Multi-Temporal-Resolution Technique for Action Recognition using C3D: Experimental Study

Bassel S. Chawky
Scientific Computing department,
Computer and Information Science,
Ain Shams University
Cairo, Egypt
bassel.safwat@cis.asu.edu.eg

Mohammed Marey
Scientific Computing department,
Computer and Information Science,
Ain Shams University
Cairo, Egypt
mohammedmarey11@gmail.com

Howida A. Shedeed
Scientific Computing department,
Computer and Information Science,
Ain Shams University
Cairo, Egypt
dr_howida@cis.asu.edu.eg

***Abstract*—In any given video containing an action, the motion conveys information complementary to the individual frames. This motion varies in speed for similar actions. Therefore, it is a promising approach to train a separate deep-learning model for different versions of action speeds. In this paper, two novel ideas are explored: single-temporal-resolution single-model (STR-SM) and multi-temporal-resolution multi-model (MTR-MM). The STR-SM model is trained on one specific temporal resolution of the action dataset. This allows the model to accept a longer temporal frame range as input and therefore, a faster action classification. On the other hand, the MTR-MM is a set of STR-SM models, each trained on a different temporal resolution with a late fusion using majority voting achieving more accurate action recognition. Both models have improvements over the traditional training approach, 3.63% and 6% video-wise accuracy respectively.**

Keywords—action recognition; deep learning; UCF101; spatio-temporal features; c3d; 3d conv net

I. Introduction

Action recognition is the task of classifying the action occurring in a video clip. The complexity of this task is due to several factors like environment factors, occlusions, illuminations change, variation in the viewpoints and performed action speed. This leads to changes in appearance and movement adding extra amount of challenges for the current algorithms.

Despite of these challenges, the number of studies for this problems has noticeably increased due to the importance of action recognition in many applications like long-length video summarization [1], video surveillance to detect suspicious actions for security reasons [2], robotics solutions for human behavior characterization [3] and many others. Moreover, it is important to develop and enhance video analysis techniques to extract meaningful information, i.e. keywords, content and/or actions, as the number of videos uploads over the internet is growing enormously which poses lots of challenges to process all this data in an acceptable time and computational power.

Convolutional neural network (CNN) [1] is, at the present time, a standard for all image-related tasks. This is due to its ability to appropriately represent the images content. It also has the potential to overcome many issues like occlusions, different illumination conditions and many others. Therefore, researchers involved CNNs in their studies [5][6][7] in different methods to tackle the task of recognizing action from videos. These methods rely on 3 main facts: (1) videos are composed of frames: the spatial dimension containing information about the locations and the main objects performing the action, (2) the successive video frames provide information about the motion: the temporal dimension. (3) The sequence of motions, i.e. the evolution of the video, provides global video information.

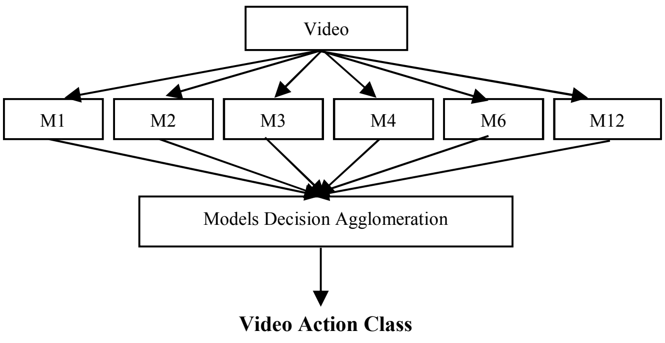


Figure 1. Representation for Multi Temporal Resolution Multi Model (MTR-MM)

Different models have been developed, each tackling one of the previously discussed facts. There are three main models; the first is a two streams CNN model [5] that separates the spatial and temporal information with a fusion at the last network layers. In this model, the spatial features are extracted using the convolution operation while temporal features explicitly involve the optical flow and use them to train the temporal CNN stream. The second model is the 3D convolutional Neural Network (C3D) [6] that merges the spatial and temporal features during the convolution steps by using mini 3D kernels (3x3x3). The 3D kernels prevent the loss of the temporal dimension along the layers instead of the traditional squashing in the normal convolutional models. Finally, the third model is the Long Short-Term memory (LSTM) [7] which works best with sequential data and is usually used to provide a global video information and therefore better classification.

In this paper we suggest a new methodology to train an action recognition model on a given video dataset. The proposed model architecture is illustrated in “Fig 1”. For each video in a given action dataset, the successive videos frames are interpolated using different steps yielding different resolution. We call this n-step based multi resolution where n is step size: the number of frames to drop. For each n-step based resolution, a different temporal resolution sub-dataset is generated. Then, several models are trained each on the corresponding sub-dataset. This methodology benefits to generate different speeds of the actions and allow representing the different action speeds, therefore, increasing the model ability to recognize the same action with different speeds. Another finding is that using a very high step (low temporal resolution) coarse motion information can be extracted and increases the model accuracy while having the ability to classify a larger portion of the input video, therefore, benefits reducing the time required. Therefore, a multi-resolution model has been also presented to use this complementary information in a fine to coarse motion model.

The paper is organized as follow: the next section provides a survey on different deep learning models used for action recognition. Section III highlight the scientific details of the C3D model implemented in this paper. Section IV presents the proposed solution. Section V lists the experiments results and provide a discussion on each experiment set. Finally, section VI provides the conclusions and further research directions.

II. RELATED WORK

The breakthroughs of Convolutional Neural Networks in the last decade proved large success in object recognition, object segmentation and different static image-based tasks. One of the reasons for this success, is the ability of convolution operator to capture spatial relationships in the image that are robust against transformations. In this section, numerous CNN-based techniques for action recognition are surveyed highlighting the usage of the temporal dimension as it provides missing crucial information in still image.

Different architectures have been suggested starting with AlexNet [10], GoogLeNet [11], VGG Net [12], and reaching ResNet [13] which becomes the default architecture for ConvNets in practice. In addition, many improvements in [9] allowed the CNNs to be robust against extra transformations like rotation, scaling and deformations.

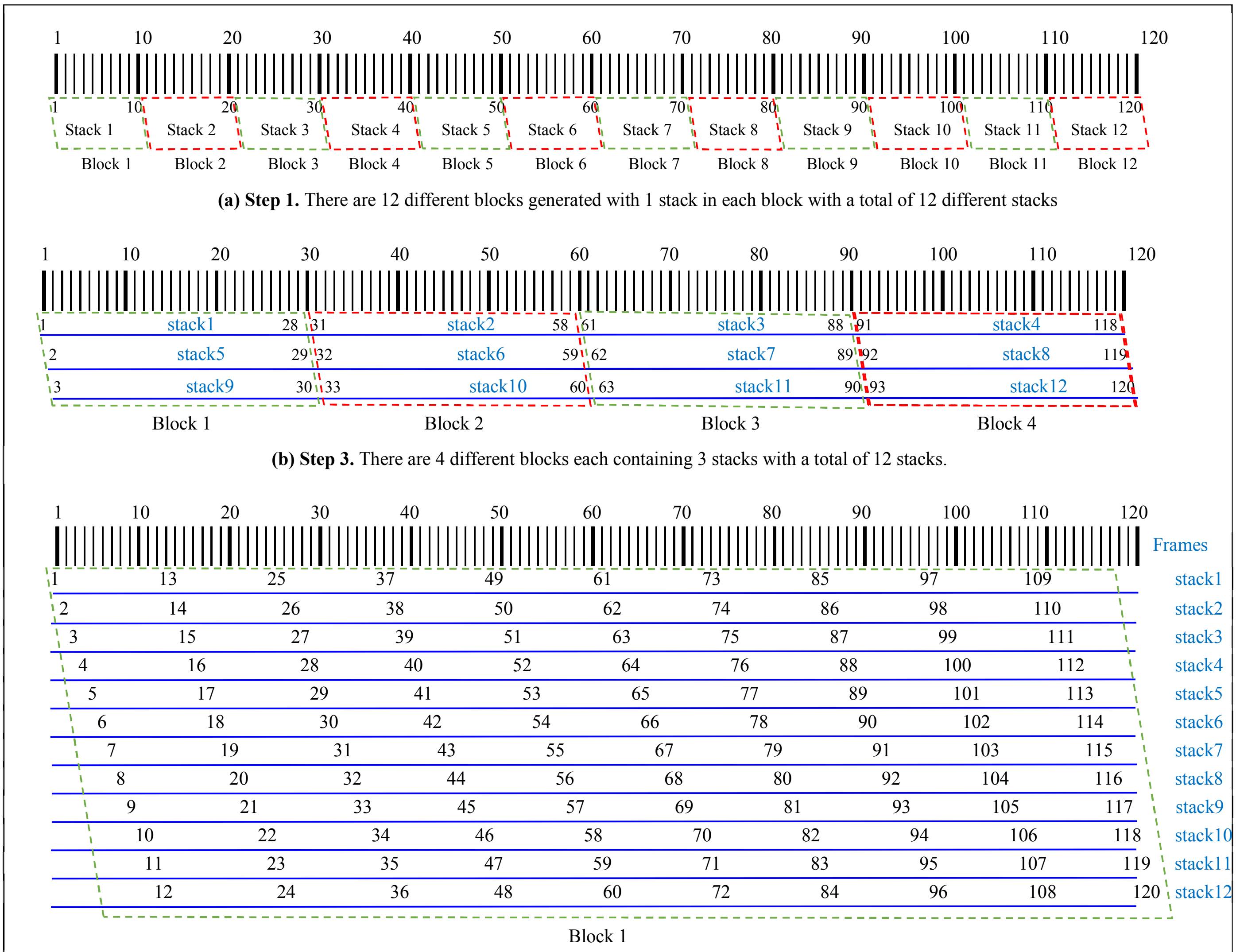
The reason of choosing CNNs for action recognition is because CNNs have a strong representation of the visual features, better than classical approaches. However, regarding action recognition in videos, applying CNN on individual video-frames is not enough for action recognition for a given video. Consequently, the main research direction is to allow the extraction of features that represents the temporal information as well, both the *local temporal features* between successive frames and the *global temporal features* along the whole video. With this in mind, two main approaches were developed. The first approach is to model separately the appearance and the motion information using two-streams of convolutional networks. In this model, the frames are fed to the appearance network while optical flows are calculated and fed to the temporal network of the model. The score of each network is combined at the end to predict the action label [5][14]. The second approach is to generate mixed spatial and temporal features called spatio-temporal features. In this network, the traditional 2d kernel of the CNN into a 3D kernel and use stacks of 16 frames to learn and generate the desired mixed spatio-temporal features. The two-stream convolutional network outperforms the simple C3D model as it explicitly uses the optical flow fields also dues to the needs of large training data to tune the C3D parameters.

More models are based on these two baseline models. For example, 3d convolutional kernels in the two-streams CNN model [15], this study combines both baseline models (two-streams CNN and C3D) with the benefit of using pretrained weights of the two-stream model for weight initialization. Another paper studies the usage of the two-streams CNN together with ResNet architecture [16] yielding a spatiotemporal ResNet architectures. It also adds extra residual connection from the motion to the appearance network allowing deep spatio-temporal features.

The action usually requires a large number of frames to be completely represented. However, the discussed approaches for action recognition require a fixed small stack of frames to predict the action class. Therefore, it is expected that the generated local temporal features from these stacks are not enough to represent the action. There are two solutions to handle this problem. The first is to apply a temporal pooling technique for the local features to provide a video representation, i.e. average the features from a specific layer or using a majority voting technique for selecting the action of the video. The second solution is to employ Recurrent Neural Network (RNNs), specifically Long Short-Term Memory (LSTM) by training on sequences of these local features and use the output of the last LSTM cell for a final classification for each video [17].

Different techniques for temporally pooling features are developed to cope with representing the whole video. The C3D uses a simple average of the fc6 layer of the model for all the volumes and achieved (85.6%) [6] with a weight initialization from Sports-1M dataset. [18] designed a temporal segment network that divides the input video into segments and then selects a snippet from each segment to predict using two stream convnet based on the Inception model. These predictions are combined using a consensus function for a video level prediction. [19] experimented a stack of 60 frames instead the default 16 frames for the C3D model and achieved an accuracy of 92.7%. However, their training is done using optical flow and RGB frames combined with Improved Dense Trajectory (iDT).

Instead of the temporal pooling step, which lose the relationships between each group of features, LSTM is employed to learn the sequential relationship. The input video is divided into several group of selected frames (stacks), and, for each of them a model is used to extract the local features. These local features are fed in sequence to an LSTM model that finds the relations between them and predict the action class. This is a better way to agglomerate the feature information without loss


1 10 20 30 40 50 60 70 80 90 100 110 120
1 10 Stack 1 | 20 Stack 2 | 30 Stack 3 | 40 Stack 4 | 50 Stack 5 | 60 Stack 6 | 70 Stack 7 | 80 Stack 8 | 90 Stack 9 | 100 Stack 10 | 110 Stack 11 | 120 Stack 12
Block 1 Block 2 Block 3 Block 4 Block 5 Block 6 Block 7 Block 8 Block 9 Block 10 Block 11 Block 12
1 10 20 30 40 50 60 70 80 90 100 110 120
1 stack1 28 | 31 stack2 58 | 61 stack3 88 | 91 stack4 118
2 stack5 29 | 32 stack6 59 | 62 stack7 89 | 92 stack8 119
3 stack9 30 | 33 stack10 60 | 63 stack11 90 | 93 stack12 120
Block 1 Block 2 Block 3 Block 4
1 10 20 30 40 50 60 70 80 90 100 110 120
Frames
1 13 25 37 49 61 73 85 97 109 stack1
2 14 26 38 50 62 74 86 98 110 stack2
3 15 27 39 51 63 75 87 99 111 stack3
4 16 28 40 52 64 76 88 100 112 stack4
5 17 29 41 53 65 77 89 101 113 stack5
6 18 30 42 54 66 78 90 102 114 stack6
7 19 31 43 55 67 79 91 103 115 stack7
8 20 32 44 56 68 80 92 104 116 stack8
9 21 33 45 57 69 81 93 105 117 stack9
10 22 34 46 58 70 82 94 106 118 stack10
11 23 35 47 59 71 83 95 107 119 stack11
12 24 36 48 60 72 84 96 108 120 stack12
Block 1

(a) Step 1. There are 12 different blocks generated with 1 stack in each block with a total of 12 different stacks

(b) Step 3. There are 4 different blocks each containing 3 stacks with a total of 12 stacks.

(c) Step 12. There are 1 block containing all the12 different stacks.

Figure 2. The extraction of 12 stacks from an input video of 120 frames.

of relations across them on the cost of further computations and memory. Moreover, it can deal with a variable number of frames while there is a fixed number of parameters. [17] compared the accuracy of a deep LSTM as feature aggregation technique with other several feature pooling methodologies using both optical flow and video frames as input and reaching the accuracy of 88.6% on UCF101 dataset. Similarly, [20] involved LSTM to build deep models in both spatial and temporal dimensions and instantiated it in 3 vision related tasks successfully: activity recognition, image captioning, and video description.

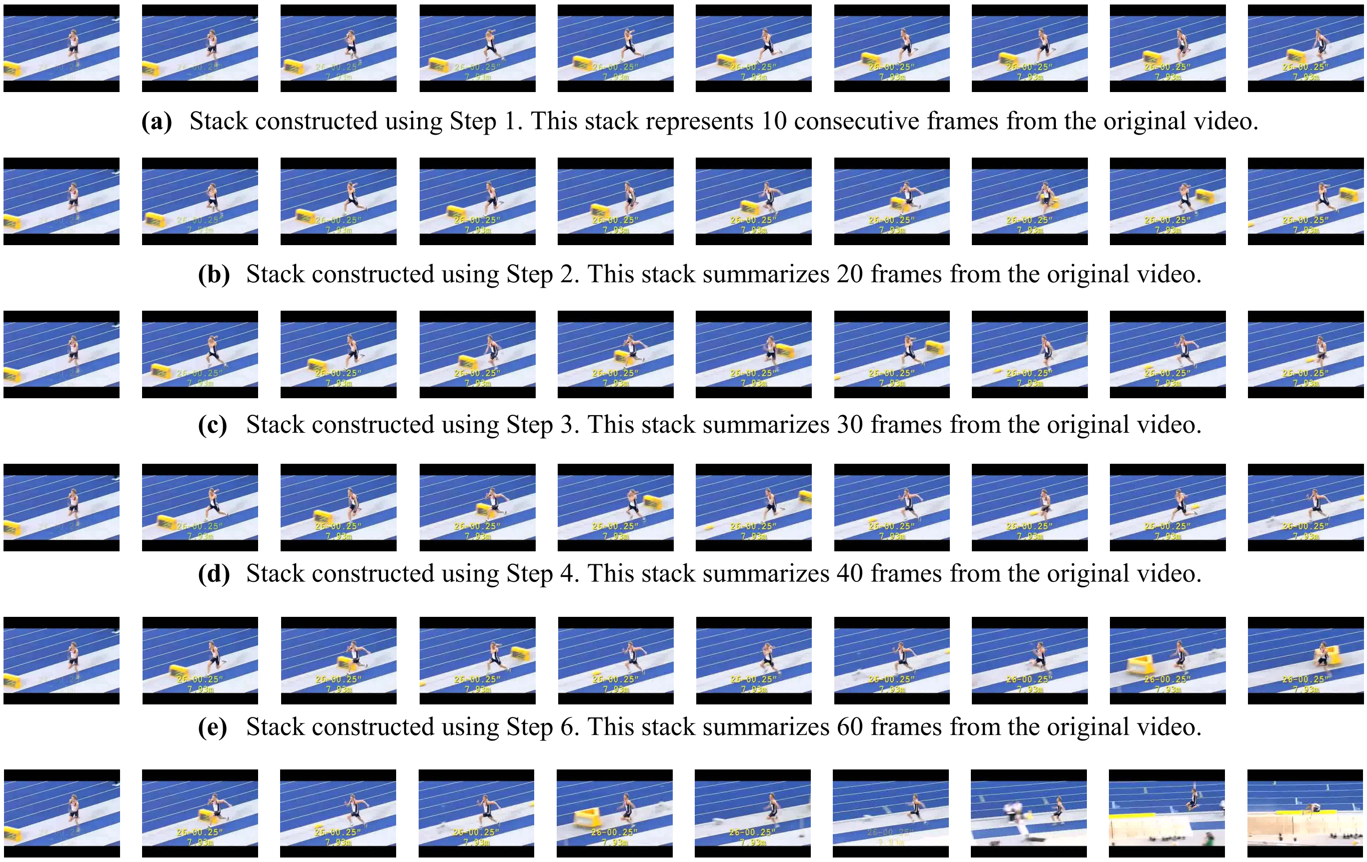
The discussed techniques require lot of computations and additional assist by hand-crafted features, large stack of frames or post processing to provide a better representation of the temporal dimension. In contrast, we maintain the simplicity by employing only the C3D model and allow it to cover longer action videos with different resolution. N-steps based resolution of the video are extracted and each stack is classified by the corresponding model resolution. After stack level classification, majority voting is performed to calculate the final video level action class. Moreover, our model is flexible to trade between speed and accuracy via the number of steps to set the desired resolution, and the number of stacks used for each resolution.

III. Learning with 3D ConvNet

In this section, the architecture of the 3d convnet is explained. Like the 2D kernel, the convolution is defined as a sum of element-wise products of 2 arrays. The exact mathematical formula for 3D kernels convolution is given by equation (1) [21]:

$$v_{ij}^{xyz} = \sum_{m} \sum_{p=0}^{P_i-1} \sum_{q=0}^{Q_i-1} \sum_{r=0}^{R_i-1} w_{ijm}^{pqr} \, v_{(i-1)m}^{(x+p)(y+q)(z+r)} \qquad (1)$$

where v_{ij}^{xyz} is the value at the position x, y, z for layer i and a feature map j. P, Q, and R are the width, height and depth



(a) Stack constructed using Step 1. This stack represents 10 consecutive frames from the original video.

(b) Stack constructed using Step 2. This stack summarizes 20 frames from the original video.

(c) Stack constructed using Step 3. This stack summarizes 30 frames from the original video.

(d) Stack constructed using Step 4. This stack summarizes 40 frames from the original video.

(e) Stack constructed using Step 6. This stack summarizes 60 frames from the original video.

(f) Stack constructed using Step 12. This stack summarizes 120 frames from the original video.

Figure 3. Six Different steps are illustrated for Long Jump video taken from UCF56. Step 1 shows fine motion details; however, it only shows portion of the action. On the opposite side, step 12 shows a coarser motion, which is the main motion of the object, and represents the evolution of the video. The features extracted from each stack are complementary to each other to better classify the action performed in a fine to coarse temporal representation of the action**.**

(temporal dimension) of the kernel, *m* is the index of the feature map in the previous layer *i-1*.

The advantages of the 2D convolution over the multiplication operator is that it prevents the loss of the spatial relationships. Similarly, the 3D convolution prevents the loss of the local temporal information existing between successive frames. In 2D convolutional kernel, the temporal dimension is squashed, meanwhile, it is preserved along the network layers. The C3D architecture used here is composed of 8 convolutional layers where the first 2 layers are followed by 1x2x2 and 2x2x2 pooling layers respectively, and for the 6 other convolutional layers, each 2 are followed by 2x2x2 pooling layer. The convolutional layers are followed by 2 fully connected layers with a SoftMax at the end to provide a representation of the probability of each class. The best kernels size is 3x3x3 based on the experiments in the original paper [6].

IV. Proposed Solution

Action recognition is composed of different temporal resolution. For lower step-based resolution, that is a high resolution, there exists fine motion details, especially for slow motion action. On contrast, for higher step-based resolution, that is a low resolution, longer portion of the video is summarized, thus providing coarse grained motion that represents the action flow or evolution.

In this paper we use the C3D model for action recognition problem and investigated the following questions: 1) Which is better for action recognition, higher or lower resolution datasets 2) Can specific frames act as key frames like the interpolation of a function 3) What is the suitable temporal resolution for action recognition 4) What is the best combination between trained-model resolution and testing-dataset resolution 5) Is it better to train a model on different resolution or have a separate model for each resolution 6) Are high and low resolutions complementary for the action recognition?

Setting the step to 1 will lead to obtain the traditional training approach, where the input stack is essential composed of *N* successive frames from the original video, with *N* being the stack size. On the other hand, 2, 3, …, *k* steps, means to construct the input stack for the C3D model such that the frames taken from the original video are *k* step separated, thus, controlling the desired temporal resolution.

Using these n-steps based resolution dataset, three experiment-sets are performed, and their results are discussed in the following section. For all the experiments, the number of epochs of the training is fixed to 5 epochs. The first experiment set inspects the STR-SM approach, where one model is trained for each temporal resolution sub-dataset. Experiments compare the different resolutions and reports the best combination of training resolution vs testing resolution. The second and third experiments sets evaluate the STR-SM and MTR-SM to compare the multi-model vs the single-model approaches when fed with all the different resolution dataset.

Each video usually contains more than one stack of frames. The stack size used in this paper is 10 frames. Then, these stacks are classified each by the C3D model and the final video classification is based on the majority votes of the individual stacks. As an example, a video of 120 frames provides 12 stacks each of size 10 frames. Each of these frames is fed to the model to calculate the probability with all the classes. This stack is classified based on the maximum probability among the 101 probabilities of the model output. Among the different classes of the different stacks, the class having the most votes represents the video class.

The original study [6] model used a stack of 16 frames. However, in this study, the stack size is set to 10 frames. While this number is temporally shorter than the stack size of the original paper, and may suffers missing more information about the action, the frames are selected with different steps. Therefore, for higher steps, it allows the stack of frames to cover a longer temporal range from the original video compared to the traditional approach, i.e. 120 frames (step 12 and stack size 10) vs 16 frames. This paper experiments 6 different temporal step-based resolutions: 1 (original video resolution), 2, 3, 4, 6 and 12 steps. Having these different resolutions is necessary to experiment different temporal resolutions and to validate and verify the multiple resolutions technique. Moreover, it benefits an increase in the batch size as it reduces the memory required for each stack.

	Training step 1	**Training step 2**	**Training step 3**	**Training step 4**	**Training step 6**	**Training step 12**
Testing s1	**32.41%**	32.24%	34.88%	29.09%	26.89%	23.08%
Testing s2	31.58%	33.38%	35.47%	30.43%	29.29%	24.86%
Testing s3	30.64%	**33.51**%	35.45%	30.93%	30.61%	26.54%
Testing s4	30.41%	33.07%	**35.55%**	31.20%	31.02%	27.59%
Testing s6	29.11%	32.59%	35.12%	31.37%	31.51%	29.22%
Testing s12	27.81%	31.94%	33.73%	**31.70%**	**32.22%**	**30.21%**
Min Acc.	27.81%	31.94%	**33.73%**	29.09%	26.89%	23.08%

Table 1. Stack-wise accuracy for testing STR-SM. Each model is trained on single resolution with the resolution-step stated in the first row and is tested against different step dataset, with the step stated in the first column.

In "Fig 2", three n-step-based resolution are illustrated: Step 1 (traditional approach), 3 (best accuracy) and 12 (fastest method). A low resolution allows covering a longer temporal dimension, which is very beneficial if the video is recorded using a high FPS, slow occurring actions or temporally long action video. That is a step of 12 can represent an action of up to 120 frames, using a stack size of 10, in only 1 stack of frames. The different steps generate a fine to coarse action motion as illustrated in "Fig 3".

In the training phase, all the 12 stacks are generated from each video. The details of stacks generation are explained in the *data preparation* subsection (V.A). However, for the STR-SM, during the inference, 1 stack can be used, thus, ultimately, reaching 12 times faster than the traditional method.

V. Experiments Results & Discussion

An n-step based temporal resolution dataset is required to train and evaluate the different temporal-resolution models. The first subsection provides the details of the dataset used, while the results for each of the STR-SM, MTR-SM, MTR-MM are discussed in subsections V.B, V.C and V.D respectively.

A. Dataset preparation – UCF56 dataset

It is important to evaluate the dataset on challenging videos to provide a realistic measurement. Moreover, as a constraint, the number of frames per video is set to 120 to allow achieving 12 different resolutions using a stack size of 10 frames. UCF101 dataset is selected as it contains challenging and realistic videos [22].

Originally, UCF-101 contains 101 action class, 9537 training data and 3783 testing videos. These videos are selected specifically to provide realistic videos with cluttered backgrounds, different perspective for each class (intra-class variation), different illuminations and camera motion. Each video contains exactly one single action spatially and temporally. Moreover, videos are grouped into 25 groups where each group shares similar properties like same background, objects, similar viewpoints, etc and test videos are selected from different groups than the training videos.

The UCF-101 dataset has variable number of frames per video and many of them are less than 120, thus, this dataset needs more tweaking to allow testing our proposed approach. UCF-56 dataset is then constructed based on UCF-101: the video of UCF-101 are filtered such that the number of frames in each video is 120 or more. In addition, another constraint is set to guarantee a fair training; the number of videos per class must above 72 videos for the training dataset after constraint 1 filtration. This resulted in having 56 classes satisfying these 2 conditions out of 101. The 56 class are listed in the GitHub repository[1]. The total number of training videos used is 5080 and the total number of testing videos is 1953 which is also

[1] https://github.com/b-safwat/multi-temporal-resolution-c3d

	Training step 1	Training step 2	Training step 3	Training step 4	Training step 6	Training step 12
Testing s1	**34.52%**	35.19%	37.74%	31.61%	29.71%	24.85%
Testing s2	33.70%	35.54%	**38.15%**	32.37%	31.30%	26.59%
Testing s3	31.91%	**35.80%**	37.23%	32.73%	**32.42%**	28.18%
Testing s4	31.40%	34.42%	36.82%	**33.19%**	32.22%	28.90%
Testing s6	30.38%	33.14%	35.80%	32.17%	32.17%	29.71%
Testing s12	27.97%	32.07%	33.91%	31.96%	32.37%	**30.28%**
Min Acc.	27.97%	32.07%	**33.91%**	31.61%	29.71%	24.85%

Table 2. Video-wise accuracy for testing STR-SM. Each model is trained on single resolution with the resolution-step stated in the first row and is tested against different step dataset, with the step stated in the first column

filtered from the original testing set with the same constraints of having at least 120 frames per video for the 56 classes.

Each video is represented by its first 120 frames to be used for extracting different resolutions. This guarantees that for all the steps, no more than 12 stacks per video can be generated. Each stack of 120 frames is divided into *b* blocks as given in equation (2):

$$b = 120/(k*N), \quad k \epsilon \{1, 2, 3, 4, 6, 12\} \quad (2)$$

where *k* is the desired step and *N* is the stack size, which is fixed to 10 in this paper. For example, for a step of 3, there are *120/(3*10)* → 4 blocks. For each of these blocks and for each step *k*, the selection of the frames follows equation (3):

$$i_s = i_o + i*k, \quad i \epsilon \{1, 2, ..., N\text{-}1, N\} \quad (3)$$

where i_s is the index of the selected frame in the original video, i_o the start frame index of the current stack in the original video, and *i* is the frame index in the current stack. This process is repeated *b* times for each block. An illustration of the frames selection is shown in "Fig. 2" for steps 1, 3 and 12.

B. Single Temporal Resolution – Single Model (STR-SM)

In this experiment-set, 6 different resolution of steps: 1, 2, 3, 4, 6 and 12 are experimented. Training datasets are used to train a separate single model on the different resolutions which is then tested on all the different resolutions test-datasets. The results listed in table 1 show that at a resolution of step 3, the best accuracy is achieved, which is better than the traditional training on sequential data by 3.14%. Moreover, the max-min accuracy across each column is at the resolution of 3 indicating the best model to use when dealing with multiple speed actions.

	Test-set step 1	Test-set step 2	Test-set step 3	Test-set step 4	Test-set step 6	Test-set step 12
Training resolution: 1, 2, 3, 4, 6, 12	32.27%	32.62%	33.01%	32.65%	33.25%	**33.43%**

Table 3. Results of testing a model with multiple different resolutions datasets extracted from the UCF-56 dataset. Results are stack wise accuracy

Comparing the traditional training of step-1 with the step-2 training, using faster, e.g. double the speed, action dataset indeed improves the action recognition: testing step-12 has the accuracy of *31.94%* using the model trained on step-2 compared to *27.81%* for model trained on step-1 on the same test dataset.

The accuracies in table 2 are for stack level classification, however, to provide a video wise classification accuracy, for each of the 12 stacks, a majority voting is used. Results are summarized in Table 2 and are consistent with table 1 that the best resolution is achieved for the model trained on step 3. In addition to the accuracy benefit, training at lower resolutions benefits longer temporal range, that is resolution of step 3 using a stack size of 10 results in summarizing a temporal range of 30 frames from the original video. This stack contains triple the temporal information that exists in a stack size of 10, by only considering specific key frames. For a faster action, in UCF-56 case the resolution 4 and higher, this method fails as accuracy decrease. Our intuition is that the key frames used are not sufficient to interpolate the fine action motion. Therefore, 2 different strategies to develop a multi temporal-resolution model are presented in the next subsections.

C. Multi Temporal Resolution - Single Model (MTR-SM)

Instead of training a different model for each different resolution, in this experiment the model is trained on all the different resolutions available for each video. Results are listed in table 3, showing deteriorated accuracy. The reason may be the low ability of the model to cover all the extra intra-class variations and/or the need for further training time.

D. Multi Temporal Resolution - Multi Models (MTR-MM)

Slow motion actions would benefit when training at higher resolutions as this eliminates similar fine motion information. However, some actions can be performed in different speeds by different persons. Moreover, other actions like jumping or running contains fast motion (unless recorded with high fps) and requires a model trained with low temporal resolution, so dropping these frames is not beneficial.

The fact that there are different temporal resolutions for each action suggests using a set of models each trained on different temporal resolution. Moreover, this method combines the fine-to-coarse information of the action, each learned in a separate model. The video is therefore divided among the 6 different resolutions, 12 stacks for each. After classification, the results are agglomerated using a majority voting technique. This results in **40.51%** which is better than the majority voting of the traditional training method by about **6%**.

VI. Conclusion and Future Research directions

In this work two novel techniques are proposed for action recognition, increasing the accuracy compared to the traditional method. The *STR-SM* achieved its best accuracy using a 3-step based resolution with an enhancement of 3.63%. In addition to the accuracy increase, it allows to speed the action recognition accuracy 3 times. Our best accuracy is achieved using the ***MTR-MM*** with a total increase in the accuracy of about 6%. Results also state that different speeds for the actions are harmful for the C3D model learning. Our experiments are performed using a dataset of 56 classes that is extracted from UCF101 dataset.

Two new perspectives are introduced to the Action Recognition research field. The first is that the ability to recognize the action in a *summarized version* of the video saves time and increase the accuracy. The second is that by *training on multi speed of the action*, the model increase its ability to recognize unseen action speed.

This technique is simple and can be further enhanced by involving LSTM model. The corresponding sequential stacks generated from each block can be fed to LSTM model and the last LSTM classification are the final video classification. This direction keeps the model end to end trainable with the benefit of using the relationship between consecutive longer-temporal stacks.

Moreover, it worth investigate to fine tuning models that already have been trained on larger dataset, i.e. sports1M dataset, and compare the multi-temporal-resolutions multi model with the state-of-the-art models.

REFERENCES

[1] Potapov D, Douze M, Harchaoui Z, Schmid C. Category-specific video summarization. InEuropean conference on computer vision 2014 Sep 6 (pp. 540-555). Springer, Cham.

[2] Nievas EB, Suarez OD, García GB, Sukthankar R. Violence detection in video using computer vision techniques. InInternational conference on Computer analysis of images and patterns 2011 Aug 29 (pp. 332-339). Springer, Berlin, Heidelberg.

[3] Rezazadegan F, Shirazi S, Upcroft B, Milford M. Action recognition: From static datasets to moving robots. arXiv preprint arXiv:1701.04925. 2017 Jan 18.

[4] Gu J, Wang Z, Kuen J, Ma L, Shahroudy A, Shuai B, Liu T, Wang X, Wang G, Cai J, Chen T. Recent advances in convolutional neural networks. Pattern Recognition. 2018 May 1;77:354-77.

[5] Simonyan K, Zisserman A. Two-stream convolutional networks for action recognition in videos. InAdvances in neural information processing systems 2014 (pp. 568-576).

[6] Tran D, Bourdev L, Fergus R, Torresani L, Paluri M. Learning spatiotemporal features with 3d convolutional networks. InProceedings of the IEEE international conference on computer vision 2015 (pp. 4489-4497).

[7] Hochreiter S, Schmidhuber J. Long short-term memory. Neural computation. 1997 Nov 15;9(8):1735-80.

[8] Srivastava N, Mansimov E, Salakhudinov R. Unsupervised learning of video representations using lstms. InInternational conference on machine learning 2015 Jun 1 (pp. 843-852).

[9] Jaderberg M, Simonyan K, Zisserman A. Spatial transformer networks. InAdvances in neural information processing systems 2015 (pp. 2017-2025).

[10] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. InAdvances in neural information processing systems 2012 (pp. 1097-1105).

[11] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A. Going deeper with convolutions. InProceedings of the IEEE conference on computer vision and pattern recognition 2015 (pp. 1-9).

[12] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556. 2014 Sep 4.

[13] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. InProceedings of the IEEE conference on computer vision and pattern recognition 2016 (pp. 770-778).

[14] Feichtenhofer C, Pinz A, Zisserman A. Convolutional two-stream network fusion for video action recognition. InProceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2016 (pp. 1933-1941).

[15] Wang X, Gao L, Wang P, Sun X, Liu X. Two-stream 3d convnet fusion for action recognition in videos with arbitrary size and length. IEEE Transactions on Multimedia. 2017.

[16] Feichtenhofer C, Pinz A, Wildes R. Spatiotemporal residual networks for video action recognition. InAdvances in neural information processing systems 2016 (pp. 3468-3476).

[17] Yue-Hei Ng J, Hausknecht M, Vijayanarasimhan S, Vinyals O, Monga R, Toderici G. Beyond short snippets: Deep networks for video classification. InProceedings of the IEEE conference on computer vision and pattern recognition 2015 (pp. 4694-4702).

[18] Wang L, Xiong Y, Wang Z, Qiao Y, Lin D, Tang X, Van Gool L. Temporal segment networks: Towards good practices for deep action recognition. InEuropean Conference on Computer Vision 2016 Oct 8 (pp. 20-36). Springer, Cham.

[19] Varol G, Laptev I, Schmid C. Long-term temporal convolutions for action recognition. IEEE transactions on pattern analysis and machine intelligence. 2018 Jun 1;40(6):1510-7.

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

[21] Ji S, Xu W, Yang M, Yu K, inventors; NEC Laboratories America Inc, assignee. 3D convolutional neural networks for automatic human action recognition. United States patent US 8,345,984. 2013 Jan 1.

[22] Chawky BS, Elons AS, Ali A, Shedeed HA. A Study of Action Recognition Problems: Dataset and Architectures Perspectives. InAdvances in Soft Computing and Machine Learning in Image Processing 2018 (pp. 409-442). Springer, Cham.