Human Action Recognition Using Multi-Stream Fusion and Hybrid Deep Neural Networks

Saurabh Chopra
Department of Computer Science
Royal Holloway, University of London
Surrey, United Kingdom
sc@saurabhchopra.co.uk

Li Zhang
Department of Computer Science
Royal Holloway, University of London
Surrey, United Kingdom
Li.Zhang@rhul.ac.uk

Ming Jiang
Department of Computer Science
Faculty of Technology
Sunderland, United Kingdom
ming.jiang@sunderland.ac.uk

***Abstract*—Action Recognition in videos is a topic of interest in the area of computer vision, due to potential applications such as multimedia indexing and surveillance in public areas. In this research, we first propose spatial and temporal Convolutional Neural Network (CNNs), based on transfer learning using ResNet101, GoogleNet and VGG16, for undertaking human action recognition. Besides that, hybrid networks such as CNN-Recurrent Neural Network (RNN) models are also exploited as encoder-decoder architectures for video action classification. In particular, different types of RNNs such as Long Short-Term Memory (LSTM), Bidirectional-LSTM (BiLSTM), Gated Recurrent Unit (GRU), and Bidirectional-GRU (BiGRU), are exploited as the decoders for action recognition. To further enhance performance, diverse aggregation networks of CNN and CNN-RNN models are implemented. Specifically, an Average Fusion method is used to integrate spatial and temporal CNNs trained on images, as well as CNN-RNN trained on videos, where the final classification is formed by combining Softmax scores of these models via a late fusion. A total of 22 models (1 motion CNN, 3 spatial CNNs, 12 CNN-RNNs and 6 fusion networks) are implemented which are evaluated using UCF11, UCF50, and UCF101 datasets for performance comparison. The empirical results indicate the significant efficiency of Average Fusion of multiple Spatial-CNNs with one Motion-CNN, and ResNet101-BiGRU, among all the networks for undertaking realistic video action recognition.**

***Index Terms*—Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), Bidirectional-LSTM (BiLSTM), Gated Recurrent Unit (GRU), CNN-LSTM, CNN-BiLSTM, CNN-GRU, CNN-BiGRU, and Average Fusion**

I. INTRODUCTION

Human action recognition experiences an exciting era in computer vision due to the recent advancement of deep learning techniques. The purpose of action recognition is to automatically interpret actions performed in videos. A video can be regarded as a sequence of frames, usually 30 frames per second. Therefore, a potential approach for action recognition is to apply a classifier on individual frames of an video, with an aggregation method used to combine results from all frames to form a final prediction. Motivated by such observations, in this research, we propose spatial and temporal Convolutional Neural Networks (CNNs), hybrid CNN-Recurrent Neural Network (RNN) models, as well as their fusion networks for video action recognition.

Firstly, owing to the powerful classification performance of pre-trained CNNs such as ResNet101, GoogleNet and VGG16, transfer learning based on these networks is employed for action recognition. We construct two types of CNN classifiers with RGB image frames and optical flow as inputs, respectively. Specifically three spatial CNNs using ResNet101, GoogleNet and VGG16 as backbones and one motion CNN based on ResNet101 are developed using transfer learning.

Secondly, hybrid encoder-decoder architectures such as CNN-RNN models are also exploited to extract spatial-temporal features directly from video inputs to inform action recognition. In particular, different CNNs such as ResNet101, GoogleNet and VGG16 are employed as the encoders and variants of RNN such as Long Short-Term Memory (LSTM), Bidirectional-LSTM (BiLSTM), Gated Recurrent Units (GRU), and Bidirectional-GRU (BiGRU), are utilized as the decoders. A set of 12 resultant CNN-RNNs is adopted for classifying different actions from video inputs.

After generating transfer learning networks (1 motion CNN and 3 spatial CNNs) and 12 hybrid models, 6 additional aggregation networks based on Average Fusion are proposed to integrate spatial and temporal CNNs and CNN-RNNs to further enhance performance.

Several UCF action datasets [1] such as UCF11, UCF50 and UCF101 are employed for evaluating the aforementioned methods. The novel aspects of this research are as follows.

- We propose diverse Spatial and Motion-CNNs, and encoder-decoder hybrid architectures for undertaking human action recognition from realistic videos. Transfer learning based on ResNet101, GoogleNet and VGG16 is used to develop one temporal and three spatial networks using video frames. In addition, encoder-decoder architectures are also utilized to conduct action recognition using video inputs directly, where several CNNs (ResNet101, GoogleNet, and VGG16) and RNN variants (LSTM, BiLSTM, GRU, and BiGRU) are employed as encoders and decoders, respectively. In short, 1 motion CNN, 3 spatial CNNs, and 12 hybrid CNN-RNN models are developed as base classifiers for action recognition.
- Distinctive fusion strategies are exploited. Besides the original two-stream spatial and temporal fusion, the integration of several spatial CNNs with a motion CNN, as

well as the fusion of several CNN-RNN networks with a motion CNN, is proposed. A total of six fusion networks are implemented in this research.
- Evaluated using several well-known video action datasets (i.e. UCF11, UCF25 (half of UCF50), UCF50 and UCF101), among all the 22 proposed base and fusion models, the aggregation networks of multiple Spatial-CNNs and one Motion-CNN, as well as some of hybrid CNN-RNN models, achieve superior performance and outperform existing methods, for realistic video action recognition.

II. RELATED WORK

Action recognition from realistic videos is a challenging problem because of complexity of the actions involved and cluttered background. Realistic action videos extracted from YouTube, movies and TV shows are employed to test model efficiency. Many benchmark algorithms have been developed. For example, state-of-the-art two-stream and spatial-temporal algorithms have been proposed with highest accuracy rates for realistic video action recognition, including MARS+RGB+Flow [2], two-stream I3D [3], two-stream LGD-3D [4], two-stream R[2+1]D [5], multi-stream I3D [6] and R[2+1]D-RGB [5]. We discuss several recent state-of-the-art developments below.

Thatipelli et al. [7] extracted higher order spatial and temporal cues using local patch-level and global frame-level feature learning components, respectively, which led to high classification performance on challenging SSv2 benchmark dataset. Radevsk et al. [8] proposed a multi-head attention method over spatio-temporal layouts, which was proven to be effective for spatial reasoning. The work showed improvement in performance when fusing appearance-based methods with layout-based models. Luo et al. [9] adopted 3D-CNN in conjunction with a decomposition method for action recognition. The decomposition method was able to break down feature channels into spatial and temporal characteristics and measure the distinctive contributions of these respective elements. But their resultant method also increased computational cost significantly.

The study of Materzynska et al. [10] showed that interactions between spatial and temporal networks can help better understand relationships between the subject and object in an video. Their experiments also hinted that when using the I3D model with ResNet50 as the base model showed a gain in model performance, in comparison with other baseline methods. In addition, when the I3D model was combined/fused with baseline models, their work obtained an even higher top-1 accuracy rate.

Du et al. [11] employed a spatial-temporal attention mechanism for the extraction of key global features for RNN-based action prediction. Their experiments also indicated the effectiveness of spatial and temporal streams for boosting network discriminative capabilities in tackling action classification. The model showed competitive performance for UCF101, HMDB51 and JHMDB datasets. Plizzari et al. [12] embedded spatial and temporal attention schemes into a two-stream architecture with the attempt to capture intra-frame interactions and inter-frame correlations, respectively, for action classification.

Yu et al. [13] integrated attention mechanisms with 3D convolution to extract more enhanced motion and spatial details. In addition, their work adopted a BiLSTM-based attention module to extract temporal patterns from video cubes to tackle action recognition using long videos. Yang et al. [14] employed temporal and spatial attention mechanisms to improve action recognition where the temporal attention component extracted the most significant sequential patterns from lengthy videos and the spatial attention function identified the most discriminative motion details from optical flow. Their model showed enhanced performance than those of existing methods for UCF101 and HMDB51 datasets.

This research is particularly motivated by the following existing state-of-the-art methods. Simonyan and Zisserman [15] investigated two-stream architectures comprising spatial and motion CNNs, with linear and Support Vector Machine (SVM)-based fusion strategies for action recognition. Kinghorn et al. [16] investigated encoder-decoder modeling for video captioning, while Inception-style Temporal-ConvNet was also explored by [17] to extract refined temporal details. There are also other 3DCNN, hybrid and transformer based methods proposed in recent years for video action recognition.

III. THE PROPOSED METHODOLOGIES

In this research, spatial and temporal CNNs, hybrid CNN-RNN models, as well as aggregation networks are exploited for video action recognition. We discuss each of the proposed models below.

Firstly, we explore transfer learning with CNNs for action recognition. As mentioned earlier, several ImageNet pre-trained CNN models are fine-tuned using video frames and optical flow respectively. Specifically, one motion and three spatial CNN models are implemented using transfer learning, i.e. ResNet101 (Motion), ResNet101 (Spatial), GoogleNet (Spatial) and VGG16 (Spatial). The Motion ResNet101 is re-trained using optical flow extracted from the video inputs, while the three spatial CNNs are fine-tuned with the video frames.

Secondly, encoder-decoder networks are constructed for action recognition. Precisely we employ three pre-trained models, i.e. ResNet101, GoogleNet and VGG16, as the encoders, and four RNN variants, i.e. LSTM, BiLSTM, GRU and BiGRU, as the decoders. Such hybrid CNN-RNN models are able to extract sufficient discriminative spatial-temporal cues from video inputs directly for video classification. Diverse permutations of encoder and decoder networks are conducted which result in the generation of 12 hybrid networks for action recognition.

Beside the proposal of spatial and temporal CNNs and hybrid networks, we perform late fusion of these methods, which includes,

– Average Fusion of Spatial-CNN & Motion-CNN (two streams).

– Average Fusion of all CNN Models & Motion-CNN (Multiple streams).

– Average Fusion of all CNN-LSTM Models & Motion-CNN (Multiple streams).

– Average Fusion of all CNN-BiLSTM Models & Motion-CNN (Multiple streams).

– Average Fusion of all CNN-GRU Models & Motion-CNN (Multiple streams).

– Average Fusion of all CNN- BiGRU Models & Motion-CNN (Multiple streams).

In these fusion networks, we aggregate three spatial CNNs with one motion CNN. We also integrate multiple hybrid CNN-RNNs with the motion CNN. Such spatial and temporal methods are able to provide complementary information for aggregated network construction.

A set of 22 models are implemented including 4 CNNs (3 spatial CNNs, 1 motion CNN), 12 CNN-RNNs, and 6 Average Fusion models. All these 22 Models are trained and evaluated on four different datasets, i.e. UCF-11, UCF-25 (half of UCF50), UCF-50, UCF-101. This results in a total of 88 training and evaluation runs.

We will introduce each of proposed models in detail below for a better understanding of its architecture and underlying reasoning principles.

A. CNNs

CNNs have achieved superior performance on many complex vision tasks in recent years. They are used as feature extractors to extract features from video frames. CNNs are mainly used in video surveillance, image retrieval/classification, and object detection for diverse applications (e.g. self-driving cars and underwater pipeline inspection). CNNs can also be used in other deployments to tackle voice recognition or acoustic scene classification.

Some basics of CNNs are as follows. CNNs embrace several different types of hidden layers for feature exaction with millions of parameters, allowing them to learn complicated objects and patterns. It uses convolution and pooling processes to "sub-sample" the given input before applying an activation function. The image is "scanned" by applying convolutions, i.e., filters., which are learnable.

Owing to the impressive performance of CNNs, in this research, we adopt pre-trained well-known CNN architectures, i.e. ResNet101, GoogLeNet and VGG16, for action recognition. All these models are pre-trained on the ImageNet dataset. These networks are then re-trained using either image frames or optical flow inputs to develop spatial or temporal CNNs. Specifically, we develop three spatial CNNs (ResNet101, GoogLeNet and VGG16) and one motion CNN (ResNet101).

In addition, we use Stochastic Gradient Descent (SGD) as the optimizer for all models. Cross-Entropy Loss is used as the objective function to calculate the gradient descent for weight adjustment. The learning rate scheduler, i.e. ReduceLROnPlateau, is used in training for all models in our study. A maximum number of 50 epochs is used to train all networks. A GPU with 32GB GPU RAM is used in our experiments.

1) Motion-CNN – ResNet101: A pre-trained ResNet101 [15] is re-trained using the optical flow images extracted from UCF101 to construct the Motion-CNN model. In other words, it is re-trained on motion streams of videos. Motion streams are extracted from the actual videos in the respective UCF datasets. To be precise, we aim to capture motion details (e.g. movement of humans, objects and camera views) to inform action classification. This Motion-CNN is adopted in Average Fusion to further boost classification accuracy in combination with different types of Spatial-CNNs.

During transfer learning using motion inputs, the weights of the pre-trained ResNet101 model are further adjusted to satisfy the requirements of action recognition tasks.

The Motion-CNN adopts a different dataloader from those of other models. It is called Motion_DataLoader, for the purposes of loading a video as it needs to convert it into optical flow outputs.

There are two transformations applied in Motion_DataLoader, i.e. the first is .resize([224, 224]), which resizes the data input, and the second is .ToTensor(), which converts the data to tensor to be used with a GPU for faster processing.

The DataLoader class from Pytorch is used in such a way that it avoids blocking computation code with data loading. The num_workers argument is used and the value is set to 8, which means there would be 8 processes simultaneously loading the incredibly large dataset into memory for faster processing.

2) Spatial-CNN – ResNet101: Similar to the above Motion-CNN, this Spatial-CNN [15] also uses pre-trained ResNet101 as the backbone. The key difference is that it is trained on image frames of videos instead of videos themselves. This Spatial-CNN model has comparatively lower number of parameters when compared with those of the Motion-CNN Model. In comparison with the Motion-CNN, this Spatial-CNN is faster to train because the inputs are image frames of the video, instead of videos. Out of three pre-trained models used in this research (i.e. ResNet101, GoogleNet, VGG16), ResNet101 has the highest top-1 accuracy among these three models on the ImageNet dataset. For the transfer learning model based on ResNet101, it also achieves more impressive performance for action recognition.

3) Spatial-CNN - GoogleNet: Another Spatial-CNN is constructed using transfer learning with pre-trained GoogleNet as the underlying network. GoogleNet is equipped with a number of Inception modules for effective feature learning. Out of three pre-trained models used in this work (i.e. ResNet101, GoogleNet, VGG16), GoogleNet has the second highest top-1 accuracy for image classification on ImageNet. When fine-tuning it using action frames, GoogleNet shows reasonable capabilities in tackling spatial structure extraction and action classification.

4) Spatial-CNN – VGG16: The third Spatial-CNN is implemented based on the ImageNet pre-trained VGG16 [18].

In VGG16, the network depth is increased by using small 3x3 convolutional filters, which shows a significant improvement on diverse image classification tasks. Out of three pre-trained models, VGG16 has the lowest top-1 accuracy on the ImageNet dataset but outperformed other well-known deep networks. We also re-train this VGG16 network using diverse video action frames for distinguishing different actions.

B. Hybrid CNN-RNN Models

RNN models are designed to extract temporal information by using data in a sequence to improve prediction accuracy. Essentially, RNNs take an input, and then re-use the hidden h_t activations of previous/later nodes in the sequence to influence the output. This is very important in action recognition, as we need temporal predictions which are based on the previous frames.

Besides the above transfer learning models, encoder-decoder architectures based on CNN and RNN are also developed for action recognition using video inputs directly. Specifically we use different pre-trained CNNs mentioned above (ResNet101, GoogleNet, and VGG16) as the encoders and different RNNs (LSTM, BiLSTM, GRU, and BiGRU) as the decoders. A total number of 12 resulting CNN-RNN models are implemented by combining the aforementioned CNNs and RNNs. We elaborate the construction of these hybrid networks below.

1) CNN-LSTM: The first series of hybrid networks are referred as CNN-LSTM, where each of three aforementioned CNN models (i.e. ResNet101, GoogleNet and VGG16) is used as the encoder, and LSTM is used as the decoder. Instead of using image frames or optical flow as inputs, it employs video as input directly for classifying different actions.

Specifically, the model is composed with CNN and LSTM networks. The CNN model is first used to extract features from the input data, which are video frames in our case, while the LSTM model is then used to learn sequential patterns from spatial features learned by CNN to distinguish different actions. In addition, LSTM has a cell state which transfers the relative information all the way till the end of the sequence chain (which can be considered as the memory). This information is added/removed to the cell states via gates.

In a CNN-LSTM architecture, CNN is integrated with LSTM, whose output is then connected with a dense layer for outputting a prediction. The spatial features are thus extracted and interpreted across time steps.

2) CNN-BiLSTM: We combine each of CNNs with BiLSTM, i.e. CNN-BiLSTM, for developing the second set of hybrid networks. BiLSTM is an extension of LSTM, as it has two LSTM layers, to learn the data from backwards (future to past) and forwards (past to future) passes respectively. Benefits of incorporating a BiLSTM network is to extract sequential information in both directions, whereby the model can analyse the data backwards and forwards which are vital for action recognition [19], [20], [21], [22].

Specifically, the LSTM model only has the forward layer, which indicates that it can only go from past to future, i.e., in only one direction. For BiLSTM model, in comparison with LSTM, it employs both forward and backward LSTM layers to interpret different actions in two directions.

3) CNN-GRU: We subsequent combine CNNs with GRU (CNN-GRU) for the third set of hybrid network generation. In our experiments, we notice that there are improvements when a GRU model is used instead of LSTM, because it fixes the problem with LSTM which is Short-Term Memory. Specifically, in LSTM, the layers stop learning when a small gradient update (in earlier layers) is obtained. The outcome of this is that the "forget gate" forgets these old layer inputs in longer sequences. That is why the name Short-Term Memory. GRUs have different types of gates and these gates can learn which data in a longer sequence is important or unimportant and will only forget the unimportant data. This helps the predictions as it only keeps the relevant information, hence better accuracy.

Therefore, unlike LSTM, GRU does not have cell state, in fact it has hidden states which are used to transfer information. GRU is also a slightly faster than LSTM, as it uses some tensor operations when running on a GPU machine. In comparison with LSTM, GRU also has less parameters to train, which speeds up training/evaluation as well.

4) CNN-Bidirectional-GRU: CNNs are also combined with BiGRU (i.e. CNN-BiGRU) to develop the fourth set of hybrid networks. Its working mechanism is very similar to that of CNN-BiLSTM. Similar to BiLSTM, BiGRU consists of two GRU layers with backwards (future to past) and forwards (past to future) passes for sequential feature learning. BiGRU has relatively less parameters to train as compared to those of BiLSTM.

C. Average Fusion

Average Fusion is an aggregation architecture inspired by [15], where two separate spatial and temporal streams, i.e. Spatial-CNN and Motion-CNN, are combined by a late fusion.

This research takes a similar approach. Instead of combining only two network streams, we combine one Motion-CNN with three Spatial-CNN streams, i.e. ResNet101, GoogleNet, and VGG16, by a late fusion.

When these fusion architectures are trained from scratch, they achieve better accuracy than all other singular architectures used prior to this section. We introduce each of the fusion architectures below.

1) Average Fusion of Motion-CNN + Spatial-CNN: We first develop a baseline Average Fusion model. It consists of traditional two streams [15]. The first stream is Motion-CNN, which is trained on optical flow input extracted from videos and the second stream is Spatial-CNN, which is trained on video frames. This hybrid model is used as the baseline to compare with other fusion models introduced below.

In this research, we re-train both Motion-CNN and Spatial-CNN on the action recognition dataset, and store the pickle files of both models (i.e. their best predictions, which are evaluated after every epoch and the best one is saved) on the disk for future use. While combining both models for fusion prediction, we load both the model predictions from the pickle

file, combine them with softmax and evaluate the results of top1 and top5 predictions.

The following models are used in this fusion scheme. 1. Motion-CNN (ResNet101) and 2. Spatial-CNN (ResNet101).

2) Average Fusion CNNs: Average Fusion of multiple Spatial-CNNs with one Motion-CNN is an evolved architecture hypothesis based on the traditional two-stream model. Specifically, we combine three Spatial CNNs with one Motion-CNN, to further improve classification accuracy.

This fusion model has achieved by far the best accuracy result among all 22 models proposed in this research.

Specifically, the following models are combined under this fusion scheme. 1. Motion-CNN (ResNet101), 2. Spatial-CNN (ResNet101), 3. Spatial-CNN (GoogleNet) and 4. Spatial-CNN (VGG16).

3) Average Fusion CNN-LSTMs: The strategy of Average Fusion CNN-LSTMs is similar to that of Average Fusion CNNs. All the CNNs models are replaced by their CNN-LSTM methods.

Therefore, this scheme includes the fusion of the following models. 1. Motion-CNN (ResNet101), 2. Spatial-CNN-LSTM (ResNet101), 3. Spatial-CNN-LSTM (GoogleNet) and 4. Spatial-CNN-LSTM (VGG16).

Similar to other Average Fusion models, in this fusion scheme, all CNN-LSTM models have to be trained prior to the fusion, hence the name late fusion.

4) Average Fusion CNN-Bidirectional-LSTMs: This Average Fusion network uses several CNN-BiLSTM models combining with the Motion-CNN. Specifically, it includes the fusion of: 1. Motion-CNN (ResNet101), 2. Spatial-CNN-BiLSTM (ResNet101), 3. Spatial-CNN-BiLSTM (GoogleNet) and 4. Spatial-CNN-BiLSTM (VGG16).

5) Average Fusion CNN-GRUs: The Average Fusion CNN-GRUs scheme includes the fusion of the following networks. 1. Motion-CNN (ResNet101), 2. Spatial-CNN-GRU (ResNet101), 3. Spatial-CNN-GRU (GoogleNet) and 4. Spatial-CNN-GRU (VGG16).

6) Average Fusion CNN-Bidirectional-GRUs: Similarly, the Average Fusion CNN-BiGRUs mechanism includes the fusion of the models below. 1. Motion-CNN (ResNet101), 2. Spatial-CNN-BiGRU (ResNet101), 3. Spatial-CNN-BiGRU (GoogleNet) and 4. Spatial-CNN-BiGRU (VGG16).

We subsequently evaluate the above six fusion models along with other transfer learning CNNs, and hybrid CNN-RNN methods using several UCF video action datasets. The detailed experimental studies are provided below.

IV. EVALUATION

In this research, we employ four video action datasets for model evaluation, i.e. UCF11, UCF25 (half of UCF50), UCF50 and UCF101. Evaluation results of all 22 models are provided in the subsequent sections.

A. CNNs

As introduced earlier, one Motion-CNN and three Spatial-CNNs are implemented in our experiments. These CNNs are pre-trained on ImageNet. We re-train the last fully connected layer with other layer weights maintaining the same as before (i.e. the pre-trained weights) in each network. The categories are updated as per the dataset used (e.g. 101 categories for UCF101).

The top1 prediction accuracies of all these CNN Models are provided in Table I.

TABLE I
CNNs RESULTS

Model Name	*UCF-11*	*UCF-25*	*UCF-50*	*UCF-101*
ResNet101 (Spatial)	90.43280029	87.5920105	81.68865204	79.91012573
VGG16 (Spatial)	85.42140961	83.17560577	73.82585907	72.21781921
GoogleNet (Spatial)	84.51024628	84.33228302	75.46173859	74.04176331
Resnet101 (Motion)	71.52619171	76.86645508	59.84168625	61.14195251

The highest accuracy is achieved by Spatial-ResNet101, as it also has the best accuracy in its pre-trained form on the ImageNet dataset among all the adopted deep architectures.

We can also see that, as the number of categories increases, from 11 to 25, 50, and then to 101, the performance of each model decreases. We can therefore conclude that there is a relation between the number of action categories as well as the size of datasets and the performance of the models.

B. CNN-RNNs

RNNs are used along with CNNs to improve the performance owing to the better extraction of temporal dynamics.

This research uses LSTMs and GRUs and their bidirectional versions for temporal feature extraction. LSTMs has short-term memory and GRUs in theory are meant to have better memory management than LSTMs. We analyse the results of each type of hybrid networks below.

1) CNN-LSTM and CNN-BiLSTM: For CNN-LSTM, LSTM is used as the last second layer, just before the fully connected layer. We can see from Table II that ResNet101-LSTM achieved the highest accuracy rate among all CNN-LSTM models for all the test datasets. Note that ResNet101 has the largest number of parameters. Again, for each network, as the number of action categories increases, the network performance decreases owing to increasing complexity of the classification tasks.

TABLE II
CNNs WITH LSTM

Model Name	*UCF-11*	*UCF-25*	*UCF-50*	*UCF-101*
ResNet101-LSTM	83.82688141	86.33017731	73.50923157	66.21728516
GoogleNet-LSTM	69.47608185	70.55731201	53.08707047	45.91593933
VGG16-LSTM	53.98633194	78.54889679	68.49604034	65.23922729

Now, we combine the same CNN models with BiLSTM layers with the results provided in Table III. ResNet101-BiLSTM obtained the best performance among all the hybrid networks. In some cases, CNN-BiLSTM models outperform CNN-LSTM methods because of the employment of both forward and backward directions for temporal information extraction.

TABLE III
CNNs WITH BiLSTMs

Model Name	*UCF-11*	*UCF-25*	*UCF-50*	*UCF-101*
ResNet101-BiLSTM	87.01594543	85.59411621	72.71767426	68.5170517
GoogleNet-BiLSTM	65.14806366	75.39432526	56.25329590	43.2989693
VGG16-BiLSTM	21.41230011	76.97161102	60.15830994	54.2162323

2) CNN-GRU and CNN-BiGRU: GRUs have better schemes for memory management and have less parameters when compared to those of LSTMs. All of the above CNNs are also combined with GRUs. Table IV shows their top1 prediction results. The empirical results indicate significant performance improvements of CNN-GRU over those of CNN-LSTM and CNN-BiLSTM.

TABLE IV
CNNs WITH GRUs

Model Name	*UCF-11*	*UCF-25*	*UCF-50*	*UCF-101*
ResNet101-GRU	88.15489960	88.85383606	81.00263977	80.9410553
GoogleNet-GRU	71.07061768	82.43953705	67.81002808	62.4900856
VGG16-GRU	70.15945435	84.85804749	73.24538422	70.8961182

Subsequently, the CNNs are also combined with BiGRU, which has two GRU layers with forward and backward passes. The results for CNN-BiGRUs are provided in Table V.

TABLE V
CNNs WITH BiGRUs

Model Name	*UCF-11*	*UCF-25*	*UCF-50*	*UCF-101*
ResNet101-Bi-GRU	92.02733612	87.9074707	82.63851929	79.83082581
GoogleNet-Bi-GRU	74.71526337	82.22923279	68.28495789	60.34893036
VGG16-Bi-GRU	37.81320953	74.44795227	61.68865204	66.87813568

C. Average Fusions

We conduct diverse late fusions of the proposed transfer learning and hybrid networks. The detailed results are provided below.

1) Average Fusion of CNNs: The Average Fusion of CNNs is the late fusion of three Spatial-CNNs (including Spatial-ResNet101, Spatial-GoogleNet and Spatial-VGG16) combined with Motion-ResNet101.

As indicated in Table VI, this multiple-stream Average Fusion scheme has achieved the best performance among those of all the 22 models. In particular, it outperforms the original baseline two-stream fusion method (1 Spatial-CNN + 1 Motion-CNN)) significantly as indicated in Table VI.

For example, this fusion scheme achieves 94.99% on UCF-11, 92.43% on UCF-25, 84.64% on UCF-50, and 84.88% on UCF-101. It benefits from the combination of multiple spatial and temporal streams which provide sufficient complementary information for action recognition.

TABLE VI
AVERAGE FUSION CNNs

Model Name	*UCF-11*	*UCF-25*	*UCF-50*	*UCF-101*
Average Fusion (Two-Stream)	88.61047363	90.01051331	85.06596375	84.37747955
Average Fusion CNNs	94.98860931	92.42902374	84.64379883	84.87972260

2) Average Fusion of CNN-LSTMs & Average Fusion of CNN-BiLSTMs: Similarly, Average Fusion of Motion-CNN with all other CNN-LSTM and CNN-BiLSTM models is also evaluated. The results are shown in Table VII.

TABLE VII
AVERAGE FUSION CNN-LSTMs & AVERAGE FUSION CNN-BiLSTMs

Model Name	*UCF-11*	*UCF-25*	*UCF-50*	*UCF-101*
Average Fusion CNN-LSTMs	86.33257294	90.01051331	82.32189941	81.20539093
Average Fusion CNN-BiLSTMs	84.96582794	89.27445221	79.6306076	79.11710358

As observed in Table VII, late fusion of Motion-CNN and CNN-LSTM models has better accuracy rates on the test datasets as compared to those of Motion-CNN combined with the CNN-BiLSTMs models, despite the fact that some CNN-BiLSTM models achieved better results than those of respective CNN-LSTM methods.

3) Average Fusion of CNN-GRUs and Average Fusion of CNN-BiGRUs: In our experiments, CNN-GRU models combined with Motion-CNN are supposed to have better performance, with CNN-BiGRU models integrating with Motion-CNN having the potential to achieve even better results. We test this hypothesis by late fusion of Motion-CNN with all CNN-GRU and CNN-BiGRU models, respectively. The detailed results are shown in Table VIII.

TABLE VIII
AVERAGE FUSION CNN-GRUs & AVERAGE FUSION CNN-BiGRUs

Model Name	*UCF-11*	*UCF-25*	*UCF-50*	*UCF-101*
Average Fusion CNN-GRUs	88.38269043	91.79811096	84.96041870	85.69918060
Average Fusion CNN-BiGRUs	90.88838196	90.32597351	85.85752106	84.45677948

When combining Motion-CNN with spatial CNN-RNN methods, we can see that GRU models are in fact better performing RNN models in comparison with LSTM models in our experimental studies.

We summarize the results of all the proposed 22 models in Table IX, which are ranked based on model performance.

TABLE IX
ALL RESULTS OF THE PROPOSED 22 MODELS

Model Name	*UCF-11*	*UCF-25*	*UCF-50*	*UCF-101*
Average Fusion CNNs	94.98860931	92.42902374	84.64379883	84.87972260
ResNet101-BiGRU	92.02733612	87.90747070	82.63851929	79.83082581
Average Fusion CNN-BiGRU	90.88838196	90.32597351	85.85752106	84.45677948
ResNet101 (Spatial)	90.43280029	87.59201050	81.68865204	79.91012573
Average Fusion (Two-Stream)	88.61047363	90.01051331	85.06596375	84.37747955
Average Fusion CNN-GRU	88.38269043	91.79811096	84.96041870	85.69918060
ResNet101-GRU	88.15489960	88.85383606	81.00263977	80.94105530
ResNet101- BiLSTM	87.01594543	85.59411621	72.71767426	68.51705170
Average Fusion CNN-LSTM	86.33257294	90.01051331	82.32189941	81.20539093
VGG16 (Spatial)	85.42140961	83.17560577	73.82585907	72.21781921
Average Fusion CNN-BiLSTM	84.96582794	89.27445221	79.6306076	79.11710358
GoogleNet (Spatial)	84.51024628	84.33228302	75.46173859	74.04176331
ResNet101-LSTM	83.82688141	86.33017731	73.50923157	66.21728516
GoogleNet-BiGRU	74.71526337	82.22923279	68.28495789	60.34893036
ResNet101 (Motion)	71.52619171	76.86645508	59.84168625	61.14195251
GoogleNet-GRU	71.07061768	82.43953705	67.81002808	62.49008560
VGG16-GRU	70.15945435	84.85804749	73.24538422	70.89611816
GoogleNet-LSTM	69.47608185	70.55731201	53.08707047	45.91593933
GoogleNet-BiLSTM	65.14806366	75.39432526	56.2532959	43.29896927
VGG16-LSTM	53.98633194	78.54889679	68.49604034	65.23922729
VGG16-BiGRU	37.81320953	74.44795227	61.68865204	66.87813568
VGG16-BiLSTM	21.41230011	76.97161102	60.15830994	54.21623230

V. CONCLUSION

In this research, we propose spatial and temporal CNNs, hybrid CNN-RNN methods, as well as diverse late fusion schemes of these networks, for undertaking action recognition. A total of 22 models are proposed in this research. Since Motion-CNN and Spatial-CNNs often adopt different backbone architectures, when evaluated on different datasets, we found that the pre-trained model such as ResNet101 which has the most accuracy on its pre-trained ImageNet dataset, also shows the best accuracy rates on UCF datasets. Hence, ResNet101 as a pre-trained model always performs exceptionally well than other CNNs in our experiments. With respect to the hybrid CNN-RNN models, CNN-BiGRU is the most performing network in comparison with other CNN-RNN variants. Moreover, among the proposed 22 models, the Average Fusion CNNs scheme which is a late fusion of one Motion-CNN and three Spatial-CNNs has the highest accuracy rates on all UCF datasets. The results of this fusion strategy also outperform a number of existing state-of-the-arts in the field. For future work, other pre-trained models such as ViT_H_14, RegNet_Y_128GF, and ViT_L_16 have even higher accuracy rates than those of ResNet101 will be studied in combination with other spatial and temporal networks [23], [24], [25]. In addition, since other advanced transformer architectures (e.g. Swinv2) show even competitive performance, the fusion of spatial and temporal CNNs with such transformer networks will be exploited to further enhance performance.

REFERENCES

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

[2] Zhang, L., Lim, C.P. and Yu, Y., 2021. Intelligent human action recognition using an ensemble model of evolving deep networks with swarm-based optimization. Knowledge-based systems, 220, p.106918.

[3] Zhang, L., Lim, C.P. and Liu, C., 2023. Enhanced Bare-Bones Particle Swarm Optimization based Evolving Deep Neural Networks. Expert Systems with Applications, p.120642.

[4] Qiu, Z., Yao, T., Ngo, C.W., Tian, X. and Mei, T., 2019. Learning spatio-temporal representation with local and global diffusion. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 12056-12065).

[5] Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y. and 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).

[6] Hong, J., Cho, B., Hong, Y.W. and Byun, H., 2019. Contextual action cues from camera sensor for multi-stream action recognition. Sensors, 19(6), p.1382.

[7] Thatipelli, A., Narayan, S., Khan, S., Anwer, R.M., Khan, F.S. and Ghanem, B., 2022. Spatio-temporal relation modeling for few-shot action recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 19958-19967).

[8] Radevski, G., Moens, M.F. and Tuytelaars, T., 2021. Revisiting spatio-temporal layouts for compositional action recognition. arXiv preprint arXiv:2111.01936. 2021.

[9] Luo, C. and Yuille, A.L., 2019. Grouped Spatial-Temporal Aggregation for Efficient Action Recognition. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 5512-5521).

[10] Materzynska, J., Xiao, T., Herzig, R., Xu, H., Wang, X. and Darrell, T., 2020. Something-else: Compositional action recognition with spatial-temporal interaction networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 1049-1059).

[11] Du, W., Wang, Y. and Qiao, Y., 2017. Recurrent spatial-temporal attention network for action recognition in videos. IEEE Transactions on Image Processing, 27(3), pp.1347-1360.

[12] Plizzari, C., Cannici, M. and Matteucci, M., 2021. Skeleton-based action recognition via spatial and temporal transformer networks. Computer Vision and Image Understanding, 208, p.103219.

[13] Yu, T., Guo, C., Wang, L., Gu, H., Xiang, S. and Pan, C., 2018. Joint spatial-temporal attention for action recognition. Pattern Recognition Letters, 112, pp.226-233.

[14] Yang, H., Yuan, C., Zhang, L., Sun, Y., Hu, W. and Maybank, S.J., 2020. STA-CNN: Convolutional spatial-temporal attention learning for action recognition. IEEE Transactions on Image Processing, 29, pp.5783-5793.

[15] Simonyan, K. and Zisserman, A., 2014. Two-stream convolutional networks for action recognition in videos. Advances in neural information processing systems, 27.

[16] Kinghorn, P., Zhang, L. and Shao, L., 2018. A region-based image caption generator with refined descriptions. Neurocomputing, 272, pp.416-424.

[17] Ma, C.Y., Chen, M.H., Kira, Z. and AlRegib, G., 2019. TS-LSTM and temporal-inception: Exploiting spatiotemporal dynamics for activity recognition. Signal Processing: Image Communication, 71, pp.76-87.

[18] Slade, S., Zhang, L., Yu, Y. and Lim, C.P., 2022. An evolving ensemble model of multi-stream convolutional neural networks for human action recognition in still images. Neural computing and applications, 34(11), pp.9205-9231.

[19] Zhang, L., Lim, C.P., Yu, Y. and Jiang, M., 2022. Sound classification using evolving ensemble models and Particle Swarm Optimization. Applied Soft Computing, 116, p.108322.

[20] Tan, T.Y., Zhang, L. and Lim, C.P. 2020. Adaptive melanoma diagnosis using evolving clustering, ensemble and deep neural networks. Knowledge-Based Systems.

[21] Lawrence, T., Zhang, L., Rogage, K. and Lim, C.P., 2021. Evolving Deep Architecture Generation with Residual Connections for Image Classification Using Particle Swarm Optimization. Sensors, 21(23), p.7936.

[22] Kinghorn, P., Zhang, L. and Shao, L., 2019. A hierarchical and regional deep learning architecture for image description generation. Pattern Recognition Letters, 119, pp.77-85.

[23] Fielding, B., Lawrence, T. and Zhang, L., 2019, July. Evolving and ensembling deep CNN architectures for image classification. In 2019 International Joint Conference on Neural Networks (IJCNN) (pp. 1-8). IEEE.

[24] Xie, H., Zhang, L. and Lim, C.P. 2020. Evolving CNN-LSTM Models for Time Series Prediction Using Enhanced Grey Wolf Optimizer, IEEE Access. 8, p. 161519-161541.

[25] Slade, S., Zhang, L., Huang, H., Asadi, H., Lim, C.P., Yu, Y., Zhao, D., Lin, H., and Gao, R., (In Press). Neural Inference Search for Multiloss Segmentation Models. IEEE Transactions on Neural Networks and Learning Systems.