-Stream Net [48] attempt to integrate optical flow estimation and action recognition into an end-to-end training framework. More recently, Optical Flow guided Feature (OFF) [49], TVNet [21], Flow-of-Flow [50], EMV [51], [52] and other methods employing fast motion feature learning have been proposed. Some motion representations such as Squeezed Image and Dynamic Image summarize both the static and dynamic visual information of videos by utilizing Temporal Squeeze Pooling [53] and Approximate Rank Pooling [19], respectively. These methods work well provided that no severe camera shaking or movement occurs. Otherwise, the resulted squeezed images are blurred, resulting in poor discrimination between the moving objects and background. Compared to these approaches, our MBPM produces higher accuracy while requiring less computation. Moreover, the proposed MBPM is like a basic component, which can be embedded into various video architectures.

C. Reducing Information Redundancy

Enabled by deep learning, Big-Little Net (bL-Net) [26] adopts a downsampling strategy operating at block level aiming to reduce the spatial redundancy of its feature maps. Then it uses two branches to separately process the feature maps with different resolutions. Chen *et al*. [25] replaced normal convolution operations with an Octave Convolution operation decomposing the video information into low-frequency and high-frequency components, while capturing additional global information. For action recognition, the Big-Little-Video-Net bLVNet [54] extends the idea of bL-Net [26] to the temporal dimension. SlowFast networks [27] introduce two pathways, and decompose the input into the Slow and Fast components along the temporal dimension for efficient temporal modeling. However, the generalization of SlowFast to existing CNN architectures is poor, as it requires specifically customized CNNs as backbones.

Different from existing methods, which only reduce feature redundancy either in spatial or in temporal dimension, the proposed BQN reduces the feature redundancy in the joint spatio-temporal space. We introduce a predefined trainable filter module, MBPM, to disentangle the video into Busy and Quiet components. In opposition to SlowFast, the BQN architecture provides excellent generalization when considering any CNN as its backbone.

III. Motion Band-Pass Module (MBPM)

Action recognition methods in videos aim to characterize the movement of persons and that of the moving regions that make up the scene as well as their interactions. Particularly, the regions from the boundaries of the moving regions defining both their shape and movement characteristics are essential for video representation. The movement of a moving object or region is defined by how their regions of transition or boundaries are changing over time. On the other hand, the interior of rigidly moving objects, as well as the background of their content, contains information which is rather constant in time, unless the recording camera is moving or there are changes in scene illumination. In this study we develop a mechanism for separating the information from regions critical to defining the movement in video, which are called Busy, from the regions which do not change much and are redundant in successions of frames, called Quiet. On one hand such a separation would enable a better characterization of the movement leading to better classification. On the other hand we aim to allocate appropriate computational resources for the efficient computation of the Busy and Quiet regions from videos.

In the following we introduce the Motion Band-Pass Module (MBPM), as a trainable 3D band-pass filter, which can distill the video information conveyed within a specific spatio-temporal frequency bandwidth, into Busy and Quiet information channels [55]. A video clip can be defined as a function of three arguments, $\boldsymbol{I}^{(t)}(x, y)$, where x, y indicate the spatial dimensions, while $t = 1, \ldots, T$ is the temporal dimension for a total of T frames. The value of $\boldsymbol{I}^{(t)}(x, y)$ corresponds to the RGB pixel at position (x, y) in the t-th video frame. When considering multi-channel video, we repeat the same procedure for each feature channel, which for the first processing layer corresponds to the color components. For the frequency band processing, we consider a filter based on the time differentials of the Laplacian of Gaussian (LoG), applied in the spatial frame data. The output $\boldsymbol{\Gamma}$ of the frequency band-selection filter is given by:

$$\begin{aligned} \boldsymbol{\Gamma}(x, y, t) &= \frac{\partial^2}{\partial t^2}\left[\boldsymbol{I}^{(t)}(x, y) * LoG_\sigma(x, y)\right], \\ &\approx \sum_{t-1 \le i \le t+1} h_{(i)} \cdot [\boldsymbol{I}^{(i)}(x, y) * LoG_\sigma(x, y)], \\ &\text{where } \; h_{(i)} = \begin{cases} \frac{2}{3} & \text{if } \; i = t, \\ -\frac{1}{3} & \text{otherwise}, \end{cases} \end{aligned} \tag{1}$$

for $t = 1, \ldots, T$ and '$*$' represents the convolution operation. Meanwhile, $LoG_\sigma(x, y)$ is a two-dimensional Laplacian of Gaussian with the scale parameter σ :

$$LoG_\sigma(x, y) = \nabla^2 G_\sigma(x, y) = -\frac{e^{-\frac{x^2+y^2}{2\sigma^2}}}{\pi\sigma^4}\left[1 - \frac{x^2 + y^2}{2\sigma^2}\right]. \tag{2}$$

In Eq. (1), the second derivative with respect to t is numerically approximated by finite differences, literally implemented by the function $h_{(i)}$. The scale parameter σ of $LoG_\sigma(x, y)$ determines what information would be disentangled and consequently plays a crucial role in defining the Busy information. $LoG_\sigma(x, y)$ with a larger σ captures smoother textures of videos, and is therefore more robust to noise. On the other hand, a smaller σ would reliably capture some high frequency information characterizing fast moving objects. In Section V-D we provide an ablation study for choosing σ.

From Eq. (1) and (2) we can observe that the 3D filtering function is fully-differentiable. In order to make the 3D band-pass filtering compatible with CNNs, we approximate it with

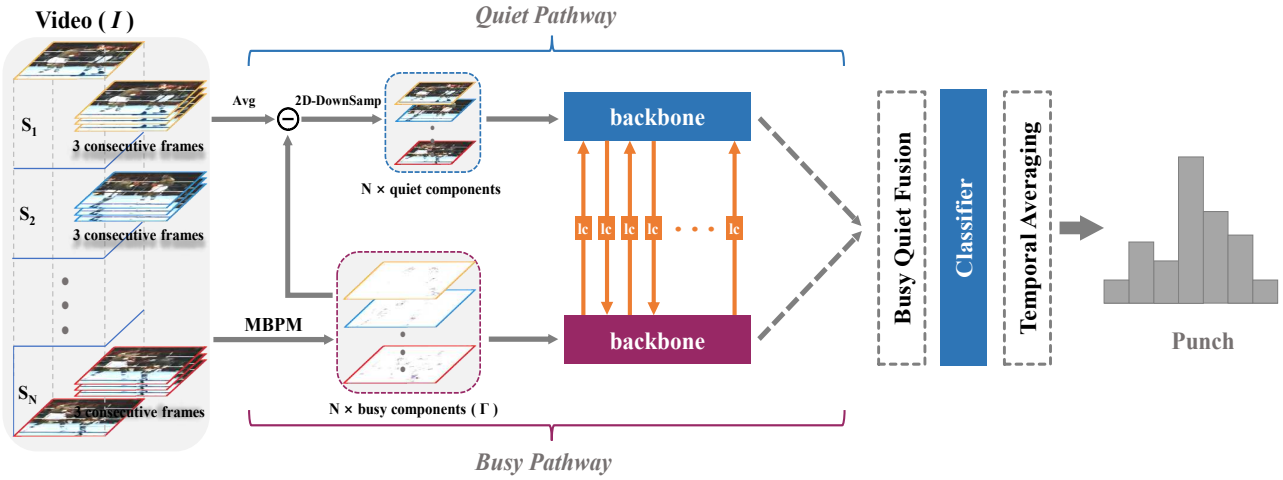


Fig. 2: The Busy-Quiet Net (BQN) is made up of two parallel pathways: Busy and Quiet. 'lc' indicates the Band-Pass Lateral Connection, which is bi-directional in this diagram. The backbone networks from the two pathways respectively take as inputs two complementary video data components, namely Busy and Quiet, disentangled by the MBPM. The outputs of the two pathways are eventually fused, and the final prediction is obtained by averaging the prediction scores across multiple segments.

two sequential channel-wise[2] convolutional layers [56], as shown in Figure 1. The Motion Band-Pass Module (MBPM) is a discrete approximation implementing $\mathbf{\Gamma}$ from Eq. (1) as :

$$\mathbf{\Gamma} \approx \text{MBPM}(\boldsymbol{I}) = H^{3\times1\times1}_{s\times1\times1}(LoG^{1\times k\times k}_{\sigma}(\boldsymbol{I})), \qquad (3)$$

where $LoG^{1\times k\times k}_{\sigma}$ is referred to as a spatial channel-wise convolutional layer [56], with a $k \times k$ kernel, each channel of which is initialized with a Laplacian of Gaussian distribution of scale σ. The sum of kernel values is normalized to 1. Meanwhile, $H^{3\times1\times1}_{s\times1\times1}$ is referred to as a temporal channel-wise convolutional layer with a temporal stride s. In each channel, the kernel $H^{3\times1\times1}_{s\times1\times1}$ is initialized with $[-\frac{1}{3}, \frac{2}{3}, -\frac{1}{3}]$, defining a high-pass filter. In order to adjust to the specifics of the motion characteristics from videos, the kernel parameters of MBPM are fine-tuned through training on a video dataset. We embed the MBPM in the CNN training process to form an end-to-end training pipeline, which is optimized with the classification loss. The training will result in an optimized MBPM, defined by the characteristics of real videos.

IV. Busy-Quiet Net (BQN)

The BQN architecture, illustrated in Figure 2, contains two different processing pathways: one for the Busy information and another for the Quiet information. Splitting the processing into two different processing pathways is justified by the fact that computational resources should be allocated differently, according to the characteristics of the information to be processed. The separation of the pathways is enabled by the MBPM, whose construction was explained in the previous section. Meanwhile, the Busy and Quiet pathways are bridged by multiple Band-Pass Lateral Connections ("lc" in Figure 2). These lateral connections enable information fusion between the two processing pathways at various processing stages.

[2]Also referred to as "depth-wise". We use the term "channel-wise" to avoid confusions with the network depth.

A. The Busy pathway

The Busy pathway is designed to learn essential fine-grained movement features, such as those characterizing the transitions of distinct regions of movement. It takes as input the information filtered by the MBPM, corresponding to a specific spatio-temporal frequency band, selected following the training of MBPM. The stride of $H^{3\times1\times1}_{s\times1\times1}$ from Eq. (3) is set in the experiments to $s = 3$, which means that for every three consecutive RGB frames, MBPM generates an one-frame output. The MBPM output preserves the temporal order within the video while significantly reducing the redundant temporal information. For extracting more distinct moving object textures or transitional movement variation patterns, we would consider larger input regions for the Busy pathway.

B. The Quiet pathway

The Quiet pathway focuses on processing Quiet information, representing the characteristics of large regions of movement, such as the movement happening in the plain-textured background regions or from the inner regions of large moving objects. Such information is usually repeating itself from frame to frame and contains a lot of redundant information. These regions would require reduced computational processing for video characterization, while they significance for the video classification should not be over-weighted. The input to the Quiet pathway is considered as the complementary to the MBPM output, given by:

$$\text{2D-DownSamp}\ (\text{Avg}^{3\times1\times1}_{3\times1\times1}(\boldsymbol{I}) - \mathbf{\Gamma}), \qquad (4)$$

where $\text{Avg}^{3\times1\times1}_{3\times1\times1}$ is the temporal average pooling, considering a stride of 3 in the experiments, and $\mathbf{\Gamma}$ defines the Busy information for $\boldsymbol{I}$, according to Eq. (3). We also perform bilinear downsampling in the spatial domain, along x and y coordinates (*i.e.* 2D-DownSamp), to reduce the redundant spatial information shared by neighboring locations in the

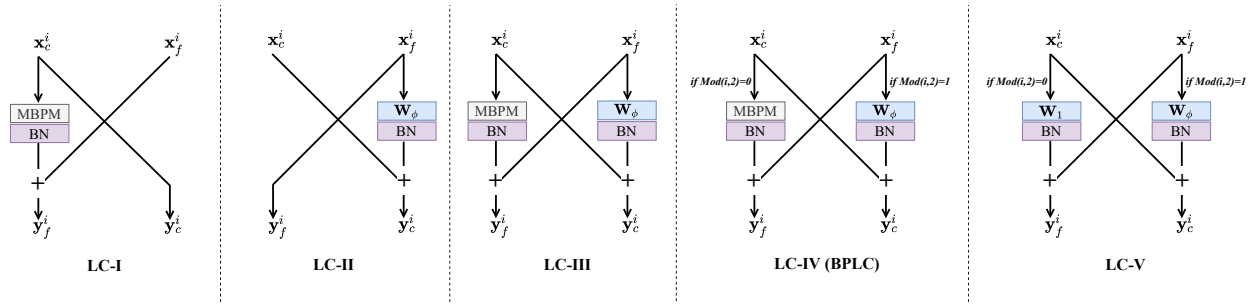


Fig. 3: Diagrams of various lateral connection (LC) designs. Bilinear interpolation is used for resizing the feature maps when $\mathbf{x}_c^i$ and $\mathbf{x}_f^i$ do not have the same spatial size. i refers to the index of the residual block. $\mathbf{W}_\phi$ and $\mathbf{W}_1$ denote the weights of the linear transformations.

Quiet information. In Section V-G1, we explore the Quiet information significance on the overall performance of BQN.

C. Band-Pass Lateral Connection (BPLC)

We also propose to include a novel Band-Pass Lateral Connection (BPLC) module, which has an MBPM embedded, in the BQN. The BPLCs, which are placed between the Busy and Quiet processing pathways, provide a mechanism for information exchange, enabling an optimal fusion of the two video information components Busy and Quiet corresponding to different frequency bands. Different from the lateral connections in other approaches [27], [44], [57], [58], the BPLC, enabled by MBPM, performs feature fusion and feature selection simultaneously, resulting in higher performance than other lateral connection designs, according to the experimental results. We denote the two inputs of BPLC from the i-th residual blocks in the Busy and Quiet pathways, as $\mathbf{x}_f^i$ and $\mathbf{x}_c^i$, respectively. For simplifying the notation, we assume that $\mathbf{x}_f^i$ and $\mathbf{x}_c^i$ are of the same size. When their sizes are different, we adopt bilinear interpolation to match them in size. The outputs $\mathbf{y}_f^i$ and $\mathbf{y}_c^i$ for the Busy and Quiet, respectively, are given by

$$\begin{aligned}
\mathbf{y}_f^i &= \begin{cases} \mathrm{BN}(\mathrm{MBPM}(\mathbf{x}_c^i)) + \mathbf{x}_f^i & \text{if} \quad \mathrm{mod}(i,2) = 0, \\ \mathbf{x}_f^i & \text{otherwise}, \end{cases} \\
\mathbf{y}_c^i &= \begin{cases} \mathbf{x}_c^i & \text{if} \quad \mathrm{mod}(i,2) = 0, \\ \mathrm{BN}(\phi(\mathbf{x}_f^i)) + \mathbf{x}_c^i & \text{otherwise}, \end{cases} \\
& \qquad i = 1, 2, \ldots, B
\end{aligned} \tag{5}$$

where BN indicates Batch Normalization [59], with the weights initialized to zero and B represents the number of residual blocks in the backbone network (considered as the network with residual block designs in experiments). $\phi(\cdot)$ is a linear transformation that can be implemented as a $1 \times 1 \times 1$ convolution, or alternatively, when the channel number is very large, as a bottleneck Multi-layer Perceptron (MLP) for reducing computation. 'mod' represents modulo and controls the feature fusion direction between the two pathways. When $\mathrm{mod}(i, 2) = 0$, the selected features from the Quiet pathway are fused to the Busy pathway. Otherwise, the features from the Busy pathway are fused to the Quiet pathway. For the MBPM in BPLC, the convolutional stride of $H_{s\times1\times1}^{3\times1\times1}$ from Eq. (3) is set to $s = 1$, maintaining the same temporal size.

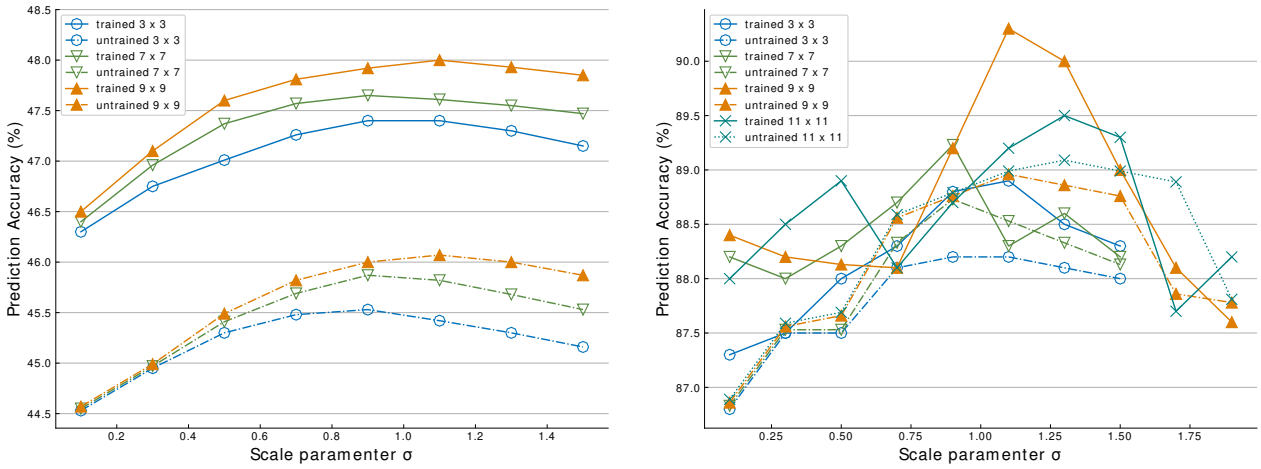
The fusion direction of BPLC reverses alternatively back and forth, as indicated in Figure 2, providing better communication for the two pathways than the unidirectional lateral connections in [27], [58] whose information fusion direction is fixed, always fusing the information from a certain pathway to the other. By default, we place a BPLC between the two pathways right after each pair of residual blocks. The MBPM embedded in BPLC acts as a soft feature selection gate, where only busy information from the Quiet pathway is allowed to flow to the Busy pathway during the information fusion process. This design gives the best performance according to our experiments. The exploration of various designs of lateral connections is analyzed in Section V-G3.

V. Experiments

In this section, we first introduce the datasets used and implementation details. Then, we conduct ablation studies to investigate the efficiency and effectiveness while evaluating the parameters for the proposed methodology. Finally, we compare with the state-of-the-art. While the MBPM can be embedded in various deep backbones, in the experiments we consider ResNet50 (R50) with TSM [13], X3D-M [12], MobileNetV2 [56], and ConvNeXt [60] as the backbones of our models. When not specified otherwise, we consider TSM R50 as the default backbone network. Aside from X3D-M [12], the backbone networks are pretrained on ImageNet [61].

A. Datasets

We evaluate our approach on challenging human activities datasets, including Something-Something V1 [3], Kinetics400 [10], UCF101 [28] and HMDB51 [29]. Most videos from Kinetics400 (K400), UCF101 and HMDB51 can be accurately classified by only considering their background scene information, while the temporal relation between frames is not really that important. Meanwhile, in Something-Something (SS) V1, many action categories are more vaguely defined and characterized by symmetrical movements (*e.g.* "Pulling something from left to right" and "Pulling something from right to left"). Discriminating these symmetric actions requires models with strong temporal modeling ability. Since Something-Something is widely used for evaluating temporal



(a) SS V1 (b) UCF101

Fig. 4: Results on Something-Something V1 (SS V1) and UCF101 when varying the scale σ and kernel size $k \times k$ of the spatial channel-wise convolution in MBPM. The results are averages of multiple experiment runs.

modeling effectiveness, we consider this dataset as central when investigating the proposed Busy-Quiet Net (BQN).

B. Training

For training, we utilize the dense sampling strategy [41] for Kinetics400. Meanwhile, for the other datasets, we consider the uniform sampling strategy as shown in Figure 2, where a video is equally divided into N segments, and 3 consecutive frames in each segment are randomly sampled, resulting in a video clip of length $T = 3N$. Unless specified otherwise, a default video clip is composed of $N = 8$ segments with a spatial frame size of 224^2. We train our models on multiple GPUs (NVIDIA Tesla V100), using Stochastic Gradient Descent (SGD) with momentum 0.9 and cosine learning rate schedule. In order to prevent overfitting, we add a dropout layer before the classification layer of each pathway in the BQN model. Following the experimental settings from [13], [23], the learning rate and weight decay parameters for the classification layers are 5 times those for the convolutional layers. Meanwhile, we only apply L2 regularization to the weights in the convolutional and classification layers to avoid overfitting.

1) Hyperparameters for the models using ResNet as backbone: For Kinetics400 [10], the initial learning rate, batch size, total epochs, weight decay and dropout ratio are set to 0.08, 512 (8 samples per GPU), 100, 2e-4 and 0.5, respectively. For Something-Something V1 [3], these hyperparameters are set to 0.12, 256, 50, 8e-4 and 0.8, respectively. We use linear warm-up [62] for the first 7 epochs to overcome early optimization difficulty. When fine-tuning on UCF101 [28] and HMDB51 [29], the models used initially on Kinetics, we freeze all batch normalization layers [59] except for the first one to avoid overfitting, following the recipe in [23]. The initial learning rate, batch size, total epochs, weight decay and dropout ratio are set to 0.001, 64 (4 samples per GPU), 10, 1e-4 and 0.8, respectively.

2) Hyperparameters for models using X3D-M [12] as backbone: For Kinetics400, the initial learning rate, batch size, total epochs, weight decay and dropout ratio are set to 0.4, 256 (16 samples per GPU), 256, 5e-5 and 0.5, respectively. For Something-Something V1, the models are trained from scratch using the following hyperparameters: learning rate 0.2, batch size 256, total epochs 100, weight decay 5e-5 and dropout ratio 0.5. When fine-tuning the models on Kinetics, the initial learning rate, batch size, total epochs, weight decay and dropout ratio are set to 0.12, 256 (16 samples per GPU), 60, 4e-4 and 0.8, respectively.

3) Hyperparameters for models using ConvNeXt [60] as backbone: For Kinetics400, the initial learning rate, batch size, total epochs, weight decay are set to 0.004, 64 (2 samples per GPU), 15, 1e-4, respectively. We do not add any dropout layer to the model in this case.

C. Testing

During testing, we sample a single clip per video with center cropping for efficient inference [13]. When pursuing high accuracy, we consider sampling multiple clips&crops from the video and then averaging the prediction scores of multiple spacetime "views" (spatial crops $\times$ temporal clips) as it was used in [27].

D. Ablation Studies for MBPM

In this section, we conduct ablation studies on multiple datasets to evaluate the best settings for the MBPM, which is described in Section III. We show top-1 and top-5 prediction accuracy (%), and also assess the computational complexity measured in GFLOPs for a single crop & single clip.

The scale σ from Eq. (2) and the kernel size of the spatial channel-wise convolution $LoG_{\sigma}^{1\times k\times k}$ have a significant impact on the performance of MBPM, when embedded in the network. We vary the scale σ and the kernel size, while evaluating the prediction accuracy, to search for the optimal

TABLE I: MBPM *vs*. other motion representation methods, when training on UCF101, SS V1 and K400 datasets and considering ResNet50 [4] as backbone. The additional parameters for the backbone network and the computational complexity (in FLOPs) required by each method are reported. † denotes our reimplementation.

Rep. Method	Efficiency Metrics		UCF101	SS V1	K400
	FLOPs	#Param.			
RGB (baseline)	-	-	87.1	46.5	71.2
RGB Diff. [23]	-	-	87.0	46.6	71.4
TV-L1 Flow [18]	-	-	88.5	37.4	55.7
DI† [19]	-	-	86.2	43.4	68.3
FlowNetC† [22]	444G	39.2M	87.3	26.3	-
FlowNetS† [22]	356G	38.7M	86.8	23.4	-
TVNet† [21]	3.3G	0.2K	88.6	45.2	58.5
PA [24]	2.8G	1.1K	89.5	45.1	57.3
(2+1)D Conv	0.3G	0.2K	86.7	46.6	70.8
MBPM	0.3G	0.2K	**90.3**	**48.0**	**72.3**

TABLE II: Restults on the SS V1 database when considering different CNNs as backbones. ResNet50 and MobileNetV2 have TSM [13] embedded.

CNN backbone	Modality	Pretrain	Seg. (N)	Accuracy
ResNet50 [4]	RGB	ImageNet	8	46.5
	RGB+Flow			49.8
	MBPM			48.0
	RGB+MBPM			50.3
MobileNetV2 [56]	RGB	ImageNet	8	38.7
	MBPM			39.8
X3D-M [12]	RGB	None	16	45.5
	MBPM			46.9

settings. Meanwhile, in order to highlight the importance of the MBPM training, we compare the performance when using the trained MBPM with that of an untrained MBPM, whose kernel weights are fixed and not optimized with the classification loss. The results on SS V1 database are shown in Figure 4(a). We summarize two facts: 1) the most appropriate value for σ in MBPM changes as its kernel size changes; 2) optimizing the parameters for MBPM with the classification loss generally produces higher prediction accuracy. In our preliminary work, we have verified that different datasets share the same optimal settings of MBPM. We search for the optimal settings of the scale σ and kernel size $k \times k$ of MBPM on UCF101 dataset. The results are provided in Figure 4(b). We observe that the experimental results vary greatly under different settings. Nevertheless, the optimal scale found is $\sigma = 1.1$ when setting the kernel size as 9×9, and this is the same as that for the Something-Something V1 dataset. Furthermore, we try a larger kernel of 11×11, but then the results show a performance drop. We speculate that this is due to insufficient training. In the following experiments, we set MBPM in the Busy pathway as trainable with the scale $\sigma = 1.1$ and the kernel size of 9×9, unless specified otherwise.

E. Efficiency and Effectiveness of the MBPM

We draw an apple-to-apple comparison between the proposed MBPM and other motion representation methods [18], [19], [21], [22], [23], [24]. The motion representations produced by these methods are used as inputs to the backbone network and the comparison results are shown in Table I. We follow the experimental settings from [24] for a fair comparison. The backbone network used for all methods is ResNet50 [4]. We use the computer code provided by the original authors for these methods to generate the input for the network. For any of these motion representations, we divide the representation of a video into 8 segments and randomly select one frame from each segment. Following the practices used in the Temporal Segment Network (TSN) [23] and the Persistent Appearance Network (PAN) [24], the output activation of 8 segments is averaged for the final prediction score. In our reimplementation, the approach from [19] generates one dynamic image for every 6 consecutive RGB frames, which consumes the same number of RGB frames as Persistent Appearance [24]. Our MBPM generates one representative frame for every 3 consecutive RGB frames. As for TVNet [21] and TV-L1 Flow [18], the backbone network input is formed by stacking 5 frames of the estimated flow along the channel dimension which requires 6 RGB frames. All models are pretrained on ImageNet. For Something-Something V1 and Kinetics400, we use the hyperparameters specified in Section V-B to train all models. For UCF101, we set the initial learning rate, batch size, total epochs, weight decay and dropout ratio to 0.01, 64 (4 samples per GPU), 80, 1e-4 and 0.5, respectively.

According to the results from Table I, the proposed MBPM outperforms all other motion representation methods by big margins, while its computation requirements are nearly negligible. These results strongly demonstrate the high efficiency and effectiveness of the MBPM. The MBPM is essentially a (2+1)D channel-wise convolution with the special band-pass filtering initialization. In order to confirm that the advantages of MBPM come from the concept of motion band-bass filtering, we compare the MBPM with a randomly initialized (2+1)D channel-wise convolution. According to the results from the bottom of Table I, we can observe that the model with a randomly initialized (2+1)D Conv shows far lower accuracy than the model with the MBPM, which indicates the significance of the proposed motion band bass filtering.

F. Generalization to different CNNs used as backbones

The proposed MBPM is a generic plug-and-play unit. The performance of existing video models could be boosted by simply placing an MBPM after their input layers. In the previous section we have considered ResNet50 [4] as the backbone network. In Table II we provide the results when employing other backbone networks, such as MobileNetV2 [56] and X3D-M [12]. The results indicate steady performance improvements for the resulting networks, after embedding our MBPM. Moreover, we consider different input modalities, and the two-stream fusion of "RGB+MBPM" has higher accuracy

than the fusion of "RGB+Flow," according to the results in Table II.

G. Ablation Studies for Busy-Quiet Net (BQN)

1) BQN vs. Quiet+Busy: In order to evaluate the architecture effectiveness, we compare the proposed BQN with the simple fusion (Quiet+Busy), which mimics the two-stream model [34], by averaging the predictions of the two pathways trained separately. The results from Table III indicate that

TABLE III: Complementarity between Quiet and Busy information. "Quiet" and "Busy" refer to the fact that the Quiet and Busy pathways are trained individually.

Model	Top-1 (%)	Top-5 (%)	GFLOPs
Quiet	46.5	75.3	32.8
Busy	48.0	76.8	32.8
Quiet+Busy	50.3	79.0	65.7
BQN	**51.6**	**80.5**	65.9

the simple fusion of two individual pathways (Quiet+Busy) generates higher top-1 accuracy (50.3%) than the individual pathways, which indicates that the features learned by the Quiet and Busy pathways are complementary. BQN has 51.6% top-1 accuracy, which is 1.3% better than the fusion, Quiet+Busy. The high-performance gain strongly demonstrates the advantages of the proposed BQN architecture.

2) Fusion strategies: In the following we evaluate the performance according to the location in the BQN architecture for fusing Busy and Quiet pathways. Table IV shows the results of

TABLE IV: Fusion Strategies. The fully-connected (fc) layers of the two pathways share their parameters

Fusion Method	Position	Top-1 (%)	Top-5 (%)
Averaging	before fc	50.9	79.8
Averaging	after fc	**51.6**	**80.5**
Max	after fc	50.1	78.7
Concatenation	before fc	51.3	80.2

different fusion strategies. We observe that the average fusion gives the best result among the listed approaches, while the concatenation fusion is second only to the averaging. Besides, placing the average fusion layer after the fully-connected (fc) layer is better than placing it before.

3) Effectiveness of the BPLC: We can set a maximum of up to 16 BPLCs in the BQN architecture when using TSM R50 [13] as the backbone. ResNet50 [4] contains four stages, named res2, res3, res4, res5, respectively. These stages are composed of 3, 4, 6, 3 residual blocks, respectively. For the BPLCs in the stages res2, res3 and res4, we set the spatial kernel size of MBPM as 7×7, and the scale $\sigma = 0.9$. As for stage res5, whose feature size is relatively small, the kernel size is therefore set to 3×3. Table V, illustrates that by adding BPLCs to all processing stages leads to improved performance. From Table VI, we can observe that the model performance improves gradually as the number of BPLCs increases. The substantial performance gains demonstrate the importance of using BPLCs for BQN.

TABLE V: Adding BPLCs to various processing stages of ResNet50 backbone. In each stage, we set one BPLC after its first residual block.

Stages	No. of BPLC	Top-1 (%)	Top-5 (%)
res2	1	49.8	79.1
res2,res3	2	50.1	78.7
res2,res3,res4	3	50.2	79.0
res2,res3,res4,res5	4	50.2	79.2

TABLE VI: The effect of the number of BPLCs.

No. of BPLCs	Top-1 (%)	Top-5 (%)	GFLOPs
0	49.6	78.9	65.7
4	50.2	79.2	65.8
8	50.7	79.7	65.8
16	**51.6**	**80.5**	65.9

4) Lateral Connection (LC) Designs: In order to illustrate the rationality of the proposed BPLC design, we compare it with other LC designs. Various LC designs are illustrated in Figure 3, where LC-I and LC-II are unidirectional, and LC-III is bidirectional. The results from Table VII indicate that the bidirectional design LC-III has higher accuracy than the unidirectional designs LC-I and LC-II. Among the listed designs, the proposed BPLC, which reverses the information fusion direction back and forth alternatively, provides the highest accuracy. We also compare the BPLC with LC-V that does not contain an MBPM. As a result, LC-V shows lower accuracy than the BPLC, which demonstrates the importance of embedding MBPM in the BPLC.

TABLE VII: Various LC designs. 16 LCs are set in the BQN.

Design	Top-1 (%)	Top-5 (%)
LC-I	50.9	79.8
LC-II	50.9	79.7
LC-III	51.5	80.2
BPLC	51.6	80.5
LC-V	51.3	79.9

TABLE VIII: Effect of the spatio-temporal input size. The input size is formatted as width^2 $\times$ time.

Input size for Quiet	Input size for Busy	Top-1 (%)	Top-5 (%)	GFLOPs
$224^2 \times 8$	$224^2 \times 8$	51.6	80.5	65.9
$192^2 \times 8$	$224^2 \times 8$	51.5	79.9	58.0
$160^2 \times 8$	$224^2 \times 8$	51.3	80.1	50.5
$128^2 \times 8$	$224^2 \times 8$	50.7	79.2	44.4
$224^2 \times 8$	$256^2 \times 8$	51.8	80.5	77.1
$192^2 \times 8$	$256^2 \times 8$	51.7	80.2	68.3
$160^2 \times 8$	$256^2 \times 8$	51.7	80.5	60.7
$128^2 \times 8$	$256^2 \times 8$	51.3	79.4	54.6
$160^2 \times 6$	$256^2 \times 8$	49.6	78.3	55.5
$224^2 \times 4$	$224^2 \times 8$	48.7	77.1	49.4

5) Spatial-temporal input size: In BQN, the Busy pathway takes as input the MBPM output, which has the same spatial

TABLE IX: Results on Something-Something V1. † denotes our reimplementation. "N/A" indicates the numbers are not available.

Method	pre-train	Backbone	Frames×Crops×Clips	FLOPs	#Param.	Top-1 (%)	Top-5 (%)
NL I3D GCN [63]	ImageNet	3D R50	32×3×2	303G×3×2	62.2M	46.1	76.8
$ECO_{En}Lite_{RGB+Flow}$ [64]		Inc+3D R18	(92+552)×1×1	N/A	300M	49.5	-
TSN [23]		R50	8×1×1	33G×1×1	-	19.7	46.6
$TRN_{RGB+Flow}$ [65]		BNInception	(8+48)×1×1	N/A	36.6M	42.0	-
TSM_{En} [13]		R50	(16+8)×1×1	98G×1×1	48.6M	49.7	78.5
$TSM_{RGB+Flow}$ [13]		R50	(16+96)×1×1	N/A	48.6M	52.6	81.9
TEA [66]		R50	16×3×10	70G×3×10	24.4M	52.3	81.9
TDN [67]		R101	16×1	132G×1	-	55.3	83.3
SmallBig [68]		-	16×3×2	105G×3×2	-	50.0	79.8
bLVNet-TAM [54]		bLR50	8×1×2	12G×1×2	25M	46.4	76.6
PAN_{Full} [24]		TSM R50	40×1×2	67.7G×1×2	-	50.5	79.2
PAN_{En} [24]		TSM R50	(40+40)×1×2	134G×1×2	-	53.4	81.1
PAN_{En} [24]		TSM R101	(40+40)×1×2	251G×1×2	-	55.3	82.8
ir-CSN [15]	None	3D R101	32×1×10	73.8G×1×10	22.1M	48.4	-
ir-CSN [15]		3D R152	32×1×10	96.7G×1×10	-	49.3	-
TSM R50 [13]	ImageNet	R50	16×1×1	65G×1×1	24.3M	47.2	77.1
BQN		TSM R50	24×1×1	60G×1×1	47.4M	51.7	80.5
BQN		TSM R50	24×3×2	60G×3×2	47.4M	53.3	82.0
BQN		TSM R50	48×3×2	121G×3×2	47.4M	54.3	82.0
BQN		TSM R101	48×3×2	231G×3×2	85.4M	54.9	81.7
X3D-M† [12]	None	-	16×3×2	6.4G×3×2	3.3M	46.7	75.5
BQN	None	X3D-M	48×3×2	9.7G×3×2	6.6M	50.6	79.2
BQN	K400	X3D-M	48×3×2	9.7G×3×2	6.6M	53.7	81.8
$\mathbf{BQN}_{En}$	ImageNet + K400	TSM R101 +X3D-M	(48+48)×3×2	241G×3×2	92M	**57.1**	**84.2**

TABLE X: Comparison results on Kinetics400. We report the inference cost of multiple "views" (spatial crops × temporal clips). † denotes our reimplementation.

Method	Backbone	Frames × views	FLOPs	Top-1 (%)	Top-5 (%)
bLVNet-TAM [54]	bLR50	16×9	561G	72.0	90.6
TSM [13]	R50	16×30	2580G	74.7	-
STM [69]	R50	16×30	2010G	73.7	91.6
X3D-M† [12]	-	16×30	186G	75.1	92.2
TDN [67]	R101	16×30	3960G	**78.5**	**93.9**
$SlowFast_{4\times16}$ [27]	3D R50	32×30	1083G	75.6	92.1
ip-CSN [15]	3D R101	32×30	2490G	76.8	92.5
SmallBigNet [68]	R101	32×12	6552G	77.4	93.3
PAN_{Full}	TSM R50	40×2	176G	74.4	91.6
$I3D_{RGB}$ [10]	Inc. V1	64×N/A	N/A	71.1	89.3
Oct-I3D [25]	-	N/A×N/A	N/A	74.6	-
NL I3D [41]	3D R101	128×30	10770G	77.7	93.3
ConvNeXt-B [60]	-	8×3	367G	75.5	92.4
BQN	TSM R50	48×10	1210G	76.8	92.4
BQN	TSM R50	72×10	1820G	77.3	93.2
BQN	X3D-M	48×30	291G	77.1	92.5
BQN	ConvNeXt-B	24×3	706G	78.0	93.4

TABLE XI: Results on HMDB51 and UCF101. We report the mean class accuracy (%) over the three official splits.

Method	Backbone	HMDB51	UCF101
Codebook-VLAD [70]	-	59.8	-
StNet [71]	R50	-	93.5
TSM [13]	R50	73.5	95.9
STM [69]	R50	72.2	96.2
TEA [66]	R50	73.3	96.9
DI Four-Stream [19]	ResNeXt101	72.5	95.5
TVNet [21]	BNInception	71.0	94.5
$TSN_{RGB+Flow}$ [23]	BNInception	68.5	94.0
$I3D_{RGB+Flow}$ [10]	3D Inception	**80.7**	**98.0**
PAN_{Full} [24]	TSM R50	77.0	96.5
BQN	**TSM R50**	**77.6**	**97.6**

size as the raw video clip, while the temporal size is one-third of the length of the raw video clip. Meanwhile, the Quiet pathway takes as input the complementary component of the MBPM output, expressed through Eq. (4). The results from Table VIII show that with the same temporal size of 8 for the inputs, the spatial size combination of 160^2 and 256^2 for the Quiet and Busy, respectively, has slightly better top-1 accuracy (+0.1%) than the combination of 224^2 and 224^2 but saves 5.2 GFLOPs in computational cost. We also attempt to reduce the temporal input size of the Quiet pathway. However, this would result in a performance drop. One possible explanation is that due to the temporal average pooling in the Quiet pathway, the input's temporal size is already reduced to one-third of the raw video clip. An even smaller temporal size could fail to preserve the correct temporal order of the video, and therefore would

harm the temporal relation modeling.

H. Comparison with the State-of-the-Art

We compare BQN with current state-of-the-art methods on the four datasets. In BQN, the Quiet and Busy pathways' spatial input size is set to 160^2 and 256^2, respectively.

1) Results on Something-Something V1: Table IX summarizes the comprehensive comparison, including the inference protocols, corresponding computational costs (FLOPs) and the prediction accuracy. Our method surpasses all other methods by good margins. For example, the multi-clip accuracy of BQN_{24f}[3] with TSM R50 is 7.2% higher than NL I3D GCN_{32f} [63] while requiring $5\times$ fewer FLOPs. Among the models based on ResNet50, BQN_{48f} has the highest top-1 accuracy (54.3%), which surpasses the second-best, TEA_{16f} [66], by a margin of +2%. Furthermore, our single-clip BQN_{24f} has higher accuracy (51.7%) than most other multi-clip models, requiring only 60 GFLOPs. By adopting a deeper backbone (TSM R101), BQN_{48f} has 54.9% top-1 accuracy, higher than any other model. When using X3D-M as the backbone, BQN achieves the ultimate efficiency, possessing very low redundancy in both feature channel and spatio-temporal dimensions. BQN with X3D-M processes $3\times$ more video frames than vanilla X3D-M, with only 50% additional FLOPs. Compared with TSM R50_{16f}, BQN with X3D-M trained from scratch produces 3.4% higher top-1 accuracy with the computational complexity of 14% of TSM R50_{16f}. The ensemble version BQN_{En} achieves the state-of-the-art top-1/5 accuracy (57.1%/84.2%).

2) Results on Kinetics400, UCF101 and HMDB51: Table X shows the comparison results on Kinetics400. For a fair comparison, we only list the models with the spatial input size of 256^2. BQN_{72f} with TSM R50 achieves 77.3%/93.2% top-1/5 accuracy, which is better than the 3D CNN-based architecture, I3D [10], by a big margin of +6.2%/3.9%. When BQN uses TSM R50 or X3D-M as its backbone, it consistently shows higher accuracy than $\text{SlowFast}_{4\times16}$. Particularly, BQN with X3D-M has 1.5% higher top-1 accuracy than $\text{SlowFast}_{4\times16}$, while requiring $3.7\times$ fewer FLOPs. Meanwhile, BQN_{72f} with TSM R50 is 2.7% better than Oct-I3D [25] for top-1 accuracy. When employing the latest ConvNeXt-B [60] as the backbone network, the BQN model achieves 78% top-1 accuracy, which is higher than ConvNeXt-B [60] by 2.5% and is on par with TDN [67]. The results on two smaller datasets, UCF101 and HMDB51, are shown in Table XI, where we report the mean class accuracy over three official splits. We pretrain our model on Kinetics400 to avoid overfitting. The accuracy of our method is calculated using the inference protocol (3 crops$\times$2 clips). BQN with TSM R50 outperforms most other methods except for $\text{I3D}_{\text{RGB+Flow}}$, which uses an additional optical flow input modality.

I. Visualization analysis

In Figures 5-(1) and (2), we visualize two complex human activities from Kinetics 400, for the "Spinning poi" and "Chopping wood", respectively, where the top row of images shows frames from the video sequence while underneath we visualize the Busy stream selected by MBPM, by representing the mapping of its output. In Figures 6-(1) and (2) we show examples of Busy streams for complex activities characteristic to Something-Something V1 dataset. Meanwhile, in Figures 7-(1) and (2) we present the visualization results of the MBPM output selection for "Biking" sequence from UCF101, and for "Kick" from HMDB51, respectively. We can observe from these examples that the extracted representations are stable when jittering and other minor camera movements are present in the videos. MBPM not only that suppresses the stationary information and the background movement, but it also highlights the boundaries of moving objects and regions, which are of vital importance for action discrimination. For example, in the "spinning poi" video, showing the movement of an illuminating object from Figure 5-(1), MBPM highlights well the poi's movement rather than the movement of the background or the performer.

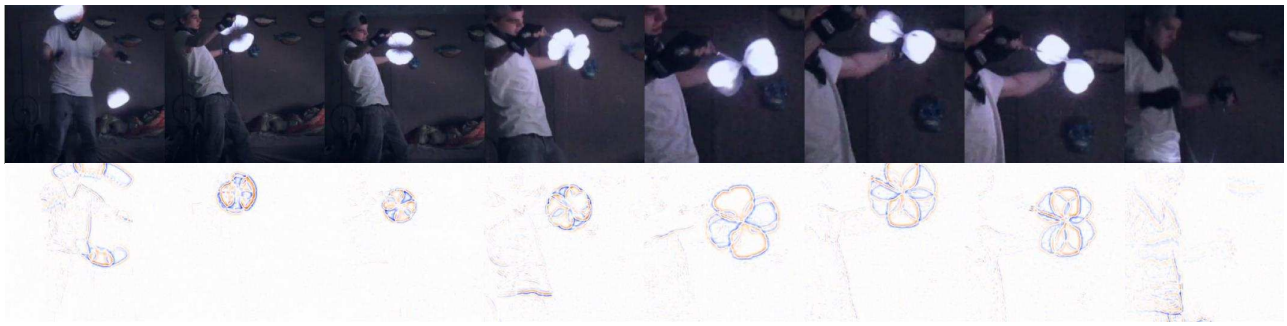
In order to visually observe the differences between the outputs of our MBPM and other motion representation methods, we show in Figure 8 some example video frames and their corresponding motion representations generated by different methods . The optical flow estimated by TV-L1 [18], displays the instantaneous velocity and direction of movement in every position, where the color represents the direction of movement and the brightness represents the absolute value of instantaneous velocity in a position. In contrast, TVNet [21], Persistent Appearance [24] and MBPM define the visual information from the boundaries of moving regions. From the results in Figure 8, we can observe that the textures captured by TVNet, Persistent Appearance and MBPM do not present obvious differences. However, MBPM has a simpler structure and requires less computation.

In Figure 9, we visualize the kernel of the spatial convolution $LoG_{\sigma}^{1\times k\times k}$ of MBPM in the Busy pathway. We can observe that kernels always present a similar shape to a Mexican hat function either before or after training, In Figure 10, we visualize the first channel of 64 filters in the first layers of the BQN and the baseline (TSM ResNet50). We can observe that the Busy and Quiet pathways' filters have quite distinct shapes for their kernels, indicating that the Busy and Quiet pathways learn different types of features after training.

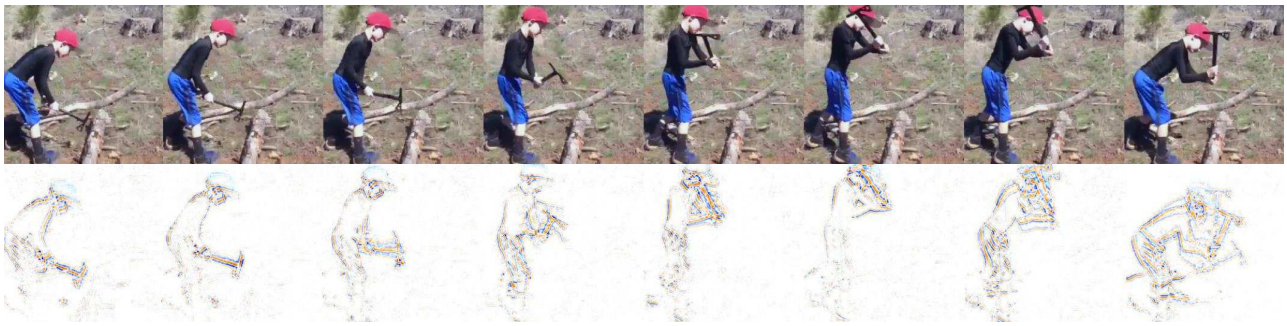
VI. Conclusion

A novel video representation learning method, decomposing video streams into Busy and Quiet information streams, is proposed in this paper for action recognition. For this aim we propose the Motion Band-Pass Module (MBPM) which, following training, defines different spatio-temporal frequency bands for the Busy and Quiet information in the video data. MBPM captures important motion cues for action recognition, such as those characterizing regions of movement variation or the boundaries of moving objects in video. Enabled by MBPM, we design an efficient and effective two-stream spatio-temporal processing architecture called Busy-Quiet Net (BQN), to separately process Busy and Quiet video

[3]The subscript 24f indicates that video clips of 24 frames are used for experiments.



(1) Spinning poi



(2) Chopping wood

Fig. 5: Example videos and their corresponding MBPM outputs from Kinetics 400.

data information. The proposed two-stream video processing network, besides disentangling the video information for better recognition, allows for a better allocation of the computational resources, where more processing power is used for the Busy stream and less for the Quiet. Besides action recognition the proposed Busy and Quiet video disentanglement can also be used for video analysis in various applications.

ACKNOWLEDGEMENT

Dr. A. G. Bors thanks for the partial support from the project COUSIN, funded by EPSRC.

REFERENCES

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman, "The kinetics human action video dataset," *arXiv preprint arXiv:1705.06950*, 2017.

[3] R. Goyal, S. E. Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fruend, P. Yianilos, M. Mueller-Freitag, F. Hoppe, C. Thurau, I. Bax, and R. Memisevic, "The "something something" video database for learning and evaluating visual common sense," in *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, 2017, pp. 5842–5850.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conference Computer Vision Pattern Recog. (CVPR)*, 2016, pp. 770–778.

[5] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, Inception-ResNet and the impact of residual connections on learning," in *Proc. AAAI Conference on Artificial Intelligence*, 2017, pp. 4278–4284.

[6] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conference Computer Vision Pattern Recog. (CVPR)*, 2017, pp. 4700–4708.

[7] S. Gao, M. Cheng, K. Zhao, X. Zhang, M. Yang, and P. Torr, "Res2Net: A new multi-scale backbone architecture," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 43, no. 2, pp. 652–662, 2019.

[8] G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler, "Convolutional learning of spatio-temporal features," in *Proc. European Conference Computer Vision (ECCV), vol. LNCS 6316*, 2010, pp. 140–153.

[9] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conference Computer Vision (ICCV)*, 2015, pp. 4489–4497.

[10] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *Proc. IEEE Conference Computer Vision Pattern Recog. (CVPR)*, 2017, pp. 4724–4733.

[11] K. Hara, H. Kataoka, and Y. Satoh, "Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and ImageNet?" in *Proc. IEEE Conference Computer Vision Pattern Recog. (CVPR)*, 2018, pp. 6546–6555.

[12] C. Feichtenhofer, "X3D: Expanding architectures for efficient video recognition," in *Proc. IEEE Conference Computer Vision Pattern Recognition (CVPR)*, 2020, pp. 203–213.

[13] J. Lin, C. Gan, and S. Han, "Tsm: Temporal shift module for efficient video understanding," in *Proc. IEEE Int. Conference Computer Vision (ICCV)*, 2019, pp. 7083–7093.

[14] Z. Qiu, T. Yao, and T. Mei, "Learning spatio-temporal representation with pseudo-3d residual networks," in *Proc. IEEE Int. Conference Computer Vision (ICCV)*, 2017, pp. 5533–5541.

[15] D. Tran, H. Wang, L. Torresani, and M. Feiszli, "Video classification with channel-separated convolutional networks," in *Proc. IEEE Int. Conference Computer Vision (ICCV)*, 2019, pp. 5552–5561.

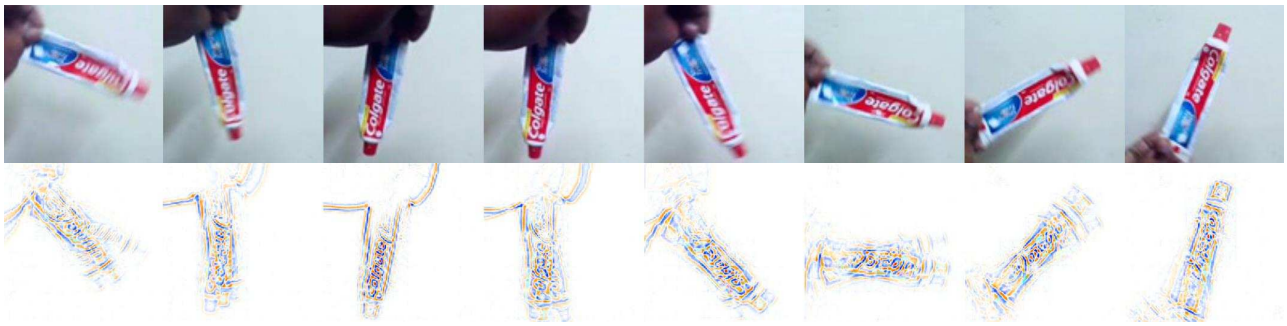
[16] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," in *Proc. IEEE Conference Computer Vision Pattern Recog. (CVPR)*, 2018, pp. 6450–6459.

[17] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy, "Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification," in *Proc. European Conference Computer Vision (ECCV), vol. LNCS 11219*, 2018, pp. 305–321.

[18] C. Zach, T. Pock, and H. Bischof, "A duality based approach for realtime tv-l 1 optical flow," in *Proc. Joint Pattern Recog. Symp., vol. LNCS 4713*, 2007, pp. 214–223.

[19] H. Bilen, B. Fernando, E. Gavves, and A. Vedaldi, "Action recognition with dynamic image networks," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 40, no. 12, pp. 2799–2813, 2018.

(1) Pretending to take something out of something



(2) Pretending to turn something upside down

Fig. 6: Example Videos and their corresponding MBPM outputs from Something-Something V1.

[20] J. Wang, A. Cherian, F. Porikli, and S. Gould, "Video representation learning using discriminative pooling," in *Proc. IEEE Conference Computer Vision Pattern Recog. (CVPR)*, 2018, pp. 1149–1158.

[21] L. Fan, W. Huang, C. Gan, S. Ermon, B. Gong, and J. Huang, "End-to-end learning of motion representation for video understanding," in *Proc. IEEE Conference Computer Vision Pattern Recognition (CVPR)*, 2018, pp. 6016–6025.

[22] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "Flownet 2.0: Evolution of optical flow estimation with deep networks," in *Proc. IEEE Conference Computer Vision Pattern Recog. (CVPR)*, vol. 2, 2017, pp. 2462–2470.

[23] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal segment networks: Towards good practices for deep action recognition," in *Proc. European Conference Computer Vision (ECCV), vol LNCS 9912*, 2016, pp. 20–36.

[24] C. Zhang, Y. Zou, G. Chen, and L. Gan, "PAN: Persistent appearance network with an efficient motion cue for fast action recognition," in *Proc. ACM Int. Conference Multimedia*, 2019, pp. 500–509.

[25] Y. Chen, H. Fan, B. Xu, Z. Yan, Y. Kalantidis, M. Rohrbach, S. Yan, and J. Feng, "Drop an octave: Reducing spatial redundancy in convolutional neural networks with octave convolution," in *Proc. IEEE Int. Conference Computer Vision (ICCV)*, 2019, pp. 3435–3444.

[26] C. Chen, Q. Fan, N. Mallinar, T. Sercu, and R. Feris, "Big-little net: An efficient multi-scale feature representation for visual and speech recognition," in *Proc. Int. Conf. of Learning Representations (ICLR), arXiv preprint arXiv:1807.03848*, 2019.

[27] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "Slowfast networks for video recognition," in *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, 2019, pp. 6202–6211.

[28] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," *arXiv preprint arXiv:1212.0402*, 2012.

[29] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: a large video database for human motion recognition," in *Proc. IEEE Int. Conference Computer Vision (ICCV)*, 2011, pp. 2556–2563.

[30] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances Neural Information Process. Systems (NIPS)*, 2012, pp. 1097–1105.

[31] 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. Computer Vision Pattern Recog. (CVPR)*, 2015, pp. 1–9.

[32] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conference Mach. Learn. (ICML), vol. PMLR 97*, 2019, pp. 6105–6114.

[33] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proc. IEEE Conference Computer Vision Pattern Recog. (CVPR)*, 2014, pp. 1725–1732.

[34] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances Neural Information Processing Systems (NIPS)*, 2014, pp. 568–576.

[35] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[36] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond short snippets: Deep networks for video classification," in *Proc. IEEE Conference Computer Vision Pattern Recog. (CVPR)*, 2015, pp. 4694–4702.

[37] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *Proc. IEEE Conference Computer Vision Pattern Recog. (CVPR)*, 2015, pp. 2625–2634.

[38] D. Li, Z. Qiu, Q. Dai, T. Yao, and T. Mei, "Recurrent tubelet proposal and recognition networks for action detection," in *Proc. European Conference Computer Vision (ECCV). vol. LNCS 11210*, 2018, pp. 303–318.

[39] Z. Li, K. Gavrilyuk, E. Gavves, M. Jain, and C. G. Snoek, "Videolstm convolves, attends and flows for action recognition," *Computer Vision and Image Understanding*, vol. 166, pp. 41–50, 2018.

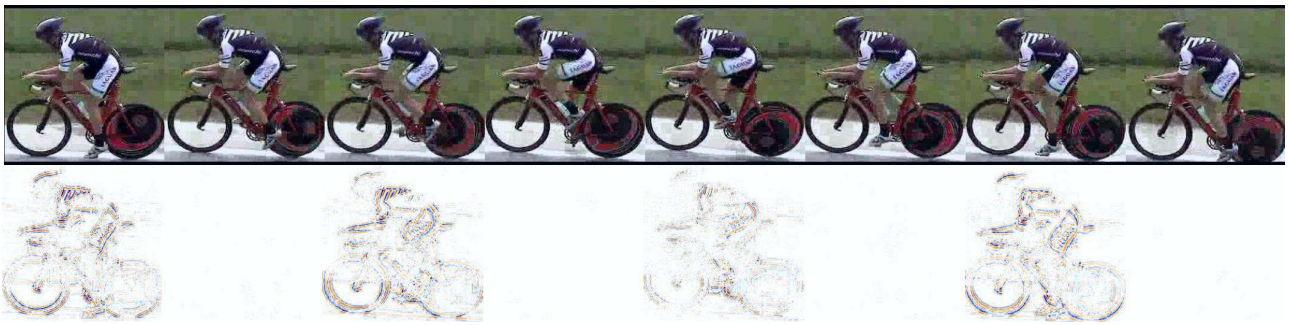
[40] L. Sun, K. Jia, K. Chen, D. Yeung, B. E. Shi, and S. Savarese, "Lattice long short-term memory for human action recognition," in *Proc. IEEE Int. Conference Computer Vision (ICCV)*, 2017, pp. 2147–2156.

[41] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE Conference Computer Vision Pattern Recog. (CVPR)*, 2018, pp. 7794–7803.

[42] Y. Cao, J. Xu, S. Lin, F. Wei, and H. Hu, "GCNet: Non-local networks meet squeeze-excitation networks and beyond," in *Proc. IEEE Int. Conference Computer Vision Workshops (ICCV-w)*, 2019.

[43] G. Huang and A. G. Bors, "Region-based non-local operation for video classification," in *Proc. Int. Conference on Pattern Recognition (ICPR)*, 2021, pp. 10 010–10 017.

[44] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *Proc. IEEE Conference Computer Vision Pattern Recog. (CVPR)*, 2016, pp. 1933–1941.



(1) Biking

(2) Kick

Fig. 7: Example Videos and their corresponding MBPM outputs. The videos (1)-(2) are randomly picked from UCF101 and HMDB51, respectively.

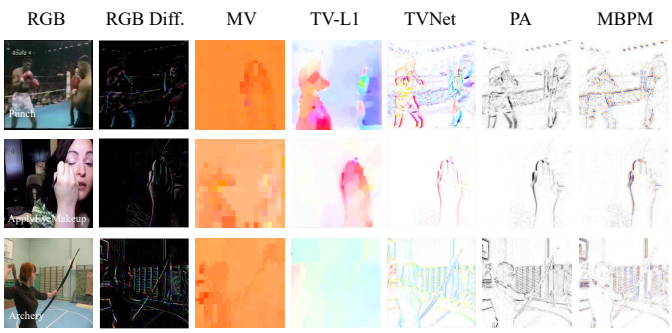
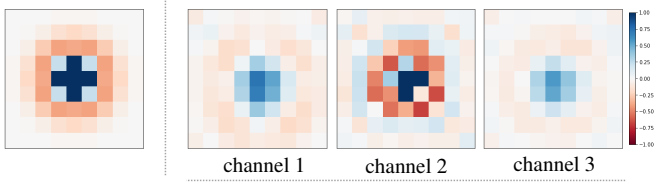


Fig. 8: Comparison between visualizations of different motion representations, including RGB Difference [23], Motion Vector (MV) [51], the optical flow extracted by the TV-L1 algorithm [18], TVNet [21], Persistent Appearance (PA) [24] and the proposed MBPM on the UCF101 dataset after training. Best viewed in color and zoomed in.



(a) Untrained LoG (b) Trained LoG

Fig. 9: Visualization of the spatial channel-wise convolution $LoG_{\sigma}^{1\times k\times k}$ of MBPM in the Busy pathway before and after training on Kinetics400. The 9×9 channel-wise convolution is initialized with a Laplacian of Gaussian with the scale parameter $\sigma = 1.1$. Best viewed in color and zoomed in.

[45] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *Proc. IEEE Int. Conference Computer Vision (ICCV)*, 2015, pp. 2758–2766.

[46] D. Sun, D. Vlasic, C. Herrmann, V. Jampani, M. Krainin, H. Chang, R. Zabih, W. Freeman, and C. Liu, "AutoFlow: Learning a better training set for optical flow," in *Proc. IEEE/CVF Conf. on Computer Vision Pattern Recog. (CVPR)*, 2021, pp. 10 093–10 102.

[47] J. Y.-H. Ng, J. Choi, J. Neumann, and L. S. Davis, "ActionFlowNet: Learning motion representation for action recognition," in *Proc. IEEE Winter Conf. on Applications of Computer Vision (WACV)*, 2018, pp. 1616–1624.

[48] Y. Zhu, Z. Lan, S. Newsam, and A. Hauptmann, "Hidden two-stream convolutional networks for action recognition," in *Proc. Asian Conf. on Computer Vision (ACCV). vol. LNCS 11363*, 2018, pp. 363–378.

[49] S. Sun, Z. Kuang, L. Sheng, W. Ouyang, and W. Zhang, "Optical flow guided feature: A fast and robust motion representation for video action recognition," in *Proc. IEEE Conference Computer Vision Pattern Recog. (CVPR)*, 2018, pp. 1390–1399.

[50] A. Piergiovanni and S. Ryoo, "Representation flow for action recognition," in *Proc. IEEE Conference Computer Vision Pattern Recog. (CVPR)*, 2019, pp. 9945–9953.

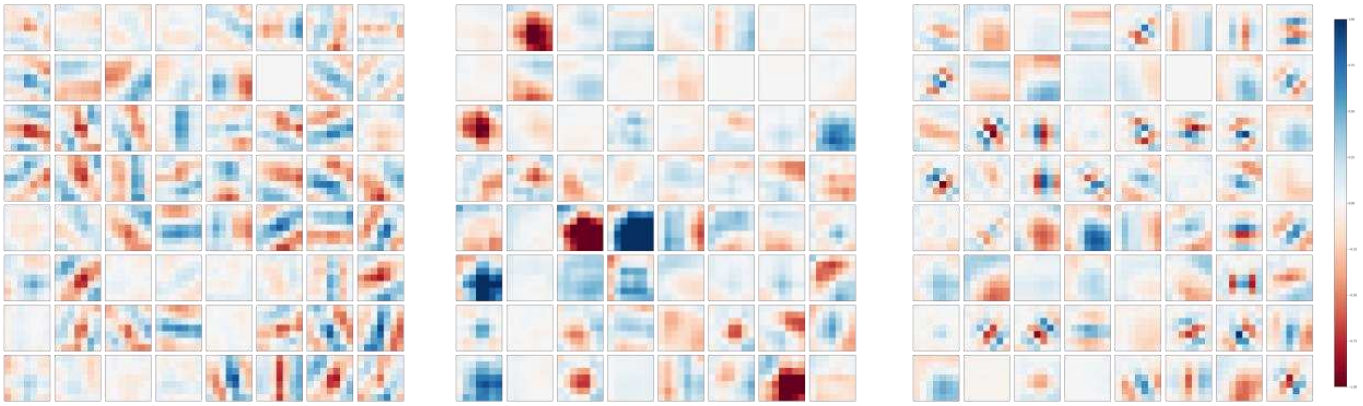
[51] B. Zhang, L. Wang, Z. Wang, Y. Qiao, and H. Wang, "Real-time action recognition with enhanced motion vector CNNs," in *Proc. IEEE Conference Computer Vision Pattern Recog. (CVPR)*, 2016, pp. 2718–2726.

[52] ——, "Real-time action recognition with deeply transferred motion vector CNNs," *IEEE Transactions on Image Processing*, vol. 27, no. 5, pp. 2326–2339, 2018.

[53] G. Huang and A. G. Bors, "Learning spatio-temporal representations with temporal squeeze pooling," in *Proc. IEEE Int. Conference on Acoustics, Speech and Signal Proc. (ICASSP)*, 2020, pp. 2103–2107.

[54] Q. Fan, C. Chen, H. Kuehne, M. Pistoia, and D. Cox, "More is less: Learning efficient video representations by big-little network and depth-wise temporal aggregation," in *Advances Neural Information Process. Systems (NIPS)*, 2019, pp. 2264–2273.

[55] G. Huang and A. G. Bors, "Busy-quiet video disentangling for video classification," in *Proc. IEEE/CVF Winter Conf. on Applications of*



(a) Busy pathway (b) Quiet pathway (c) TSM ResNet50

Fig. 10: Visualization of the first channels of the 64 conv1 filters of BQN after training on Kinetics400. All 64 filters have a size of 7×7. From left to right, in (a), (b) and (c), we respectively present the trained conv1 filters in the Busy pathway, Quiet pathway and TSM ResNet50. We observe that the kernels of the 64 filters in the Busy pathway display stripe-like shapes, consistent with band-pass filters, while those for the filters in the Quiet pathway are more like larger blobs. The conv1 in TSM ResNet50 (baseline) contains both types of filters from the Busy and Quiet pathways. Best viewed in color and zoomed in.

Computer Vision (WACV), 2022, pp. 756–765.

[56] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE Conference on Computer Vision Pattern Recog. (CVPR)*, 2018, pp. 4510–4520.

[57] C. Feichtenhofer, A. Pinz, and R. Wildes, "Spatiotemporal residual networks for video action recognition," in *Advances in Neural Information Processing Systems (NIPS)*, 2016, pp. 3468–3476.

[58] T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conference Computer Vision Pattern Recog. (CVPR)*, 2017, pp. 2117–2125.

[59] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conference Mach. Learn. (ICML), vol. PMLR 37*, 2015, p. 448–456.

[60] Z. Liu, H. Mao, C. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," in *Proc. IEEE Conference Computer Vision Pattern Recog. (CVPR), arXiv preprint arXiv:2201.03545*, 2022.

[61] 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 Conference Computer Vision Pattern Recog. (CVPR)*, 2009, pp. 248–255.

[62] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," *Int. Conference Learn. Representations (ICLR), arXiv preprint arXiv:1608.03983*, 2017.

[63] X. Wang and A. Gupta, "Videos as space-time region graphs," in *Proc. European Conference Computer Vision (ECCV), vol LNCS 11209*, 2018, pp. 413–431.

[64] M. Zolfaghari, K. Singh, and T. Brox, "ECO: Efficient convolutional network for online video understanding," in *Proc. European Conference Computer Vision (ECCV), vol. LNCS 11206*, 2018, pp. 695–712.

[65] B. Zhou, A. Andonian, A. Oliva, and A. Torralba, "Temporal relational reasoning in videos," in *Proc. European Conference Computer Vision (ECCV), vol LNCS 11205*, 2018, pp. 803–818.

[66] Y. Li, B. Ji, X. Shi, J. Zhang, B. Kang, and L. Wang, "TEA: temporal excitation and aggregation for action recognition," in *Proc. IEEE/CVF Conference Computer Vision Pattern Recog. (CVPR)*, 2020, pp. 909–918.

[67] L. Wang, Z. Tong, B. Ji, and G. Wu, "TDN: temporal difference networks for efficient action recognition," in *Proc. IEEE/CVF Conference Computer Vision Pattern Recognition (CVPR)*, 2021, pp. 1895–1904.

[68] X. Li, Y. Wang, Z. Zhou, and Y. Qiao, "SmallBigNet: integrating core and contextual views for video classification," in *Proc. IEEE/CVF Conference Computer Vision Pattern Recognition (CVPR)*, 2020, pp. 1092–1101.

[69] B. Jiang, M. Wang, W. Gan, W. Wu, and J. Yan, "STM: spatiotemporal and motion encoding for action recognition," in *Proc. IEEE/CVF Int. Conference Computer Vision (ICCV)*, 2019, pp. 2000–2009.

[70] Z. Wang, Y. Wang, L. Wang, and Y. Qiao, "Codebook enhancement of vlad representation for visual recognition," in *Proc. IEEE Int. Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 1258–1262.

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

Guoxi Huang is currently a fourth-year Ph.D. student in computer science at the University of York, U.K. He received the B.Sc. degree from South China Institute of Software Engineering, Guangzhou University, Guangzhou, China, in 2016, the M.Sc. degree in Artificial Intelligence from the University of Edinburgh, U.K., in 2017. His research interests include video understanding, action recognition and image processing.

Adrian G. Bors (Senior Member, IEEE) received the M.Sc. degree in electronics engineering from the Polytechnic University of Bucharest, Bucharest, Romania, in 1992, and the Ph.D. degree in informatics from the University of Thessaloniki, Thessaloniki, Greece, in 1999. In 1999 he joined the Department of Computer Science, Univ. of York, U.K., where he is currently an Associate Professor. Dr. Bors was a Research Scientist at Tampere Univ. of Technology, Finland, a Visiting Scholar at the Univ. of California at San Diego (UCSD), and an Invited Professor at the Univ. of Montpellier, France. Dr. Bors has authored and co-authored more than 150 research papers including 34 in journals. His research interests include computer vision, computational intelligence and image processing.

Dr. Bors was a member of the organizing committees for IEEE WIFS 2021, IPTA 2020, IEEE ICIP 2018, 2001, BMVC 2016, IPTA 2014, and CAIP 2013. He was an Associate Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING from 2010 to 2014 and the IEEE TRANSACTIONS ON NEURAL NETWORKS from 2001 to 2009. He was a Co-Guest Editor for a special issue on Machine Vision for the International Journal for Computer Vision in 2018 and the Journal of Pattern Recognition in 2015.