Expert Systems With Applications 147 (2020) 113203

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Data-level information enhancement: Motion-patch-based Siamese Convolutional Neural Networks for human activity recognition in videos

Yujia Zhang[a,*], Lai Man Po[a], Mengyang Liu[a], Yasar Abbas Ur Rehman[a,b], Weifeng Ou[a], Yuzhi Zhao[a]

[a] Department of Electrical Engineering, City University of Hong Kong, Hong Kong, China
[b] TCL Corporate Research Co. Limited, Hong Kong, China

ARTICLE INFO

Article history:
Received 17 July 2019
Revised 11 January 2020
Accepted 13 January 2020
Available online 14 January 2020

Keywords:
Human activity recognition
Data augmentation
Deep learning
3D Convolutional Neural Networks

ABSTRACT

Data augmentation is critical for deep learning-based human activity recognition (HAR) systems. However, conventional data augmentation methods, such as random-cropping, may generate bad samples that are unrelated to a particular activity (*e.g.* the background patches without saliency motion information). As a result, the random-cropping based data augmentation may affect negatively the overall performance of HAR systems. Humans, in turn, tend to pay more attention to motion information when recognizing activities. In this work, we attempt to enhance the motion information in HAR systems and mitigate the influence of bad samples through a Siamese architecture, termed as Motion-patch-based Siamese Convolutional Neural Network (MSCNN). The term motion patch is defined as a specific square region that includes critical motion information in the video. We propose a simple yet effective method for selecting those regions. To evaluate the proposed MSCNN, we conduct a number of experiments on the popular datasets UCF-101 and HMDB-51. The mathematical model and experimental results show that the proposed architecture is capable of enhancing the motion information and achieves comparable performance.

1. Introduction

Human activity recognition (HAR) describes the recognition of the actions and goals of one or more agents through a series of observations on the actions and the environmental conditions (Gottfried & Aghajan, 2009). Video-based HAR is a popular topic in the field of image processing since its applications cover the industries of automatic surveillance, health care, human-computer interaction, robot learning and so on (Mabrouk & Zagrouba, 2018; Ranasinghe, Al Machot, & Mayr, 2016; Turaga, Chellappa, Subrahmanian, & Udrea, 2008). Recently, due to the great success of AlexNet (Krizhevsky, Sutskever, & Hinton, 2012) in the image classification competition ILSVRC 2012, deep Convolutional Neural Networks (CNN) has been widely used in many computer vision tasks including HAR. With the appearance of video-based activity recognition datasets (Caba Heilbron, Escorcia, Ghanem, & Carlos Niebles, 2015; Karpathy et al., 2014; Kay et al., 2017; Kuehne, Jhuang, Stiefelhagen, & Serre, 2013; Soomro, Zamir, & Shah, 2012), and deep CNN technologies such as data augmentation, drop out (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014), batch normalization (Ioffe & Szegedy, 2015), residual network (He, Zhang, Ren, & Sun, 2016), deep CNN have become a dominant research method in video-based HAR and achieved the best performance in this field (Carreira & Zisserman, 2017; Crasto, Weinzaepfel, Alahari, & Schmid, 2019; Qiu, Yao, & Mei, 2017; Tran et al., 2018; Yi, Zheng, & Lin, 2017; Zhu, Zhu, & Zou, 2018).

Research on visual attention mechanism in the human brain has been conducted for several decades (Tsotsos, 1990; Tsotsos et al., 1995). It has been noted that our visual system has a selection of a region of interest (ROI) in the field of view when recognizing. Moreover, it has been noted that features associated with important motion regions will lead to a more discriminative action representation (Ni, Moulin, Yang, & Yan, 2015). Inspired by these findings, we propose to shepherd the neural network by selecting motion patches in the sequence of frames, such that the network focus more on critical motion information and

* Corresponding author.
E-mail addresses: yzhang2383-c@my.cityu.edu.hk (Y. Zhang), eelmpo@cityu.edu.hk (L. Man Po), mengyaliu7-c@my.cityu.edu.hk (M. Liu), yasar.abbas@my.cityu.edu.hk (Y.A. Ur Rehman), weifengou2-c@my.cityu.edu.hk (W. Ou), yzzhao2-c@my.cityu.edu.hk (Y. Zhao).

https://doi.org/10.1016/j.eswa.2020.113203
0957-4174/

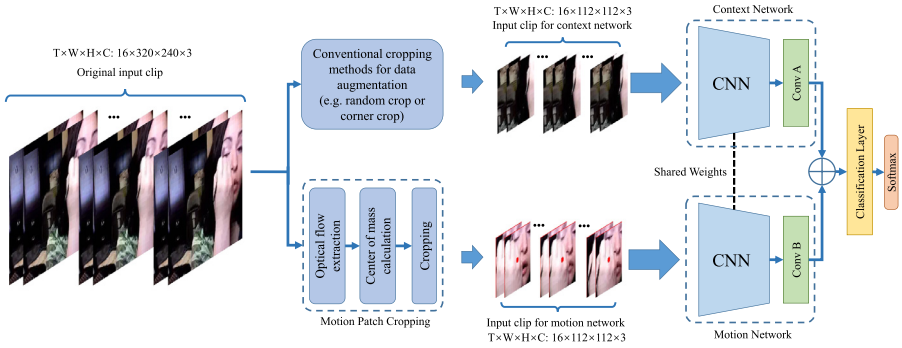

Fig. 1. An overview of proposed MSCNN: MSCNN is made of two partially-weight-shared branches of CNN named of context network and motion network. The context network is fed with the clip made by conventional data augmentation method so as to extract the whole features of the input video while the motion network is fed with a motion-patch-cropped clip so as to extract the motion features of the input video. Then the two features are combined by a feature combination method for classification.

less on the background regions. Here we define motion patch as a fixed-sized square region containing key motion information of a frame in a video. Through the attention mask results of Meng et al. (2018), Sharma, Kiros, and Salakhutdinov (2015) and the visualization of neural networks in Tran, Bourdev, Fergus, Torresani, and Paluri (2015), we found that most of the attention masks and activated regions in the visualization are the regions containing significant motion. These findings motivate us that paying attention to the motion information when recognizing actions, similar to our humans, could benefit our machine-based action recognition.

In view of the above-mentioned findings, we propose a Motion-patch-based Siamese Convolutional Neural Network (MSCNN) which is a new attempt to realize feature enhancement at the data level. In addition, we propose a motion patch extraction scheme based on optical flow for extracting motion patches, which include the pattern of apparent motion of objects, surfaces and edges in a visual scene of a neural network. The proposed architecture of MSCNN is shown in Fig. 1, which consists of two partially shared-weight neural networks. The two branches of networks are named as context network and motion network, which are connected to a classification layer to form a Siamese network architecture. There are two different inputs – one from the entire image and the other from the extracted motion patches, which are fed into MSCNN simultaneously to generate two different bags of features. The features are processed by the features combination method and predicted by a classification layer. Similar ideas can be found in Karpathy et al. (2014), Ren, He, Girshick, and Sun (2015) and Tu et al. (2018). Karpathy et al. (2014) highlighted multi-resolution two-stream neural networks for reducing computing expense while keeping the performance. Ren et al. (2015) proposed region proposal networks to select effective bounding boxes along with region-based CNN (Girshick, 2015) to achieve higher performance.

The architecture proposed in our work has two advantages. First, we can focus more on the non-trivial motion information by inputting motion patches, which we call motion information enhancement. The second advantage is that this can also be considered as a special kind of data augmentation for the HAR system. In HAR systems, training process always utilizes random cropping-based data augmentation methods to reduce over-fitting (Krizhevsky et al., 2012; Wang et al., 2016), however, these random cropping-based data augmentation method may generate motion-unrelated samples or bad samples, such as blue sky or playgrounds shown in many different kinds of activities, which affects the performance of HAR systems. The proposed MSCNN has the ability to alleviate the adverse impact of the bad samples from random cropping-based data augmentation and boost the performance of each HAR system which is based on random cropping as the data augmentation method since the network reads the motion information at each iteration when training and performing activity recognition. To evaluate the proposed method, we conduct a number of experiments on two popular data sets: UCF-101 and HMDB-51. The experimental results demonstrate that our proposed method can enhance non-trivial motion information and the performance is competitive to the state-of-the-art HAR methods.

The main contributions of this study are five folds: (1) The proposed MSCNN alleviates the adverse impact of random cropping-based data augmentation methods of HAR systems; (2) We propose an innovative data-level motion information enhancement and provide experimental results to testify its effectiveness; (3) The proposed method is an end-to-end training method without training time-consuming features such as optical flow or other hand-crafted features and without extra labels of motion-related regions; (4) We propose a novel cropping method that can accurately extract motion patches from videos. (5) The proposed method can achieve superior results than other counterparts on two popular action recognition benchmarks UCF-101 and HMDB-51.

The remainder of this paper is organized as follows. In Section 2, we introduce related research work on HAR. In Section 3, we demonstrate our proposed motion patch extraction method and the model of our proposed networks in detail. The implemental details and extensive ablation studies are elaborated in Section 4. Finally, our work is concluded in Section 5.

2. Related work

Video-based HAR has witnessed an explosive growth in deep CNN in recent years. In this section, we will cover the work related to our approach, such as the development of CNN in HAR, attention mechanism, as well as Siamese neural networks.

2.1. Convolutional Neural Networks for HAR

Prior to the CNN-based HAR methods, the traditional HAR methods were based on hand-crafted features, features combination, and classification methods (Liu, Luo, & Shah, 2009; Niebles, Chen, & Fei-Fei, 2010; Rahman, Song, Leung, Lee, & Lee, 2014; Varol & Salah, 2015; Wang, Ullah, Klaser, Laptev, & Schmid, 2009). With the development of CNN technologies, CNN-based methods have been fully applied in HAR (Carreira & Zisserman, 2017; Donahue et al., 2015; Feichtenhofer, Pinz, & Wildes, 2016a; 2017; Hernández-García, Ramos-Cózar, Guil, García-Reyes, & Sahli, 2018; Wang et al., 2016; Zhou, Sun, Zha, & Zeng, 2018), not only because of improvement of accuracy but also because of the generalization. Among these methods, the main concern is how to

extract discriminative spatial and temporal features from the video and how to well associate those features. The most influential CNN-based HAR network technology can be attributed to four major families – LSTM (Donahue et al., 2015), two-stream network (Simonyan & Zisserman, 2014), 3D CNN (Ji, Xu, Yang, & Yu, 2013; Tran et al., 2015) and feature reinforcement architecture (attention mechanism, Bahdanau, Cho, & Bengio, 2014; non-local structure, Wang, Girshick, Gupta, & He, 2018b, etc.). Most newly created CNN-based HAR algorithms are almost composition of those four families (Carreira & Zisserman, 2017; Feichtenhofer, Pinz, & Zisserman, 2016b; He et al., 2018; Meng et al., 2018; Qiu et al., 2017; Song, Lan, Xing, Zeng, & Liu, 2018; Wang, Cherian, Porikli, & Gould, 2018a; Wang et al., 2016; Zhou et al., 2018). Wang et al. (2016) proposed a long-term Temporal Segment Network (TSN) and video-level prediction based on two-stream methods. Carreira and Zisserman (2017) proposed a Two-Stream Inflated 3D ConvNet (I3D) that combined two streams with 3D CNN together and achieved the highest accuracy at that time. Song et al. (2018) proposed an LSTM with spatial attention and temporal attention for extracting spatial-temporal features of 3D video HAR. Wang et al. (2018b) proposed a non-local structure that enabled the CNN to consider different features in the neural network. In those CNN-based methods, random cropping-based data augmentation methods were comprehensively utilized. Though this kind of data augmentation methods increased the sample diversity so as to improve the performance, there still exists a problem with the generation of bad samples.

2.2. Attention mechanism for HAR

Attention mechanism has been studied for more than a decade (Borji & Itti, 2013; Firat, Cho, & Bengio, 2016; Harel, Koch, & Perona, 2007; Itti, Koch, & Niebur, 1998; Koch & Ullman, 1987). Since Bahdanau et al. (2014) have achieved a great success by first applying attention mechanism on neural networks to machine translation, attention mechanisms have been implemented in many computer vision tasks (Vaswani et al., 2017; Xu et al., 2015; Yang, He, Gao, Deng, & Smola, 2016). Xu et al. (2015) proposed attention-based model with Recurrent Neural Networks (RNN) for the generation of image captions, which gave the state-of-the-art performance at that time. Yang et al. (2016) proposed a stacked attention network for image question answering tasks, which were capable of selecting multiple regions of interest from the question text. Vaswani et al. (2017) proposed a novel network architecture named Transformer based on the attention mechanism for machine translation.

Attention mechanism applied to HAR tasks has been shown to be an effective method to improve the recognition accuracy (Sharma et al., 2015; Song et al., 2018; Wang et al., 2018b). Sharma et al. (2015) proposed a soft attention-based model in multi-layered RNN and proved that the attention model tends to recognize important elements in video frames. Song et al. (2018) proposed spatiotemporal attention-based Long Short-Term Memory (LSTM) networks for 3D HAR, which embraces two attention mechanisms for spatial and temporal attention with different levels of importance for different joints and different frames.

Among these attention mechanism-based methods, we found that these attention mechanisms are at the structural level. In other words, the attention of these methods is learned by the networks. Inspired by these works, we argue that the attention mechanism through input data still can benefit the network. According to the theory of Tsotsos et al. (1995), primates tend to select regions of interest when recognizing named selective attention. The main challenge is how to feed them with selected regions. Our motivation comes from the scenario of teaching children to recognize human actions. It would be better when we hold a picture where a man running on the ground on the one hand while the other hand holds a picture that crops the runner to teach recognizing than just holding one entire picture. This idea is the base of the proposed MSCNN, where the context network represents a hand holding an entire picture, the motion network represents the hand holding the critical motion region.

2.3. Siamese neural networks for HAR

Siamese neural networks (SNN) were first proposed for binary classification tasks, such as signature recognition and fingerprint recognition (Baldi & Chauvin, 1993; Bromley, Guyon, Lecun, & Shah, 1993), where SNN were used for metric learning. Then the idea extended to various tasks (Bertinetto, Valmadre, Henriques, Vedaldi, & Torr, 2016; Hadsell, Chopra, & LeCun, 2006; Hoffer & Ailon, 2015; Koch, Zemel, & Salakhutdinov, 2015; Zagoruyko & Komodakis, 2015) including HAR. Yucer and Akgul (2018) proposed a Siamese LSTM for 3D HAR similarity metric between two 3D joint sequences. Ryoo, Kim, and Yang (2018) proposed multi-Siamese two-stream CNN for low-resolution activity recognition. But this matrix learning method needs similar activities as input and its generalization is not favorable when applied on the same activity with quite different contents.

Our proposed MSCNN is not a metric learning method and basically it is a kind of two-stream method. The main difference between metric learning and two-stream is that two-stream networks perform different functions in each stream, while Siamese networks typically perform similar functions in each stream. Conventional SNN is leveraged for metric learning with same weight parameters for the two branches such that obtaining the difference at the feature level. The proposed MSCNN combines SNN and two-stream together, which utilizes partially shared weights so as to perform different functions as well as saving a great number of memories. Leveraging SNN to improve the performance in HAR appeared in Karpathy et al. (2014) at the first time, they highlighted a multi-resolution CNN structure which has two branches of CNN, where one branch is for the resized entire image and the other branch is for center cropped image. The inputs of the two branches of CNN were low-resolution so as to reduce the computation expense while still retaining the performance. However, this paper did not explain why this kind of architecture works, while we explain the reasons through the mathematical model in Section 3.2.2 and experiments in Section 4.3.6. In addition, the method built in the assumption that the motion information is on the center of videos, when the condition changes, the method will fail to live up to the expectations. Singh, Marks, Jones, Tuzel, and Shao (2016) proposed a multi-stream bidirectional RNN for fine-grained action detection. The method took the RGB images and the optical flow images as two streams networks with bidirectional RNN. Each stream was divided into two branches – one branch for the entire scene and the other branch for human-centered region of the original image. Although the cropped image may contain some motion information, it might not be specifically related to the action. Moreover, the storage and computation of these networks were unnecessarily large. To further improve the method of Singh et al. (2016), Tu et al. (2018) proposed a human-related multi-stream CNN architecture with six CNN branches to encode appearance, motion, and the captured tubes of the human-related regions for HAR. In this method, each RGB stream and optical flow stream has three branches - one for the entire scene, one for the human-centered image, and the last one for motion saliency image. However, the computation of this method was enormous. When an activity has multiple motion saliency regions, it will miss important information. In addition, this method ignored information about human-interactions, which is critical for identifying activities, such as when a person is playing tennis, the bracket will be

missed. However, the proposed MSCNN does not require to extract human-centered image and saliency motion tubes, because the critical information of those has been included in the branch of motion in MSCNN.

3. Motion-patch-based Siamese CNN model

Recently, the attention mechanism has been proposed to enhance motion information at the network structure level for HAR, but there is no literature about motion enhancement at the data level. We believe that CNN can also benefit from data-level motion information enhancement. This is the main idea behind the proposed Siamese architecture for MSCNN, where we simultaneously feed the HAR networks with clips from conventional data augmentation and motion patch.

The proposed framework of MSCNN is shown in Fig. 1, which consists mainly of two CNNs with partially-shared weights. The upper branch named as context network has an input preprocessed by conventional data augmentation methods (*e.g.* corner cropping, random cropping) which is used for analyzing general information of the input video similar to the conventional video-based HAR approach. The lower branch named as motion network is to analyze salient motion sub-region of the input video, which is used to make sure that the salient motion information of the input video is not missed in the HAR. In addition, the salient motion region is extracted by an optical flow map based region of interest (ROI) selection method, which is named as Motion Patch Cropping (MPC) and its details are firstly described in this section.

3.1. Motion patch cropping

The proposed MPC method is inspired by the optical flow map-based motion estimation (Brox, Bruhn, Papenberg, & Weickert, 2004; Brox & Malik, 2010; Xu, Jia, & Matsushita, 2011). This map can be easily leveraged to highlight motion information within a video as well as providing good guidance to identify salient motion sub-region. In addition, optical flow maps can provide pixel-wise motion information accuracy (Brox et al., 2004). These are the main reasons for the proposed MPC method to adopt optical flow map.

The workflow of the proposed MPC method is illustrated as Fig. 2, which shows how to construct a motion patch cropped video clip as an input to the 3D-CNN based on the optical flow map. The input of this process is a video clip with the entire scene, typically it is a collection of consecutive video frames of size $T \times W \times H \times C$, where T represents the number of frames (length) of the input video clip, W and H represents width and height of the frames of the input clip which is in spatial domain, C represents the channel of the frames. For the convenience of the description of MPC, we set input size as $16 \times 320 \times 240 \times 3$ and the output size of the motion patch as $16 \times 112 \times 112 \times 3$ in Fig. 2, noting that the size of the input and output can be regularized according to application scenarios. The proposed MPC method can be divided into three parts: optical flow extraction, motion center computation, and motion patch cropping.

3.1.1. Optical flow extraction

Optical flow is intended to find the correspondence between two video frames by temporal variation of pixels in the sequence of frames. The optical flow extraction method is based on the intensity consistency assumption of one pixel in two consecutive image frames (Horn & Schunck, 1981), which is formulated as follows:

$$P(x, y, t) = P(x + \Delta x, y + \Delta y, t + \Delta t) \tag{1}$$

where $P(x, y, t)$ denotes the pixel intensity value at the position (x, y) of the frame at temporal location t. $(\Delta x, \Delta y)$ is the spatial displacement of the image at the corresponding axis between frames t and $t + \Delta t$. Expand the right hand side of Eq. (1) by Taylor series, we get

$$P(x, y, t) = P(x, y, t) + \Delta x \frac{\partial P}{\partial x} + \Delta y \frac{\partial P}{\partial y} + \Delta t \frac{\partial P}{\partial t} + \epsilon \tag{2}$$

where ϵ is the second and higher order terms in Δx, Δy, Δt. Assuming Δt is infinitesimal and dividing Eq. (2) by Δt, we have

$$\frac{\partial P}{\partial x} u + \frac{\partial P}{\partial y} v + \frac{\partial P}{\partial t} = 0 \tag{3}$$

where $u = \partial x / \partial t$ and $v = \partial y / \partial t$ constitute the optical flow vector for the pixel $P(x, y, t)$. The time between two neighboring frames is commonly fixed in a video, if the displacement of pixels become large, u and v will be large, so the optical flow u and v represent the strength of the motion of the object. If we ignore the environmental disturbances, such as the camera shake problems, the larger the range of motion, the greater the optical flow value. Therefore, we choose dense optical maps to obtain sub-region input clips composed of most of the motion information, while considering other trivial motion.

3.1.2. Motion center computation

After obtaining the dense optical flow maps for each input frame, we calculate the Center of Mass (CoM) of these optical flow maps to obtain the motion center of the input video clip for motion patch cropping. Based on the magnitude of the optical flow maps as grayscale images, we compute the CoMs of all these grayscale maps $O_i, i = 1, \ldots, T-1$, and use the average of these CoMs as the motion center of the input video clip for temporal smoothing. The motion center computation can be represented as Eq. (4) (van Assen, Egmont-Petersen, & Reiber, 2002) and Eq. (5).

$$c_i(x, y) = \left(\frac{\sum_{x,y \in \Omega} x \cdot v(x, y)}{\sum_{x,y \in \Omega} v(x, y)}, \frac{\sum_{x,y \in \Omega} y \cdot v(x, y)}{\sum_{x,y \in \Omega} v(x, y)} \right). \tag{4}$$

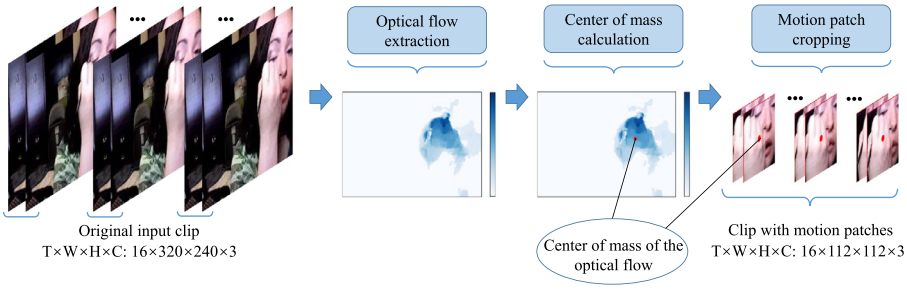

Fig. 2. The workflow of motion patch cropping: obtain the input of this method which is a clip made of a series of consecutive frames of a video (clip length is 16 in figure); calculate the dense map of optical flow of non-overlap every two consecutive frames of the obtained clip; calculate the CoM of the optical flow images and average all the CoM; spatially crop a fixed-size patch based on the CoM for each frame and obtain the motion patch cropped clip.

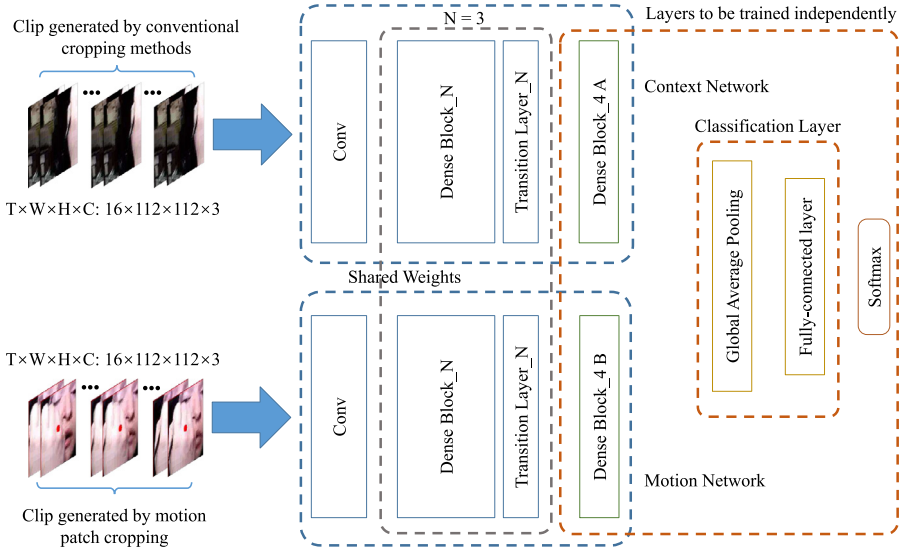

Fig. 3. Network structure of MSCNN (3D-DenseNet as the backbone): MSCNN consists of two partial-weight-shared branches of CNN – context network and motion network. In the proposed CNN, lower layers are weight shared while higher layers are independent. Taking DenseNet as the backbone for example in the figure, the first three blocks of DenseNet are weight-shared and the fourth block is trained independently.

$$c_m(x, y) = \frac{\sum_{i=1}^{T-1} c_i(x, y)}{T - 1}. \tag{5}$$

where Ω represents the pixels of O_i, x, y means the coordinate of pixels in Ω, $v(x, y)$ means the magnitude value of pixel (x, y) of optical flow maps. Since the dense optical flow maps represent the magnitude of the motion, the motion center tends to locate at the center of the salient motion region. It is worth noting that it is sufficient to calculate the middle pair of original image stack, but considering the influence of untrimmed video and the problems caused by video compression, the average of all calculated centers is more stable.

3.1.3. Cropping

After obtaining the motion center c_m, we spatially crop a fixed-size region (e.g. 112 × 112) from the original input clip using c_m as the region center as shown in Fig. 2. This cropped clip is the input of the motion network of the proposed MSCNN as shown in Fig. 1.

3.2. Motion-patch-based Siamese Convolutional Neural Networks

In HAR, motion information can be divided into two types – spatial and temporal information. Conventional CNN-based HAR approaches tend to learn two kinds of information through the network itself, but the degree to which the network understands the information is difficult to know and control. In order to solve the problem, we propose to provide a specially selected input video region, which is composed of important motion content, as an additional CNN path, focusing on motion information analysis.

3.2.1. Architecture

The proposed architecture of MSCNN is illustrated in Fig. 1, which includes two paths for processing the video input from the entire scene and the motion patch. The input of context network is obtained by conventional data augmentation methods and is resized to a fixed dimension of 112 × 112 × 3 with T frames (T is set 16 in Figs. 1 and 3). The input of motion network is the motion patch clip generated by the method in Section 3.1.

It is well known that lower layers of neural networks tend to extract low-level features such as edges and corners while higher layers provide high-level features with semantic information (Zeiler & Fergus, 2014). Therefore, the proposed MSCNN is designed to have the shared weights of the lower layers while the weight of higher layers will be trained independently. This design allows the two CNNs to extract their specific features separately. For the convenience of description, we take 3D-DenseNet (Huang, Liu, Van Der Maaten, & Weinberger, 2017) as the backbone and the network structure of MSCNN is shown as Fig. 3. It is worth noting that the backbone of MSCNN is flexible to choose (*e.g.* C3D, 3D-ResNext), which depends on the degree of the network fitting to the task. As shown in Fig. 3, the first three Dense Blocks and the transition layers of the two CNNs are set to share weights. Starting with the fourth Dense Block, the two Dense Blocks have their own weights during the training process. In addition, we link the features extracted by the two CNNs and apply them to a Global Average Pooling layer, followed by a fully-connected layer for Softmax classification. Basically, the scores after the classification layer are normalized by the Softmax function to obtain the prediction likelihood of each activity class. In this way, more weights are played back on the motion information without ignoring other information (*e.g.* environmental information, human-object interaction) of the input video. Furthermore, since the proposed MSCNN reads motion information for each iteration, it has the ability to mitigate the adverse effects of bad samples generated by random cropping-based data augmentation methods.

It is worth noting that if we only use a single-branch network with motion patch cropped video clip, the network tends to over-fit due to the insufficient video information and the loss of activity-related information such as environmental information, which affects the performance. In addition, since dense optical flow patterns are used for motion patch positioning, high accuracy of optical flow is not required.

3.2.2. Model formulation

In this sub-section, mathematical model of the proposed MSCNN is formulated. Let $X = \{X_i\}, i \in [1, N]$ denotes the training dataset, where N is the number of the videos in the training set and $X_i = \{x_{i1}, x_{i2}, \ldots, x_{iG}\}$ is ith video with G non-overlapping clips. x_{ij}^c denotes the clip generated by the conventional cropping methods and x_{ij}^m denote the clip generated by the MPC method from j_{th} clip in i_{th} video of the training set. $\mathcal{F}(x_{ij}; \mathbf{W})$ is the function

of the network on the clip x_{ij} with the entire parameters $\mathbf{W}$ and outcomes the scores $s_{ij} = \left\{s_{ij}^1, s_{ij}^2, \ldots, s_{ij}^C\right\}$, where C is the number of classes and s_{ij}^c is the score of cth class. In order to predict the likelihood, we adopt a normalization method Softmax function $\mathcal{S}$, which is computed as

$$\bar{s}_{ij}^c = \frac{e^{s_{ij}^c}}{\sum_{k=1}^{C} e^{s_{ij}^k}} \tag{6}$$

where $\bar{s}_{ij}^c$ is the normalized score of cth class. Formulating the loss function of the network with a regularized cross-entropy loss, we have

$$\mathcal{L}(\mathbf{y}, x, \mathbf{W}) = -\sum_{k=1}^{C} y_k \log \mathcal{S}_k(\mathcal{F}(x; \mathbf{W})) + \frac{1}{2}||\mathbf{W}|| \tag{7}$$

where $\mathbf{y} = (y_1, \ldots, y_C)^T$ is the one-hot vector of the ground truth of the input x and $\mathcal{S}_k = \bar{s}_{ij}^k$.

In the proposed MSCNN, $\mathcal{F}(x; \mathbf{W})$ can be expanded as

$$\mathcal{F}(x; \mathbf{W}) = \mathcal{C}(\mathcal{G}(\mathcal{F}_c(\mathcal{F}_s(x^c, \mathbf{W}^s), \mathbf{W}^c), \mathcal{F}_m(\mathcal{F}_s(x^m, \mathbf{W}^s), \mathbf{W}^m))) \tag{8}$$

where $\mathcal{F}_s$ is the function of the weight-shared network, which is the first three dense blocks in Fig. 3. $\mathcal{F}_c$ is the function of the separated network in the context network, which is the upper fourth Dense Block in Fig. 3. $\mathcal{F}_m$ is the function of the separated network in the motion network, which is the bottom fourth Dense Block. $\mathcal{G}$ is feature combination function (*e.g.* sum, concatenation, max) $\mathcal{C}$ is the classification function. $\mathbf{W}^s, \mathbf{W}^c, \mathbf{W}^m$ represent the parameters of the weight-shared network, the separated network in the context network and the separated network in the motion network, respectively. $\mathbf{W} = \left\{\mathbf{W}^s, \mathbf{W}^c, \mathbf{W}^m\right\}$. To simplify the explanation, we denote $F^\star = \{F_m^\star; F_o^\star\}, \star = \{s, c, m\}$ as the feature maps of $\mathcal{F}_s, \mathcal{F}_c, \mathcal{F}_m$, respectively. $F_m^\star$ denotes the feature maps activated by the information in motion patch. $F_o^\star$ denotes the feature maps activated by the information outside the motion patch. We assume no information outside the motion patch in the motion network, $F_o^s = \mathbf{0}$ for the feature maps in the motion network. Substituting the symbol of these feature maps into Eq. (8), we get

$$\mathcal{F}(x; \mathbf{W}) = \mathcal{C}(\mathcal{G}(\mathcal{F}_c(\{F_m^s; F_o^s\}), \mathbf{W}^c), \mathcal{F}_m(\{F_m^s\}), \mathbf{W}^m))) \tag{9}$$

On the other hand, a single branch neural network without motion network can be formulated as:

$$\mathcal{F}_S(x; \mathbf{W}) = \mathcal{C}(\mathcal{F}_c(\{F_m^s; F_o^s\}), \mathbf{W}^c) \tag{10}$$

Comparing Eq. (9) to Eq. (10), the proposed architecture obtains more motion features (information) of the motion patch than the single branch neural network. In another word, the proposed architecture adds weight to the motion information by feeding the motion patch.

4. Experiment

The experiments were conducted on two widely used benchmarks for human activity recognition: UCF-101 and HMDB-51. In this section, we will first introduce the experimental setup and then introduce the ablation studies of the proposed MSCNN network. Finally, we will compare our method with the state-of-the-art video-based HAR methods and demonstrate that the proposed method can achieve comparable results in RGB-only-trained methods.

4.1. Datasets

4.1.1. UCF-101

UCF-101 [dataset] (Soomro et al., 2012) is an action recognition data set containing 13,320 videos from 101 action categories with natural disturbance and avoiding non-motion frames. This dataset is typically used as a benchmark dataset for trimmed video performance evaluation.

4.1.2. HMDB-51

HMDB-51 [dataset] (Kuehne et al., 2013) includes 7000 activity videos distributed across 51 action categories with natural disturbances. This dataset is more challenging than UCF-101 because it contains untrimmed videos which may contain motion-unrelated frames, fierce camera jittering, and more. This dataset is typically used as an untrimmed dataset for testing evaluating HAR algorithms.

4.2. Experimental settings

4.2.1. Training

Since transfer learning has been proven to be highly efficient (Carreira & Zisserman, 2017), if the head (first few layers) of the model is not specified training from scratch in this paper, the model was fine-tuned based on a pre-trained model by Kinetics (Kay et al., 2017). When training the model from scratch, the learning rate was 0.1 and was divided by 10 when the validation loss plateau. When the training model was fine-tuned, it started at a learning rate of 0.001, and the learning rate was divided by 10 when the validation loss plateau. We set the weight decay to 1e−5 and set stochastic gradient descent (SGD) momentum to 0.9. In addition, batch normalization is applied to all convolutional layers. The input settings are listed as follows:

1. Context CNN Input Data Preprocessing: we refer to data generation method of the 3D-resnext in Hara, Kataoka, and Satoh (2018), which is the scaled corner cropping method. We uniformly selected a temporal position with the size of $112 \times 112 \times 3$ in a video and took the following consecutive N frames ($N = 16$ if not specified as 64f), then we stacked them into clips, which were the initially generated clips. Thus, we got the initially generated clip with a size of $N \times 112 \times 112 \times 3$. We also performed the input normalization for each generated clip refer to the mean values of ActivityNet. The clips were horizontally flipped with a probability of 0.5.
2. Motion CNN Input Data Preprocessing: the size of the motion patch, looping method, the normalization methods were same as the Context CNN Input Data Preprocessing, however, it was based on the proposed motion optical flow map based motion patch cropping method.

4.2.2. Inference

For fair comparisons with Hara et al. (2018), the input clip of the context network of MSCNN consists of the corresponding center cropped images from the images with smaller video side resized to 112 pixels. The input clip of the motion network consists of motion patch cropped images that are same as the training method but without flipping. Also, we adopt the sliding window manner for the classification as it in Hara et al. (2018). The sliding window manner was that we averaged the scores of each non-overlapping clip in one video for the classification.

4.2.3. Other implemental details

1. Due to the disturbance of untrimmed video, the magnitude of optical flow maps may include some noises that affects the accuracy of the motion center. We set a threshold when calculating the center of motion by the threshold selection method OTSU (Otsu, 1979). In order to spatially smooth the optical flow, only magnitude values above the pre-defined threshold are used to calculate the center of mass (CoM).
2. Training and inference were done by Pytorch on a platform with one GTX1080Ti GPU and one Intel Xeon 2104 processer.

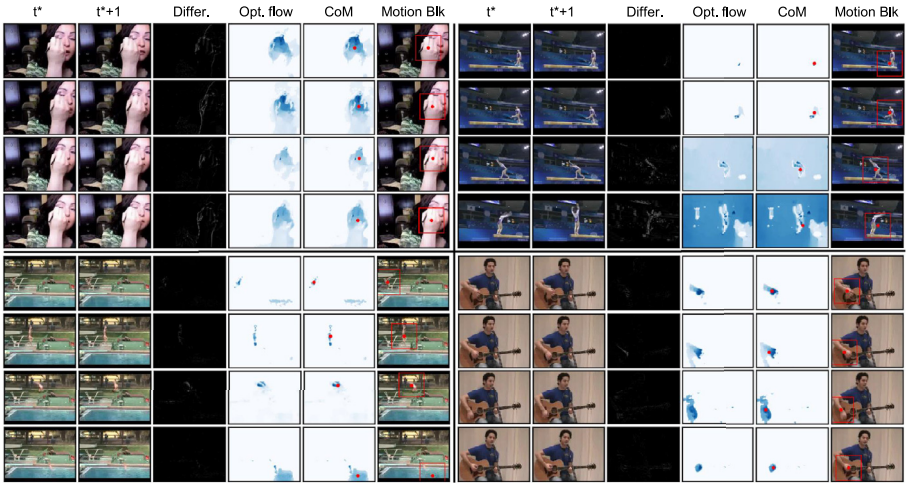

Fig. 4. Results of motion patch cropping: from left to right, the contents in each section are the frame at time t^*, the frame at time t^*+1, the difference between the two frames, the magnitude of the dense optical flow map between the two frames, the CoM (the red points) of the optical flow, the motion patch (contents in the red rectangle) of t^*.

4.3. Experimental results

4.3.1. Motion patch cropping

The performance of the optical flow-based motion patch extraction method is shown in Fig. 4. The figure consists of four parts that belong to four videos categories of UCF-101 dataset. From left to right, the contents in Fig. 4 in each part are the frames at time t^* and t^*+1, the difference between the two frames, the magnitude of the dense optical flow map based on the two frames, the estimated CoM as a red dot and the identified motion patch as a red rectangle region of t^*. In the MPC, the optical flow value is truncated to $[-20, 20]$ for mitigating the effects of interference. And we normalized it into [0, 255] for JPG compression. When calculating the CoM, the pixel values under the threshold calculated by OTSU method will be regarded as noise and neglected. In Fig. 4, it is obvious that the proposed MPC method has the ability to crop the key motion region of the frame. In the upper right corner, the background of the light flow diagram in the last two series is blue while others are white because the camera was moving to track the player. Even if the shooting is unstable, the MPC method we proposed is accurate.

4.3.2. Comparison with different backbones

To test the effectiveness of our proposed networks, we compared the proposed method to Hara et al. (2018) under different backbones, as well as attention mechanism – non-local (Wang et al., 2018b). The results are shown in Table 1. It is not hard to conclude that our MSCNN outperformed the method in Hara et al. (2018) at each backbone. This is because our method can enhance the motion information and alleviate the influence of bad samples generated from conventional data augmentation methods. We inserted the attention mechanism non-local block into resnet-18 and trained the network from scratch. The dot product non-local method was utilized and the non-local block was inserted after the second block of resnet-18, which was the setting of the best results in Wang et al. (2018b). In Table 1, though the accuracy increased with the non-local method, the performance of the proposed method MSCNN was better. It is an interesting finding that the improvement in UCF-101 is more than that on HMDB-51 in both the non-local method and our proposed method. The reason is that HMDB-51 is an untrimmed video dataset which makes the two kinds of networks hard to find effective motion information.

Table 1
Comparison with different backbones.

Backbone	Hara et al. (2018)		MSCNN (proposed)	
	UCF-101	HMDB-51	UCF-101	HMDB-51
ResNet-18 (scratch)	42.4	17.1	**52.8**	**20.2**
ResNet-18-non-local (scratch)	45.8	17.7	–	–
ResNet-18	84.4	56.4	**88.3**	**57.5**
ResNet-50	89.3	61.0	**91.7**	**64.7**
ResNet-101	88.9	61.7	**91.3**	**64.4**
DenseNet-121	87.6	59.6	**91.5**	**60.8**
ResNext-101	90.7	63.8	**93.5**	**70.7**
ResNext-101 (64f)	94.5	70.2	**96.8**	**74.8**

Table 2
Comparison with different feature fusion methods.

Fusion methods	UCF-101 (split 1)	HMDB-51 (split 1)
Baseline (single branch)	86.0	60.1
Sum	89.7	60.9
Concatenate	**90.6**	**61.3**
Convolution	90.0	60.8
CBP (Gao et al., 2016)	86.3	60.2
Multiply	89.1	60.4

4.3.3. Fusion methods

We compared different feature fusion methods in our networks following Feichtenhofer et al. (2016b). We chose the methods of sum, concatenation, convolution and multiply for experiments. Furthermore, we testified another popular feature fusion method – compact bilinear pooling (CBP) (Gao, Beijbom, Zhang, & Darrell, 2016). In these experiments, the split 1 of UCF-101 and HMDB-51 were the datasets and 3D DenseNet-121 was chosen as the backbone. The results are shown in Table 2. Although each feature fusion method improved the accuracy, we could conclude that the concatenation method was the best feature fusion method for this architecture. The conclusion was different from it in Feichtenhofer et al. (2016b) where the convolution method was the best method. The reason was that the generated features from the two branches of MSCNN were of the same kind of features and very similar to each other. When we used the concatenation method, the architecture was capable of enhancing the overlapped features which were the critical motion information. For the architecture in Feichtenhofer et al. (2016b), the features belonged to

Table 3
Comparison with different concatenation location.

Location	UCF-101 (split 1)	HMDB-51 (split 1)
Baseline	86.7	60.1
DenseBlock-1	88.7	60.4
DenseBlock-2	88.9	60.7
DenseBlock-3	89.4	60.9
DenseBlock-4	**90.6**	**61.3**

Table 4
Comparison with different motion patch size.

Size	UCF-101 (split 1)	HMDB-51 (split 1)
Baseline	86.7	60.1
56 × 56	87.4	60.3
84 × 84	88.8	60.4
112 × 112	**90.6**	**61.3**
168 × 168	90.4	60.9
224 × 224	89.0	60.6

two different kinds – RGB frames and optical flow, so the feature fusion method such as the convolution method worked the best. The reason was the same for the CBP method.

4.3.4. Fusion location

In MSCNN, we concatenated the features before the classification layer based on the assumption that the higher layers would provide semantic features which could benefit final classification. To testify our assumption, the ablation study of different fusion locations was conducted based on the UCF-101 dataset and HMDB-51. The results can be seen in Table 3. It is not difficult to conclude that fusing the features after the last block was the best fusion location and with the fusion location climbing up, the performance decreased. This phenomenon showed that semantic features from higher layers are better for data-level motion information enhancement.

4.3.5. Motion area size

We believe that the motion branch in Section 3.2 has the ability to enhance the critical motion information so as to improve the performance. Different sizes of the motion region contain different amounts of information. In order to find a convincing size of motion patch, we conducted experiments cropping different sizes of the motion patch and then spatially resize the motion patch into 112 × 112 for the motion network. The results can be seen as Table 4. The different sizes are 56 × 56, 84 × 84, 112 × 112, 168 × 168, 224 × 224, respectively. It is obvious that when the cropping size is 112 × 112, MSCNN performed the best. The reason is that when the cropping size is too small, the motion network will get insufficient motion information to enhance, which affects the performance. On the contrary, when the cropping size is too large, MSCNN gets too much information to distinct which information should be emphasized. In addition, we found that the performances were improved at any size of the motion patch. It means that though the different amounts of emphasized motion lead to a different performance, the data-level motion information enhancement can always improve the performance.

Table 5
Comparison with different cropping method.

Cropping method	UCF-101 (split 1)	HMDB-51 (split 1)
Baseline (single branch)	86.7	60.1
Random-cropping	88.5	60.5
Corner-cropping	87.4	60.6
Center-cropping	88.9	60.9
Motion-patch-cropping	**90.6**	**61.3**

4.3.6. Cropping methods of motion branch

Our proposed method is based on the assumption that the motion network in MSCNN provides attention to the critical motion information and has the ability to enhance the information. But there may exist doubt that the increase of accuracy benefited from the ensemble of our proposed structure, however, not benefited from the motion information enhancement. To testify this assumption and remove the doubt, we compared our motion patch cropping method for the motion branch with conventional data augmentation methods random-cropping, corner-cropping, and center-cropping. The results are as Table 5.

It is obvious that although conventional data augmentation methods could increase the accuracy, the accuracies were not comparable to the accuracy of the motion patch cropping method. This is because the conventional cropping methods could provide attention on the motion information but the amount of attention was not comparable to that the motion patch cropping method could provide. When the corner cropping was adopted for the motion branch, the accuracy increased the least. The increase by the corner cropping method actually was benefit from the ensemble of networks which was much lower than the increase by the motion patch cropping method. Thus, this experiment evaporates the doubt that MSCNN is an ensemble and testified the effectiveness of data-level motion information enhancement.

4.4. Comparison to state-of-the-art methods

We compare the performance of the proposed method with state-of-the-art HAR methods in Table 6. As can be seen in Table 6, the proposed method achieves state-of-the-art performance on 3D-ResNext, *i.e.*, 96.8% and 74.8% on datasets UCF-101 and HMDB-51, respectively. In contrast, it can be clearly seen that without using the proposed method, the 3D-ResNext could only achieve 94.5% accuracy on UCF-101 and 70.2% on HMDB-51. On HMDB-51, we found that the proposed method improves the performance of 3D-ResNext by 4.6% and further gains 0.3% improvement compared

Table 6
Comparison to the state-of-the-art on UCF-101 and HMDB-51 (mean accuracy across 3 splits).

Streams	Method	Pretrain	UCF-101	HMDB-51
RGB	LRCN (Donahue et al., 2015)	None	82.9	–
	C3D (3 nets) (Tran et al., 2015)	Sports-1M	85.2	–
	Res3D (Tran, Ray, Shou, Chang, & Paluri, 2017)	Sports-1M	85.8	54.9
	P3D (Qiu et al., 2017)	ImNet+Spo	88.6	–
	MiCT-Net (Zhou et al., 2018)	None	88.9	63.8
	3D-ResNext (64f) (Hara et al., 2018)	Kinetics	94.5	70.2
	RGB-I3D (64f) (Carreira & Zisserman, 2017)	ImNet+Kin	95.4	74.5
	R(2+1)D (Tran et al., 2018)	Kinetics	96.8	74.5
	MSCNN$_{\text{ResNext101}}$ (64f)	Kinetics	**96.8**	**74.8**

to R(2+1)D. It is noted that the inputs of RGB-I3D are 224 × 224 which is four times larger than our method. If we simply use 3D-ResNext, the accuracy in HMDB-51 is 70.2 which is quite lower than RGB-I3D. When 3D-ResNext work with our method, we could outperform RGB-I3D on the two datasets. Thus, we can conclude that the proposed method has the ability to enhance the performance of HAR systems and achieved comparable results to the state-of-the-art methods.

It should be noted that the proposed method is a data-level motion information enhancement which can guide HAR system to pay more attention to critical motion information and alleviate the negative influence of bad samples generated by random cropping-based data augmentation. According to the results in Section 4.3.2, our method can be used with R(2+1)D and I3D as the backbone to further improve the performance since the data augmentation of these two methods are based on random cropping.

5. Conclusion and future work

In this paper, we proposed a Siamese neural network named as Motion-patch-based Siamese Convolutional Neural Network which is the first attempt to alleviate the influence of bad samples generated by conventional data augmentation methods and to enhance the motion information of videos through input data in HAR systems. In addition, we proposed a simple but effective way to select the motion patch containing the critical motion information for the Siamese neural network. Sufficient experiments have testified the effectiveness of our proposed method. This method outperforms a lot than the original backbones and achieved comparable results. The proposed method can be utilized to boost the performance of any HAR algorithm which is based on random cropping as the data augmentation method. In addition, this paper reveals a new research direction for deep learning-based HAR systems that it is effective to improve the performance by using the MSCNN network structure without much calculation expense.

The limitations of the proposed method are three folds. Although there are fast optical flow extraction methods such as learning-based optical flow extraction method (Ilg et al., 2017; Sun, Yang, Liu, & Kautz, 2018) which can obtain optical flow in real time, using optical flow to get the motion patch is not optimized since the feature of motion patches are included in the compressed videos and the extracted features by MSCNN. What is more, MSCNN concatenates the features of two branches of the network to generate final features for reasoning, which is a simple way to synthesis the features. The concatenation method for such kind of features deserves deep research.

Considering the limitations, the future work includes proposing an elegant method to extract motion patches in MSCNN such as motion vector which can be obtained through the compressed video without much computation expense or designing learning-based method for the extraction. In addition, finding an optimized way to concatenate the features from two branches of MSCNN is also an interesting direction since the features of motion patches are partially related to the features of the full scene. Furthermore, the weights of the two branches in MSCNN are equal in this paper. Thinking how to use learning-based method to search for good weights for the two branches can be interesting work.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Credit authorship contribution statement

Yujia Zhang: Conceptualization, Methodology, Software, Writing - original draft, Formal analysis. **Lai Man Po:** Validation, Supervision, Writing - review & editing. **Mengyang Liu:** Resources, Investigation. **Yasar Abbas Ur Rehman:** Formal analysis, Writing - review & editing. **Weifeng Ou:** Data curation, Writing - review & editing. **Yuzhi Zhao:** Visualization.

References

van Assen, H. C., Egmont-Petersen, M., & Reiber, J. H. (2002). Accurate object localization in gray level images using the center of gravity measure: accuracy versus precision. *IEEE Transactions on Image Processing, 11*(12), 1379–1384.

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. arXiv:1409.0473.

Baldi, P., & Chauvin, Y. (1993). Neural networks for fingerprint recognition. *Neural Computation, 5*(3), 402–418.

Bertinetto, L., Valmadre, J., Henriques, J. F., Vedaldi, A., & Torr, P. H. (2016). Fully-convolutional siamese networks for object tracking. In *Proceedings of the European conference on computer vision* (pp. 850–865). Springer.

Borji, A., & Itti, L. (2013). State-of-the-art in visual attention modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 35*(1), 185–207.

Bromley, J., Guyon, I., Lecun, Y., & Shah, R. (1993). Signature verification using a "siamese" time delay neural network. In *Proceedings of the international conference on neural information processing systems*.

Brox, T., Bruhn, A., Papenberg, N., & Weickert, J. (2004). High accuracy optical flow estimation based on a theory for warping. In *Proceedings of the European conference on computer vision* (pp. 25–36). Springer.

Brox, T., & Malik, J. (2010). Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 33*(3), 500–513.

Caba Heilbron, F., Escorcia, V., Ghanem, B., & Carlos Niebles, J. (2015). ActivityNet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 961–970).

Carreira, J., & Zisserman, A. (2017). Quo vadis, action recognition? a new model and the kinetics dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 6299–6308).

Crasto, N., Weinzaepfel, P., Alahari, K., & Schmid, C. (2019). Mars: Motion-augmented RGB stream for action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7882–7891).

Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., & Darrell, T. (2015). Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2625–2634).

Feichtenhofer, C., Pinz, A., & Wildes, R. (2016a). Spatiotemporal residual networks for video action recognition. In *Proceedings of the Advances in neural information processing systems* (pp. 3468–3476).

Feichtenhofer, C., Pinz, A., & Wildes, R. P. (2017). Spatiotemporal multiplier networks for video action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4768–4777).

Feichtenhofer, C., Pinz, A., & Zisserman, A. (2016b). Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1933–1941).

Firat, O., Cho, K., & Bengio, Y. (2016). Multi-way, multilingual neural machine translation with a shared attention mechanism. arXiv:1601.01073.

Gao, Y., Beijbom, O., Zhang, N., & Darrell, T. (2016). Compact bilinear pooling. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 317–326).

Girshick, R. (2015). Fast R-CNN. In *Proceedings of the IEEE international conference on computer vision* (pp. 1440–1448).

Gottfried, B., & Aghajan, H. K. (2009). *Behaviour monitoring and interpretation-BMI: Smart Environments*: 3. IOS Press.

Hadsell, R., Chopra, S., & LeCun, Y. (2006). Dimensionality reduction by learning an invariant mapping. In *Proceedings of the 2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06): 2* (pp. 1735–1742). IEEE.

Hara, K., Kataoka, H., & Satoh, Y. (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* (pp. 6546–6555).

Harel, J., Koch, C., & Perona, P. (2007). Graph-based visual saliency. In *Proceedings of the advances in neural information processing systems* (pp. 545–552).

He, D., Zhou, Z., Gan, C., Li, F., Liu, X., Li, Y., Wang, L., & Wen, S. (2018). StNet: local and global spatial-temporal modeling for action recognition. arXiv:1811.01549.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).

Hernández-García, R., Ramos-Cózar, J., Guil, N., García-Reyes, E., & Sahli, H. (2018). Improving bag-of-visual-words model using visual n-grams for human action classification. *Expert Systems with Applications, 92*, 182–191.

Hoffer, E., & Ailon, N. (2015). Deep metric learning using triplet network. In *Proceedings of the IEEE international workshop on similarity-based pattern recognition* (pp. 84–92). Springer.

Horn, B. K., & Schunck, B. G. (1981). Determining optical flow. *Artificial Intelligence, 17*(1–3), 185–203.

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4700–4708).

Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., & Brox, T. (2017). FlowNet 2.0: evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2462–2470).

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv:1502.03167.

Itti, L., Koch, C., & Niebur, E. (1998). A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis & Machine Intelligence, 20*(11), 1254–1259.

Ji, S., Xu, W., Yang, M., & Yu, K. (2013). 3D convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 35*(1), 221–231.

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., & Fei-Fei, L. (2014). Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1725–1732).

Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P. et al. (2017). The kinetics human action video dataset. arXiv:1705.06950.

Koch, C., & Ullman, S. (1987). Shifts in selective visual attention: towards the underlying neural circuitry. In *Matters of intelligence* (pp. 115–141). Springer.

Koch, G., Zemel, R., & Salakhutdinov, R. (2015). Siamese neural networks for one-shot image recognition. In *Proceedings of the ICML deep learning workshop: 2*.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In *Proceedings of the advances in neural information processing systems* (pp. 1097–1105).

Kuehne, H., Jhuang, H., Stiefelhagen, R., & Serre, T. (2013). HMDB51: a large video database for human motion recognition. In *High performance computing in science and engineering 12* (pp. 571–582). Springer.

Liu, J., Luo, J., & Shah, M. (2009). Recognizing realistic actions from videos "in the wild". In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1996–2003).

Mabrouk, A. B., & Zagrouba, E. (2018). Abnormal behavior recognition for intelligent video surveillance systems: A review. *Expert Systems with Applications, 91*, 480–491.

Meng, L., Zhao, B., Chang, B., Huang, G., Tung, F., & Sigal, L. (2018). Where and when to look? Spatio-temporal attention for action recognition in videos. arXiv:1810.04511.

Ni, B., Moulin, P., Yang, X., & Yan, S. (2015). Motion part regularization: Improving action recognition via trajectory selection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3698–3706).

Niebles, J. C., Chen, C.-W., & Fei-Fei, L. (2010). Modeling temporal structure of decomposable motion segments for activity classification. In *Proceedings of the European conference on computer vision* (pp. 392–405). Springer.

Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics, 9*(1), 62–66.

Qiu, Z., Yao, T., & Mei, T. (2017). Learning spatio-temporal representation with pseudo-3d residual networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 5533–5541).

Rahman, S. A., Song, I., Leung, M. K., Lee, I., & Lee, K. (2014). Fast action recognition using negative space features. *Expert Systems with Applications, 41*(2), 574–587.

Ranasinghe, S., Al Machot, F., & Mayr, H. C. (2016). A review on applications of activity recognition systems with regard to performance and evaluation. *International Journal of Distributed Sensor Networks, 12*(8). 1550147716665520

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: towards real-time object detection with region proposal networks. In *Proceedings of the advances in neural information processing systems* (pp. 91–99).

Ryoo, M. S., Kim, K., & Yang, H. J. (2018). Extreme low resolution activity recognition with multi-siamese embedding learning. In *Proceedings of the thirty-second AAAI conference on artificial intelligence*.

Sharma, S., Kiros, R., & Salakhutdinov, R. (2015). Action recognition using visual attention. arXiv:1511.04119.

Simonyan, K., & Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. In *Proceedings of the advances in neural information processing systems* (pp. 568–576).

Singh, B., Marks, T. K., Jones, M., Tuzel, O., & Shao, M. (2016). A multi-stream bi-directional recurrent neural network for fine-grained action detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1961–1970).

Song, S., Lan, C., Xing, J., Zeng, W., & Liu, J. (2018). Spatio-temporal attention-based LSTM networks for 3D action recognition and detection. *IEEE Transactions on Image Processing, 27*(7), 3459–3471.

Soomro, K., Zamir, A. R., & Shah, M. (2012). Ucf101: A dataset of 101 human actions classes from videos in the wild. arXiv:1212.0402.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research, 15*(1), 1929–1958.

Sun, D., Yang, X., Liu, M.-Y., & Kautz, J. (2018). PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 8934–8943).

Tran, D., Bourdev, L., Fergus, R., Torresani, L., & Paluri, M. (2015). Learning spatiotemporal features with 3D convolutional networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 4489–4497).

Tran, D., Ray, J., Shou, Z., Chang, S.-F., & Paluri, M. (2017). ConvNet architecture search for spatiotemporal feature learning. arXiv:1708.05038.

Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., & Paluri, M. (2018). A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 6450–6459).

Tsotsos, J. K. (1990). Analyzing vision at the complexity level. *Behavioral and Brain Sciences, 13*(3), 423–445.

Tsotsos, J. K., Culhane, S. M., Wai, W. Y. K., Lai, Y., Davis, N., & Nuflo, F. (1995). Modeling visual attention via selective tuning. *Artificial Intelligence, 78*(1–2), 507–545.

Tu, Z., Xie, W., Qin, Q., Poppe, R., Veltkamp, R. C., Li, B., & Yuan, J. (2018). Multi-stream CNN: Learning representations based on human-related regions for action recognition. *Pattern Recognition, 79*, 32–43.

Turaga, P., Chellappa, R., Subrahmanian, V. S., & Udrea, O. (2008). Machine recognition of human activities: A survey. *IEEE Transactions on Circuits and Systems for Video technology, 18*(11), 1473.

Varol, G., & Salah, A. A. (2015). Efficient large-scale action recognition in videos using extreme learning machines. *Expert Systems with Applications, 42*(21), 8274–8282.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., … Polosukhin, I. (2017). Attention is all you need. In *Proceedings of the advances in neural information processing systems* (pp. 5998–6008).

Wang, H., Ullah, M. M., Klaser, A., Laptev, I., & Schmid, C. (2009). Evaluation of local spatio-temporal features for action recognition. In *Proceedings of the 2009 British machine vision conference, BMVC*. BMVA Press. 124–1

Wang, J., Cherian, A., Porikli, F., & Gould, S. (2018a). Video representation learning using discriminative pooling. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1149–1158).

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

Wang, X., Girshick, R., Gupta, A., & He, K. (2018b). Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7794–7803).

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., … Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. In *Proceedings of the International conference on machine learning* (pp. 2048–2057).

Xu, L., Jia, J., & Matsushita, Y. (2011). Motion detail preserving optical flow estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 34*(9), 1744–1757.

Yang, Z., He, X., Gao, J., Deng, L., & Smola, A. (2016). Stacked attention networks for image question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 21–29).

Yi, Y., Zheng, Z., & Lin, M. (2017). Realistic action recognition with salient foreground trajectories. *Expert Systems with Applications, 75*, 44–55.

Yucer, S., & Akgul, Y. S. (2018). 3D human action recognition with siamese-LSTM based deep metric learning. arXiv:1807.02131.

Zagoruyko, S., & Komodakis, N. (2015). Learning to compare image patches via convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4353–4361).

Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In *Proceedings of the European conference on computer vision* (pp. 818–833). Springer.

Zhou, Y., Sun, X., Zha, Z.-J., & Zeng, W. (2018). MiCT: Mixed 3D/2D convolutional tube for human action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 449–458).

Zhu, J., Zhu, Z., & Zou, W. (2018). End-to-end video-level representation learning for action recognition. In *Proceedings of the 24th international conference on pattern recognition (ICPR)* (pp. 645–650). IEEE.