CoReHAR: A Hybrid Deep Network for Video Action Recognition

Akram Mihanpour , Mohammad Javad Rashti*, Seyed Enayatallah Alavi
Department of Computer Engineering, Shahid Chamran University of Ahvaz, Ahvaz, Iran
a-mihanpoor@stu.scu.ac.ir, {mohammad.rashti, se.alavi}@scu.ac.ir

Received: 2020/08/05 Revised: 2020/08/29 Accepted: 2020/09/27

***Abstract*— Automating the processing of videos in applications such as surveillance, sport commentary and activity detection, human-machine interaction, and health/disability care is crucial to their correct functioning. In such video processing tasks, recognition of various human actions is a pivotal component for the correct understanding of videos and making decisions upon it. Accurately recognizing human actions is a complex process, demanding high computing capabilities and intelligent algorithms. Several factors, such as object occlusion, camera movement, and background clutter, further challenge the task and its accuracy, essentially leaving deep learning approaches the only viable option for properly detecting human actions in videos. In this study, we propose CoReHAR, a novel Human Action Recognition method that employs both deep Convolutional and Recurrent neural networks on raw video frames. Using the pre-trained ResNet152 CNN, deep features are initially extracted from video frames. The sequential information of the frames is then learned using DB-LSTM RNN. Multiple stacked layers in forward and backward passes of the DB-LSTM provide increased network depth for higher accuracy. A number of techniques are also applied to improve CoReHAR's processing speed on heterogeneous GPU-enabled systems. The proposed method is evaluated using PyTorch, and is compared to the state-of-the-art methods, showing a considerable efficiency increase, with nearly 95% recognition accuracy measured as an average over all splits of the challenging UCF101 dataset.**

Keywords*— *Human Action Recognition; Deep Learning; Convolutional Neural Network; Recurrent Neural Network, Data Augmentation.

1. INTRODUCTION

The ability to recognize and distinguish among human actions in videos has widespread applications in today's social and technical life. Surveillance and security cameras are everywhere, and news feeds are hardly spread without an accompanying video. As the most influential means of communication and socialization, videos are also very popular in social networking platforms. Much valuable information can be retrieved from videos, elevating security at social and national levels, saving and enhancing people's lives, and increasing productivity in jobs.

With videos being the primary information source for computer vision (CV) applications, a machine's ability to recognize specific activities in a video stream (a.k.a action recognition, and human action recognition or HAR, when the actors are human), is pivotal to many real-life CV applications. To name a few, we can point to public and private security, area surveillance, IoT and smart fields, video information retrieval, human motion analysis and synthesis (for robotics, autonomous vehicles, and man-machine interaction), sports analysis, health/disability care, and gaming / entertainment [1]. With such a wide application spectrum, human action recognition is proven to be an integral part of computer vision research [1, 2].

Obviously, any manual extraction and analysis of such big data mines is out of equation, while even AI-enabled computing systems are challenged by the sheer volume, variety and production velocity of such videos. This is in addition to the inherent complexities of action recognition algorithms, and intra-class variation of human actions. Beside data and algorithmic complexities, HAR faces numerous challenges due to various adverse factors in real-world situations. Lack of annotations, similarity of visual content, capture noise, view point and camera movements (w.r.t. the actor), object occlusion, scaling, gesture variation, and variable imaging and lighting conditions are some of those factors, primarily stemming from field conditions and equipment technical limitations [3]. The main challenge in HAR is to come up with a representation of actions that provides both proper distinction among different actions, and accurate classification to detect actions of the same kind in the presence of adverse video conditions.

State-of-the-art automatic HAR algorithms [4, 5] remarkably reduce human load in analyzing large-scale video data [1]. With the emergence of deep-learning AI approaches, even more complicated features and concepts of a video stream can be extracted with significantly higher accuracy. While classical methods have struggled for accuracy, modern deep learning approaches such as Convolutional Neural Networks (CNN) and Long Short Term Memory (LSTM) Recurrent Neural Networks (RNN) greatly contribute to the success in image/video classification and voice recognition [6-8] . Modern computing platforms such as those with GPU-enabled hybrid processing, further enhance our experience by improving the time complexity and scalability of the algorithms, while providing real-time processing capabilities.

In this study, we propose a new method for HAR, called CoReHAR, which processes only raw video frames using a pre-trained CNN called ResNet152 and a Deep Bidirectional LSTM Network (DB-LSTM). The implementation results using the UCF101 dataset show that the proposed method has a high performance in HAR in terms of accuracy and speed.

The rest of this paper is organized as follows: Section II provides an overview of previous related work. Section III presents the proposed deep learning architecture. The implementation details are laid out in Section IV, and experimental results are reported and analyzed in Section V. Finally, Section VI provides conclusions and suggestions for future enhancements.

2. RELATED WORK

In this section, we classify and overview state of the art video-based human action recognition methods. We start by

covering classic models, such as those using spatio-temporal methods or holistic representations. Since classic models prove incompetent in differentiating complex actions in videos, deep learning models with their high classification accuracy are investigated next.

2-1. Classic Models

Classic models are based on handcrafted features and typically use local ones. There are generally two main classical approaches [9]. Early methods such as [10] use simple body representation models based on optical sources. Such methods are inflexible in facing challenges such as viewpoint movement and video disturbance. A number of other classic methods use holistic representations as a global representation of the human body and its movements [11]. Holistic representation methods such as Space-Time Volumes (STVs) were first introduced in [12] . In these methods, a three-dimensional space-time volume is obtained by stacking the image of the person, i.e., the foreground areas where the person is, in the direction of the time dimension. This type of representation uses a three-dimensional space (x, y, t) to represent objects, people, and the relationship between spatial and temporal information. The basis of identification methods using space-time is built on measuring the similarity of two different three-dimensional spaces. Instead of creating a three-dimensional space for each action, a two-dimensional pattern image is used for representing the action. Motion-Energy Image (MEI) binary image and Motion-History Image (MHI) numerical value image are examples of such pattern images [11]. Obviously, holistic representation methods can only learn a general pattern of the action and are unable to detect much of the details. To address this issue, some methods use local features [13].

Many studies have tackled action recognition problem by extracting spatio-temporal features [14, 15]. One of the basic spatio-temporal models is presented in [16], which provides a method for extracting key spatio-temporal points, i.e., regions of the image that are prominent and sudden changing regions in both space and time dimension.

A better approach is to use dense optical flows and classify such flows into motion bubbles [17]. Obviously, such a process tends to be very costly and time-consuming, despite yielding significantly higher accuracy [18, 19]. In most optical-flow-based methods, video representation is created using the bag of words (BoW) model and Fisher vector coding[20]. The authors in [20] present the use of a dense sampling method to create a dense trajectory. This technique fuses several different features with a particular structure, and is one of the strongest methods among the classic models. Despite being very slow, this technique is highly accurate, when compared to early deep-learning HAR solutions, and is considered as a base classical method.

Despite relatively acceptable performance of handcrafted-features-based approaches, research has recently been more interested towards the use of deep learning models, due to several challenges faced by classical approaches, including low accuracy and little flexibility in intra-group variations of actions.

2-2. Deep Learning Models

Deep learning models can extract more complicated concepts from videos by considering frame sequences. Many efforts have been made to involve the time information available in the video into CNN models, including the work in [21]. Deep-learning action recognition methods can be grouped in two categories: two-stream networks and space-time networks. Most existing deep-learning HAR methods use two-stream networks [22], one stream for spatial information and the other for inter-frame motion information. In [22], the frames are fed into a two-stream network with two inputs. The raw frame images are fed into the spatial stream, and the optical flow images are fed into the temporal stream.

Three-dimensional CNNs are examples of space-time networks introduced by Ji et al. [23]. Other space-time networks such as Recurrent Neural Networks (RNN), including LSTM, have also been used to incorporate temporal information into the video [24, 25].

In [26], the authors propose a recurrent two-stream spatial-temporal attention network for video-based human action recognition. They apply the developed spatial-temporal attention in the feature learning process, reaching up to 96% accuracy on the UCF11 dataset. In another attention based work, the authors in [27] use a dual-attention network (STDAN) architecture, which combines convolutional LSTM and fully-connected LSTM. Moreover, PCA is used to condense video-level features to low-dimension vectors. This work reports a 91% accuracy on split 1 of the UCF101 dataset, when combining STDAN with RGB difference method.

In [28], the authors propose an effective approach for spatiotemporal networks(3D ConvNets). They showed that C3D can model appearance and motion information simultaneously and outperforms the 2D ConvNet features on various video analysis tasks.

In [29], a motion vector CNN is proposed to accelerate deep learning speed for action recognition. To improve the recognition performance, the authors have developed three knowledge-transfer techniques to adapt the models of optical flow CNN to motion vector CNN, significantly boosting the recognition performance of the latter. In [30], DTMV-CNN is proposed to enable knowledge transfer from optical flow domain to motion vector domain. Performance of DTMV-CNN on three challenging datasets verifies the effectiveness of the training approach, and shows significantly higher performance compared to an MV-CNN, when trained from scratch.

The authors in [31] proposed an action recognition framework by utilizing CNN frame-level deep features and processing them through DB-LSTM. First, deep features are extracted from every sixth frame of the video, then the sequential information among frame features is learnt using DB-LSTM network.

Khalid and Yu [32] presented a multi-modal three stream network to utilize human body poses along with RGB frames and optical flows for action recognition. The complementary behavior of RGB, optical flow and pose is observed and analysed in their experiments. A major drawback in two-stream approaches is that the motion information is processed separately from the visual information. Compared to the two-stream approaches, both 3D and recurrent networks demonstrate high processing volumes and have a high number of training parameters, hence the need for very large datasets for training. Since the production of such video datasets is

onerous and costly, there is a need for methods that can optimally incorporate temporal information into the processing without requiring large training datasets. The method presented in this paper incorporates temporal information appropriately into processing while requiring less training data compared to spatio-temporal networks. A list of techniques and their accuracy is presented in Table 1.

3. THE PROPOSED HAR ALGORITHM

In this section, the proposed CoReHAR Model and its main components are discussed in detail, including the recognition of an action from the sequence of frames in video using DB-LSTM and features extraction through CNN for video frames using transfer learning. In deep learning, the processing model includes neural networks, the most important and widely used of which are convolutional neural networks and recurrent neural networks. We use these two deep neural networks subsequently in the proposed method. The model consists of two major sections: in the first section, a CNN is used for categorizing images and videos. The second section is a bidirectional RNN.

The motivation behind this idea is to improve the action recognition's time and memory consumption. For this we just use raw video frames and not to employ optical flow and motion vectors. Moreover, using the pre-trained ResNet152 CNN and DB-LSTM make paired with data augmentation and proper selection of input parameter values, make action recognition more accurate, efficient, and stable. In addition, the feature extraction step requires acceleration, which we propose doing it using multithreading.

Pretrained CNNs for feature extractionThe initial role of the CNN is to extract the desired features from each video frame, producing a feature vector whose dimensions are reduced after removing the undesirable features. Training a deep learning model to display images is a demanding process with heavy computations. The model requires thousands of images to be properly trained and become ready to act, plus a high processing power to execute functions such as adjusting the CNN model weights.

To avoid unnecessarily long delays due to training, we use learning transfer where a pre-trained model is employed. We use a pre-trained CNN network named ResNet. Residual networks (ResNet) [33] were proposed as a family of multiple deep neural networks with similar structures but different depths. ResNet introduces a structure called residual learning unit to alleviate the degradation of deep neural networks. This unit's structure is a feedforward network with a shortcut connection which adds new inputs into the network and generates new outputs. The main merit of this unit is that it produces better classification accuracy without increasing the complexity of the model. We select ResNet152 as it achieves the best accuracy among ResNet family members [33]. Fig.1 illustrates the basic architecture of ResNet152.

TABLE 1 . COMPARISON OF VARIOUS ACTION RECOGNITION TECHNIQUES

Paper	Year	Method	Accuracy (%)	Dataset(s)
Fernando et al. [21]	2016	Rank pooling CNN	87	UCF-sports
Simonyan and Zisserman [22]	2014	Two-stream CNN	88	UCF-101
			59.4	HMDB-51
Ji et al. [23]	2012	3D Convolution	90.2	KTH
Du et al. [24]	2015	HBRNN-L (skeleton based)	94.49	(MSR Action3D)
Srivastava et al. [25]	2015	LSTM autoencoder	75.8	UCF-101
			44.1	HMDB-51
Dai et al. [26]	2020	Two-stream LSTM(optical flow + attention)	76.3	HMDB-51
			98.6	UCF-sports
Zhang et al. [27]	2020	STDAN + RGB difference	91	UCF-101
			60.4	HMDB-51
Tran et al. [28]	2015	C3D generic descriptor	90.4	UCF-101
Zhang et al. [29]	2016	EMV+RGB-CNN	86.4	UCF-101
Zhang et al. [30]	2018	DTMV+RGB-CNN	87.5	UCF-101
Ullah et al. [31]	2017	DBLSTM+CNN	91.21	UCF-101
Khalid and Yu [32]	2019	Three-stream CNN GT pose	83.1	HMDB-51

ResNet152 consists of 152 layers, including 150 convolutional layers, a pooling layer and a fully connected layer. It is pre-trained on a very large ImageNet[34] dataset with over 15 million images. In very deep architectures such as ResNet and its variants, the number of learnable parameters increases with depth. This increases the computational overhead, and as a result, reduces the speed of network training and learning. To avoid this problem, a technique called bottleneck is used in ResNet architecture, where the number of feature maps is reduced, while the dimensionality is increased. After the first two layers in ResNet, the input space is compressed from 224x224 size to 56x56. The input blocks are then presented to three layers of convolution for processing. Batch normalization is a technique for improving the performance and stability of neural networks, which is applied to the input blocks in each layer. The idea is to normalize the input of each layer with mean and variance. By normalizing data, learning and error rate reduction are performed faster.

After normalizing the batches in each layer, the ReLU nonlinear activation function is used, which better updates the weights in the layer. These updates reduce the computation overhead and further improve the learning time.

3-1. Deep Bidirectional LSTM

Video contains sequential data in which movements are displayed in the visual content of many frames. Thus, the sequence of frames is essential in detecting actions. An LSTM recurrent neural network is able to make decisions about maintaining current memory through input, output, and forget gates. Intuitively, if the LSTM unit detects an important feature in the input sequence in initial steps, it can easily transmit this information over a long distance, and receive and maintain potential long-term dependencies.

In frame sequences, a frame does not only depend on its previous frames, but may also depend on its subsequent ones. In this case, to increase the accuracy and efficiency of the model, it is advisable to have a structure that processes the input frames in both directions. LSTM networks can be bidirectional[35]. Therefore, a Deep Bidirectional LSTM (DB-LSTM) network is employed here to help detecting complex sequential patterns hidden in video frames by considering frame dependencies in both directions. In a DB-LSTM structure, two LSTM networks simultaneously process input in both directions. The forward network reads and processes the input in forward direction, while the backward network reads and processes the same input in the backward direction. The basic architecture of DB-LSTM is illustrated in Fig.2.

Training large datasets with complex sequence patterns (such as video data) cannot be achieved using an LSTM cell [31]. Therefore, in the proposed CoReHAR method, we use Multi-layer LSTM (ML-LSTM) by aggregating multiple LSTM cells to learn long-term dependencies in video data. This allows the recurrent neural network to extract higher level sequence information.

Once the appropriate features are extracted from the ResNet CNN, the features are passed to the LSTM network. The output of each layer in the bi-directional RNN represents the action performed in the frame processed in that layer. Finally, the input video action tag is determined by the maximum frequency of the frame tags. In the classification step, the highest score tag is displayed as the output and the action recognized in the input video. Fig.3 illustrates a big picture of the CoReHAR model, including its internal structure, as described above in this section.

4. Implementation Details

In this section, we introduce the dataset and implementation details used in our experiments. CoReHAR algorithm is implemented using the PyTorch framework[36]. The simulation in this research has been performed in 120 iterations on a machine with a dual core i5 CPU at 2.7GHz each, and 8GBs of RAM, fortified with a 1080 NVIDIA GeForce GTX GPU, including 8 GB of GPU RAM.

4-1. The Dataset

The UCF101 dataset is one of the most popular action recognition datasets, with real-world videos covering a wide set of action classes. The videos in UCF101 are carefully selected to encompass various challenges for action recognition algorithms. Containing 13320 videos taken from YouTube and divided into 101 classes, UCF101 includes videos that represent a wide variety of human actions such as music playing, makeup, various sports, personal care, and cooking, among others. These videos are grouped in five main categories: human-object interaction, body movement, human-to-human interaction, playing musical instruments, and sports. Each category contains 100 to 200 videos. Some UCF101 categories such as sports are further divided into sub-classes. The shortest video contains 28 frames, with frame dimensions of 320*240 [37].

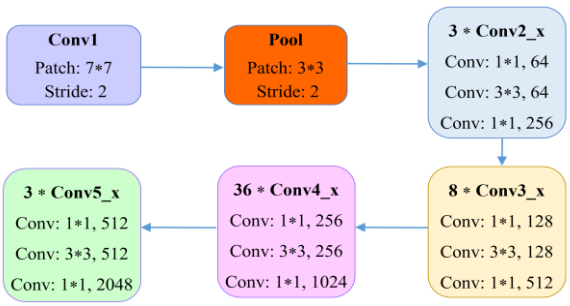

Fig. 1. The basic architecture of ResNet152

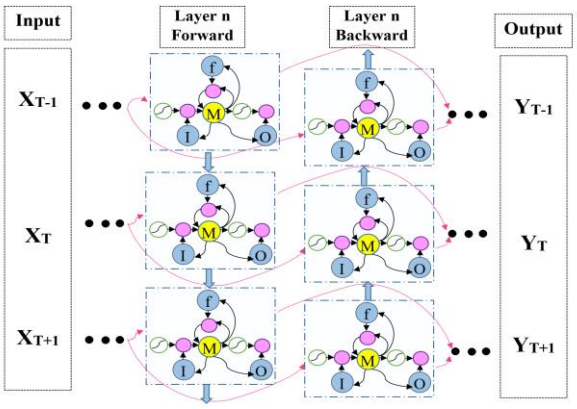

Fig. 2. The basic architecture of DB-LSTM

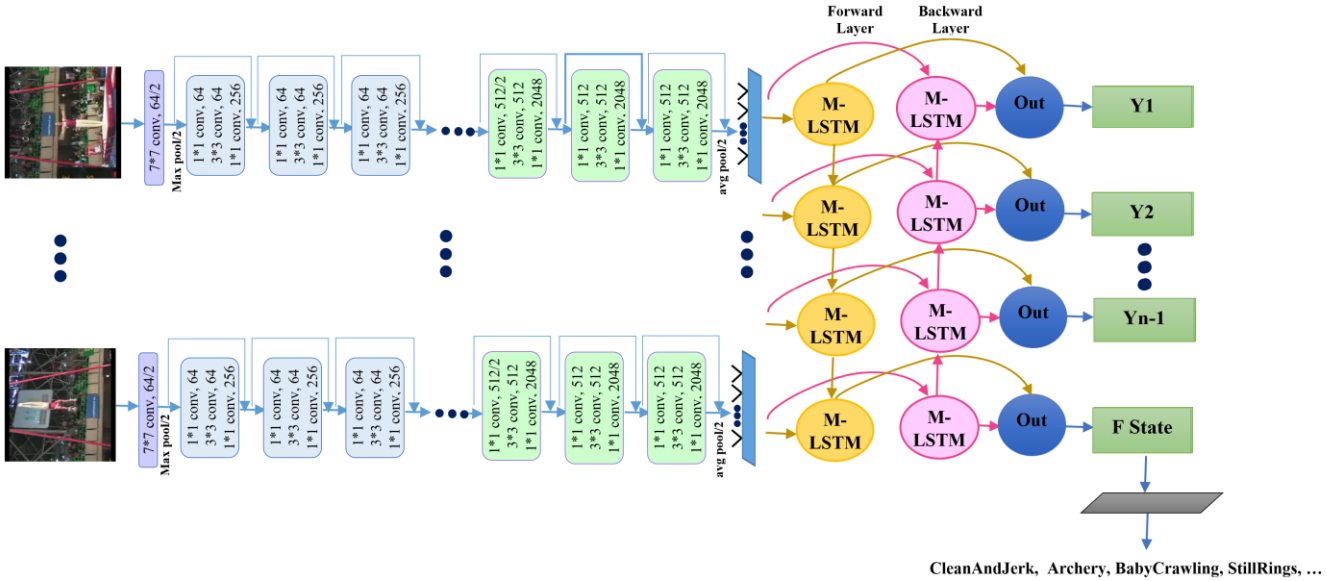

Fig. 3. Detailed structure of the proposed CoReHAR method

Fig.4 shows a number of UCF-101 dataset frames. In addition to the sub-optimal quality of many videos in the dataset, videos also have varying lighting conditions, actor gestures and viewing points, among other obstacles, as previously pointed to. Most sports videos also have a similar (green) background, which adds an extra challenge to the recognition process.

We first need to pre-process the videos by extracting the input video frames before feeding them to the proposed model. Libraries like OpenCV or FFmpeg can be used to extract video frames. However, we directly imported the pre-processed dataset used in [38]. In the UCF-101 dataset, action tags are non-numerical, thus requiring an encoding step. Label encoding is used due to its much lower number of zeros compared to the one-hot encoding, thereby reducing the overall network training overhead.

4-2. Cost Function

The accuracy of the learning process in the proposed model is crucial for the correct and timely classification of actions in videos. We need to evaluate the model, using a cost function that measures the model's accuracy. Once measured, we will then try to minimize the cost in order to maximize the model's performance.

The cost function quantity is defined as the amount of difference between the actual output and the network's expected output. There are various cost functions for classification problems, one of the most famous ones (that usually works well in binary and multi-class classification problems[36]) being the Cross Entropy function [39] as defined in (1):

$$\mathrm{CE}(y, s) = -\sum_c y_c \,.\, \log(s_\theta(x)_c) \qquad (1)$$

In (1), c represents the class index, yc is the actual value of the class and $s_\theta(x)_c$ is the output value of the model designed for input x.

4-3. Optimizer Function

As an optimization problem, we need to minimize the cost function in order to maximize our detection accuracy. Various optimization algorithms, including SGD, StepLR and Adam [36] were used for the proposed model. Based on our experimental results shown in Table 2, the Adam optimizer function is selected as it yields a better performance and accuracy compared to other optimizers.

Fig. 4. Sample action categories in the UCF-101 dataset

4-4. Data Augmentation

To reduce overfitting, we use data augmentation[40]. Three processing strategies have been applied to the network input frames for this purpose, namely:

- RandomRotation(5), which randomly rotates the images by a degree between 0 to 5,
- RandomCrop(crop_size), which crops the image by the crop_size as its dimension, and
- RandomHorizontalFlip(), which flips the image horizontally with the default probability of 50%,

where based on our experiments, yield higher accuracy compared to other methods.

With these three transformations, from each frame we produce three additional frames with a slightly different look but the same label. This technique has enabled our proposed model to learn robust and differentiated data, leading to higher accuracy.

Table 3 lists the internal parameters of the implemented method. The parameters are selected after running mutiple tests, evaluating the accuracy and resource usage of the model. For example, we chose 1×10^{-3} as the learning rate, then tested the model with different batch sizes. Considering the accuracy and the available memory, we chose 128 for the

batch size parameter. Finally, Fig.5 presents the flowchart of the proposed method, illustrating the mentioned steps in section III and IV. This flowchart represents the overall logical flow of our proposed algorithm, part of which is the deep learning model presented in Fig.3.

5. Experimental Results

We report our accuracy and timing results in two separate subsections. For testing and training, the dataset is divided into two parts: 75% for training and 25% for testing. Parts of the training data are randomly and variably used for validation. Due to GPU memory limitations, we need to use frame jumps to reduce memory usage. We have selected the four (4) frame jump option in our experiments, as it proves to yield the highest accuracy on our existing GPU.

5-1. Accuracy Analysis

The accuracy is reported in terms of the percentage of matching between the actual video tag values in question and the values detected by the proposed model. The accuracy results are compared to the state-of-the-art research, as outlined in Table 4. The results indicate a meaningful improvement in the accuracy of action recognition using the proposed CoReHAR algorithm. Particularly, the superiority is observed when compared to the two-stream methods[22] with one flow on raw video frames and the other on optical flow frames. The accuracy is also higher than the CNN based methods such as deeply-transferred motion vector[30], 3D CNNs[28], and factorized spatio-temporal CNNs[41]. Moreover, CoReHAR proves more accurate and faster than the study in [31], despite its smaller number of iterations.

The use of ResNet152 network besides data augmentation could be the main reason for increasing accuracy. A pre-trained ResNet152 network extracts frame features and passes the output to the bidirectional long-term memory RNN, which in turn extracts inter-frame sequence information. Such a network enables the sequences of video input frames to be processed in both directions (forward and backward), thereby increasing the probability of an accurate action classification. By the use of techniques such as data augmentation, this is further improved, while reducing the probability of overfitting. Moreover, the proper selection of input parameters values is another contributor to the accuracy score.

Fig.6 presents the training and test steps accuracy and loss values at each iteration of the implementation. As observed, with the increasing number of implementation steps, the accuracy is increased and the loss value is decreased, which is an evidence that the training is well performed and no overfitting is present. We also optimize the batch size to improve the execution performance of the proposed model. The batch size training parameter refers to the number of videos that are input to the network at a time. This parameter, not related to the number of videos in each class, is usually determined by the size of the data set and the amount of memory available. Current research shows that the batch size parameter has a crucial effect on the accuracy of action recognition. We selected the optimal batch size with various experiments, limited by the available GPU memory. The results of these experiments are presented in Fig.7. According to these results, the greater the batch size value, the higher the action recognition accuracy. On the other hand, the large batch size value leads to huge memory costs. Based on these experiments, the batch size of 128 has been selected for the main executions.

In addition, to demonstrate the efficiency of the method for different action classes, we present the confusion matrix for 40 classes of the split1 of UCF101 dataset in Fig.8. For example, in this matrix, the class "HandstandWalking" in our method achieves 97% true positive prediction and the class "CliffDiving" has only 1% false prediction. As observed, the intensity of true positives (diagonal) is high for all presented categories, indicating the high capability of the proposed method for human action recognition in video. Overall, we observe an average of 95% accuracy among the classes across all splits of the dataset, as compared to other methods on UCF-101 in Table 5.

5-2. Timing Analysis

Recognition timing becomes significant in real-time implementations, where the system needs to perform an action in response to the detected activities. In order to understand the timing behavior of CoReHAR, we have

TABLE 2. Comparison of Optimizer Functions

Optimizer Function	Average Time Complexity	Average Accuracy
Adam	1.63 sec	94.55
SGD	1.55 sec	91.36
StepLR	1.87 sec	92.23

TABLE 3. Deep Network Optimized Parameters

Parameter	Value
ResNet152 Parameters	
1^{st} & 2^{nd} fully connected layers' neurons	1024 & 768
Feature vector size	512
Input frame size	224
Random dropout rate	0.0
DB-LSTM Parameters	
No. of hidden layers	3
Hidden vector size	512
Feature vector size	256
Training Parameters	
Dataset no. of classes (K)	101
Epochs	120
Learning rate	$1x10^{-3}$
Batch size	128
Number of worker threads	4

TABLE 4. Comparison of average recognition score of the proposed method with other methods on UCF-101

Method	UCF-101 Accuracy
C3D(3net)[28]	85.2%
Two-stream CNNs[22]	88.0%
EMV+RGB-CNN[29]	86.4%
RLSTM-g3[42]	86.9%
Multiple dynamic images[43]	89.1%
Factorized spatio-temporal CNNs[41]	88.1%
DTMV+RGB-CNN[30]	87.5%
DBLSTM+CNN[31]	91.21%
CoReHAR (ResCNN-DBLSTM)	94.79%

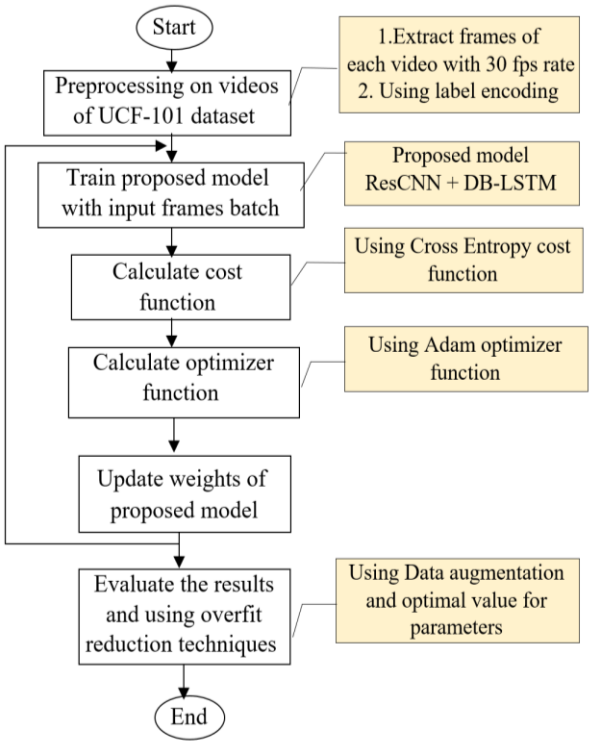

Fig. 5. Flowchart of proposed model

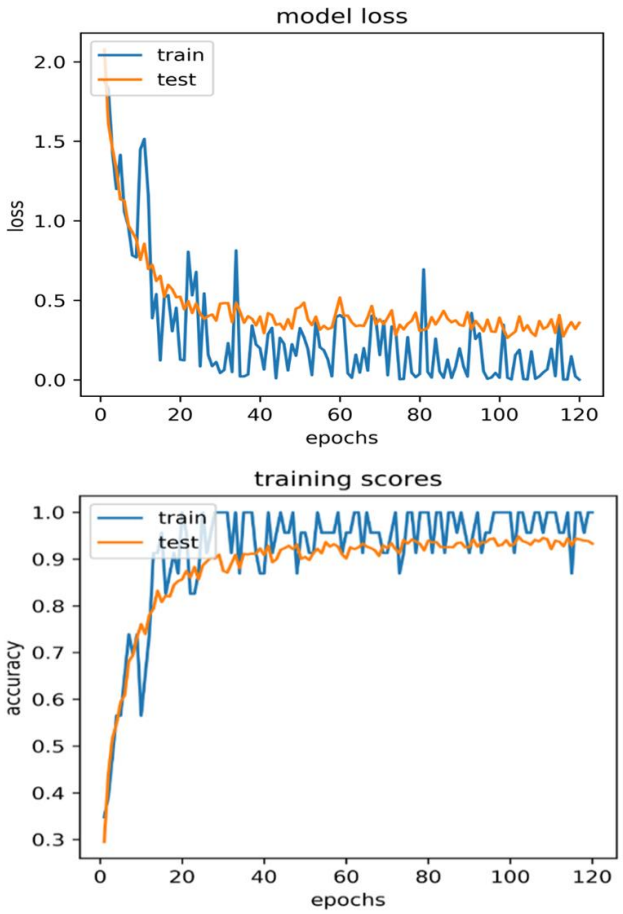

Fig. 6. Accuracy and loss diagrams of the proposed model in both training and testing steps

evaluated the implementation using different experiments with the various number of frame jumps. Table 6 shows the statistics of these experiments, using a batch size of 40.

We also use multithreading to improve the execution speed. Since the task of transferring data between the host machine and the GPU is a time-consuming task, we used multithreading to accelerate it. Based on experimental results presented in Table 5, a four-thread transfer is used in our main executions, which considerably increases the efficiency compared to the single thread case.

As for the processing speed with multithreading, a 1-second video clip takes approximately 0.11 sec per frame for feature extraction. Feeding the extracted features to DB-LSTM for classification takes 0.47 sec for a 30 frames per second video clip. Overall, our method takes approximately 0.58 seconds, yielding a 52 frames per second processing throughput, making it a good candidate for real-time action recognition.

6. Conclusion

In this study, CoReHAR, a new hybrid deep learning method called is presented for human action recognition in videos, where both CNN and RNN networks operate on raw video frames, instead of optical flow data. The proposed method initially extracts deep features of the video frames using the pre-trained ResNet152 CNN, to accelerate the learning process and improve the performance. Then, the frame sequence information is learned using the DB-LSTM RNN in both forward and backward transitions, followed by the final classification of the video.

TABLE 5. AVERAGE TIME COMPLEXITY AND ACCURACY ON DIFFERENT FRAME JUMPS FOR 30 FPS VIDEO CLIP

Frame Jump	Average Time Complexity	Average Accuracy
2	1.55 sec	94.2
4	1.25 sec	93.8
6	1.01 sec	92.2
8	0.88 ec	90.1

TABLE 6. AVERAGE TIME USING MULTITHREADING

Number of Threads	Average Transfer and Execution Time
1	2.45 sec
2	2.23 sec
3	1.95 sec
4	1.75 sec

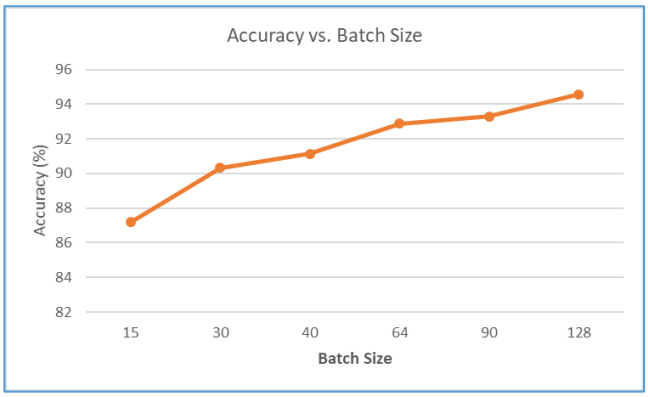

Fig. 7. CoReHAR Accuracy vs. Batch Size

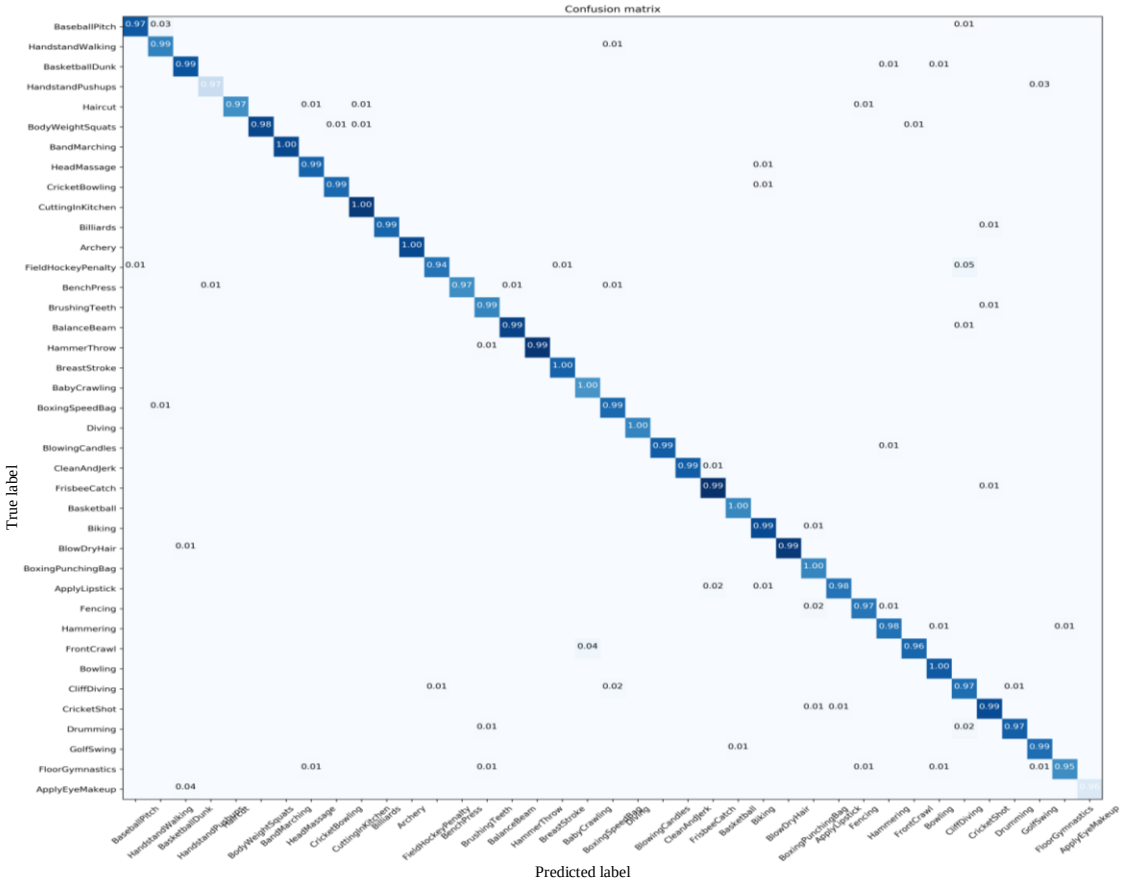

Fig. 8. Confusion matrix of UCF101 dataset for the proposed HAR method

Our implementation results using PyTorch on a GPU-equipped machine over the UCF101 dataset show that CoReHAR can perform more accurately, compared to state-of-the-art methods, while keeping the ability to run on video streams in real time.

Proper selection of input parameters, use of the pre-trained ResNET152, and techniques such as data augmentation have increased the performance of the proposed method. Moreover, using multiple threads in parallel, for data preparation and injection, reduces the training time. This makes our method fit for real-time visual data processing and can be used as an integral part of smart detection systems. In the future, we aim to focus on the effective features for the original input frames for action recognition. We also plan on test and analysis of networks such as GRU for different video classes and various datasets.

ACKNOWLEDGMENT

"This study was funded by Shahid Chamran University of Ahvaz (SCU), grant number SCU.EC98.30899. The authors would also like to thank SCU's Deep Learning Laboratory at the Department of Computer Engineering for the computing resources."

REFERENCES

[1] Y. Kong, and Y. Fu, "Human action recognition and prediction: A survey", *arXiv preprint arXiv:1806.11230*, 2018.

[2] S. Ranasinghe, F. Al Machot, and H. C. Mayr, "A review on applications of activity recognition systems with regard to performance and evaluation", *International Journal of Distributed Sensor Networks*, vol. 12, p. 1550147716665520, 2016.

[3] R. K. Tripathi, A. S. Jalal, and S. C. Agrawal, "Suspicious human activity recognition: a review", *Artificial Intelligence Review*, vol. 50, pp. 283-339, 2018.

[4] Y. Kong, S. Gao, B. Sun, and Y. Fu, "Action prediction from videos via memorizing hard-to-predict samples", in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[5] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, *et al.*, "Temporal segment networks: Towards good practices for deep action recognition", in *European Conference on Computer Vision*, 2016, pp. 20-36.

[6] G. Varol, I. Laptev, and C. Schmid, "Long-term temporal convolutions for action recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, pp. 1510-1517, 2017.

[7] K. Wang, X. Wang, L. Lin, M. Wang, and W. Zuo, "3d human activity recognition with reconfigurable convolutional neural networks", in *Proceedings of the 22nd ACM International Conference on Multimedia*, 2014, pp. 97-106.

[8] H. Gammulle, S. Denman, S. Sridharan, and C. Fookes, "Two stream lstm: A deep fusion framework for human action recognition", in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2017, pp. 177-186.

[9] S. Herath, M. Harandi, and F. Porikli, "Going deeper into action recognition: A survey", *Image and Vision Computing*, vol. 60, pp. 4-21, 2017.

[10] G. Johansson, "Visual perception of biological motion and a model for its analysis", *Perception & psychophysics*, vol. 14, pp. 201-211, 1973.

[11] A. F. Bobick and J. W. Davis, "The recognition of human movement using temporal templates", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, pp. 257-267, 2001.

[12] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes", in *Tenth IEEE International*

Conference on Computer Vision (ICCV'05) vol. 1, 2005, pp. 1395-1402.

[13] D. Weinland, R. Ronfard, and E. Boyer, "A survey of vision-based methods for action representation, segmentation and recognition", *Computer Vision and Image Understanding*, vol. 115, pp. 224-241, 2011.

[14] J. Liu, J. Luo, and M. Shah, "Recognizing realistic actions from videos in the wild" In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2009, pp. 1996-2003.

[15] M. Marszałek, I. Laptev, and C. Schmid, "Actions in context", in *CVPR 2009-IEEE Conference on Computer Vision & Pattern Recognition*, 2009, pp. 2929-2936.

[16] I. Laptev, "On space-time interest points", *International Journal of Computer Vision*, vol. 64, pp. 107-123, 2005.

[17] R. Cutler and M. Turk, "View-based interpretation of real-time optical flow for gesture recognition", in *Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition*, 1998, pp. 416-421.

[18] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping", in *European Conference on Computer Vision*, Springer, Berlin, Heidelberg, 2004, pp. 25-36.

[19] V. Kantorov and I. Laptev, "Efficient feature extraction, encoding and classification for action recognition", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2593-2600.

[20] H. Wang and C. Schmid, "Action recognition with improved trajectories", in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 3551-3558.

[21] B. Fernando and S. Gould, "Learning end-to-end video classification with rank-pooling", in *International Conference on Machine Learning*, 2016, pp. 1187-1196.

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

[23] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, pp. 221-231, 2012.

[24] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1110-1118.

[25] N. Srivastava, E. Mansimov, and R. Salakhudinov, "Unsupervised learning of video representations using lstms", in *International Conference on Machine Learning*, 2015, pp. 843-852.

[26] C. Dai, X. Liu, and J. Lai, "Human action recognition using two-stream attention based LSTM networks", *Applied Soft Computing*, vol. 86, p. 105820, 2020.

[27] Z. Zhang, Z. Lv, C. Gan, and Q. Zhu, "Human action recognition using convolutional LSTM and fully-connected LSTM with different attentions", *Neurocomputing*, vol. 410, pp. 304-316. 2020.

[28] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks", in *Proceedings of the IEEE International Conference on computer Vision*, 2015, pp. 4489-4497.

[29] B. Zhang, L. Wang, Z. Wang, Y. Qiao, and H. Wang, "Real-time action recognition with enhanced motion vector CNNs", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2718-2726.

[30] B. Zhang, L. Wang, Z. Wang, Y. Qiao, and H. Wang, "Real-time action recognition with deeply transferred motion vector cnns", *IEEE Transactions on Image Processing*, vol. 27, pp. 2326-2339, 2018.

[31] A. Ullah, J. Ahmad, K. Muhammad, M. Sajjad, and S. W. Baik, "Action recognition in video sequences using deep bi-directional LSTM with CNN features", *IEEE Access*, vol. 6, pp. 1155-1166, 2017.

[32] M. Usman Khalid and J. Yu, "Multi-Modal Three-Stream Network for Action Recognition", *arXiv*, p. arXiv: 1909.03466, 2019.

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770-778.

[34] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks", in *Advances in Neural Information Processing Systems*, 2012, pp. 1097-1105.

[35] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks", *IEEE Transactions on Signal Processing*, vol. 45, pp. 2673-2681, 1997.

[36] V. Subramanian, *Deep Learning with PyTorch: A practical approach to building neural network models using PyTorch*, Packt Publishing Ltd, 2018.

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

[38] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1933-1941.

[39] N. T. Vu, P. Gupta, H. Adel, and H. Schütze, "Bi-directional recurrent neural network with ranking loss for spoken language understanding", in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 6060-6064.

[40] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, *et al.*, "Going deeper with convolutions", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1-9.

[41] L. Sun, K. Jia, D.-Y. Yeung, and B. E. Shi, "Human action recognition using factorized spatio-temporal convolutional networks", in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4597-4605.

[42] B. Mahasseni and S. Todorovic, "Regularizing long short term memory with 3D human-skeleton sequences for action recognition", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3054-3062.

[43] H. Bilen, B. Fernando, E. Gavves, A. Vedaldi, and S. Gould, "Dynamic image networks for action recognition", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3034-3042.

Akram Mihanpour received her bachelor's degree in software engineering from Shahid Chamran University of Ahvaz (SCU), Iran, in 2015, with research in cloud computing. She received her master's degree in artificial intelligence from the Shahid Chamran University of Ahvaz in 2019. Her research interests include deep learning, image and video processing, optimization algorithms, features extraction, pattern recognition, machine learning, data mining, and computer vision.

Mohammad Javad Rashti is an assistant professor of computer engineering at Shahid Chamran University of Ahvaz (SCU), Iran. He received his BSc, MSc, and Ph.D. degrees from the University of Tehran, Sharif University of Technology, and Queen's University at Kingston, respectively. He has conducted his research in the area of high-performance computing and networking, in collaboration with leading companies, universities, and national labs in Canada, USA , and

Iran, publishing several scholarly papers in these areas. He is the founder of Innovation and Creativity Center and the Director of IT services at SCU.

Seyed Enayatallah Alavi is an assistant professor at the Department of Computer Engineering, Shahid Chamran University of Ahvaz (SCU), Iran. He received his B.Sc. degree from the Isfahan University of Technology, Isfahan, Iran, in computer engineering in 1992 and his M.Sc. degree in computer engineering-machine intelligence and robotics in 1996 from Shiraz University, Shiraz, Iran. In 2011, he received his Ph.D. degree in Computer Engineering, major in Artificial intelligence, from Belarusian National Technical University, Minsk, Belarus. He has over 17 years of academic experience and has published more than 60 papers in international and national conferences and more than 20 papers in international and national journals, in addition to 5 books. His current research interests are deep learning and evolutionary processing.