Real-time Big Data Analytics for Multimedia Transmission and Storage

Kun Wang*, Jun Mi*, Chenhan Xu*, Lei Shu†, and Der-Jiunn Deng‡
*School of Internet of Things, Nanjing University of Posts and Telecommunications, Nanjing, China.
†Guangdong University of Petrochemical Technology, Guangdong, China.
‡Department of Computer Science and Information Engineering,
National Changhua University of Education, Changhua City, Taiwan.
Email: *kwang@njupt.edu.cn, *junmiia@163.com, *xchank@outlook.com,
†lei.shu@ieee.org, ‡djdeng@cc.ncue.edu.tw

***Abstract*—With increase demand on wireless services, equipment supporting multimedia applications has been becoming more and more popular in recent years. With billions of devices involved in mobile Internet, data volume is undergoing an extremely rapid growth. Therefore, data processing and network overload have become two urgent problems. To address these problems, extensive study has been published on image analysis using deep learning, but only a few works have exploited this approach for video analysis. In this paper, a hybrid-stream big data analytics model is proposed to perform big data video analysis. This model contains four procedures, i.e., data preprocessing, data classification, data recognition and data load reduction. Specifically, an innovative multi-dimensional Convolution Neural Network (CNN) is proposed to obtain the importance of each video frame. Thus, those unimportant frames can be dropped by a reliable decision-making algorithm. Then, a reliable key frame extraction mechanism will recognize the importance of each frame or clip and then decide whether to abandon it automatically by a series of correlation operations. Simulation results illustrate that the size of the processed video has been effectively reduced. The simulation also shows that proposed model performs steadily and is robust enough to keep up with the big data crush in multimedia era.**

***Index Terms*—Big Data, Multimedia, Real-Time, Load Reduction, Networking, Convolutional Neural Networks**

I. INTRODUCTION

People today are living in the era of multimedia data where data are unprecedentedly increasing and mobile devices are becoming mainstream. In the era, Internet traffic are experiencing an exponential growth in volume as well as in heterogeneity [1]. With the growth trends, on one hand, we enjoy the convenience of multimedia digital resources, on the other hand, many problems appear that we have to deal with simultaneously, e.g., multimedia transmission [2] and storage [3].

Generally, multimedia transmission and storage are a problem of network overload [4], and solving network overload problem usually has two strategies, one is dispersing blocked traffic in advance by optimizing route selection, and the other is recognizing abnormal traffic and abandoning it before transmission. Adaptive bidirectional optimisation (ABO) is a route selection strategy [5], it has a good performance on optimizing uplink and downlink performance, but has little improvement on data storage. When it comes to the other strategy, the primary task is how to correctly classify the videos. Convolutional Neural Networks (CNN), a typical feed-forward neural network, outperform in classifying 2D-shapes [6], and hybrid deep convolutional neural networks (HDNN) is a model which can extract variable-scale features and acquire improved speed and better precision of pattern recognition [7]. Compared with a large number of works concentrating on using deep learning methods to conduct image analysis, relatively a few works have exploited such approaches for analysing video [8]. A two-stream CNN structure was proposed to process video by dividing the original video information into spatial information and temporal information [9]. Then, an improved two-stream model aimed at dealing with video classification was proposed [10]. The model achieves competitive performance by training two CNNs, but the fusing multiple network model still needs to be improved when in a strong network where the traffic is very heavy.

In this paper, we address the problem of multimedia transmission and storage, especially for videos. In the process of transmission and storage, we consider abnormal traffics as unimportant frames and clips in the video streams. Specifically, different from conventional single input model, we inspire from a two-stream model which divides the input information into spatial and temporal information, and we set two inputs to separately deal with input videos' different information. One input deals with statics information such as scenes and objects, and the other deals with dynamic information such as motion information. We consider that the video stream is made up with numerous frames and clips. Then, we can analyse and monitor the abnormal traffic by recognizing these images, and thus solve the problem of multimedia transmission and storage. Based on these considering, we propose a hybrid-stream big data analytics model which contains a reliable key frame extraction mechanism and an improved CNN classification algorithm to enhance the classification precision and relieve the load in transmission and storage.

The contributions of our paper are summarized as follows:

- A hybrid-stream big data analytics model considered to solve multimedia transmission and storage problem is proposed. We set two types of input to improve the classification precision.

- An improved CNN classification algorithm which used to recognize video information is proposed. In the algorithm, we add a time dimension forming a 3-dimension matrix to classify video frames and clips.
- A key frame extraction mechanism is proposed. We aim to improve multimedia transmission and storage by adjusting suitable parameters.

The rest of the paper is organized as follows. In Section II, we give a briefly system overview and then introduce the classification module and the load-reduction module of the system. Section III provides simulation results to validate the performance of the proposed model. Finally, we draw the main conclusion in Section IV.

II. MODEL ANALYSIS

In this section, we first give a briefly system overview. Then, we introduce the classification module which used to label video frames' information value. Finally, we detail the load-reduction module which used to abandon less important frames.

A. Model Description

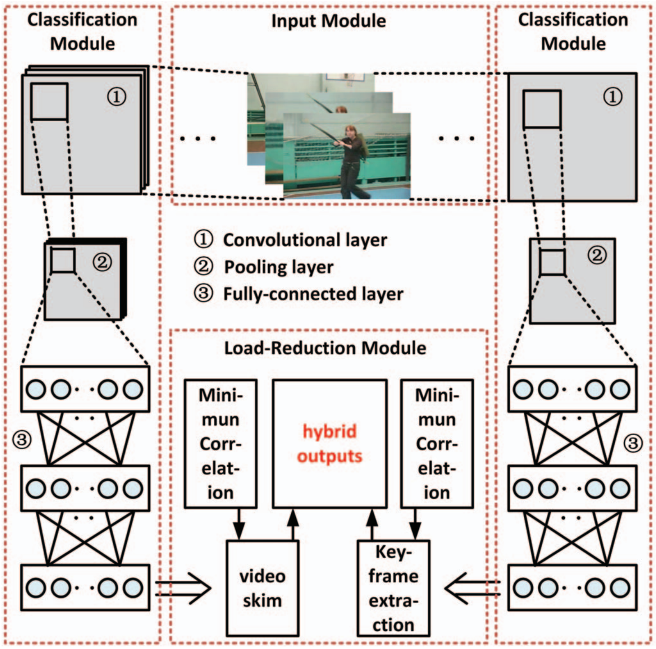

Fig. 1: Hybrid-stream big data analytics model

The primary problem of next-generation networks is the increasing data volume. To overcome the problem, we design a hybrid-stream big data analytics model as shown in Fig. 1. The functionality of hybrid-stream model can be summarized as three parts: Input Module, Classification Module and Load-Reduction Module. Input Module's main task is data preprocessing which deals with raw video resources. To improve the next module's classification precision, in input module, we divide raw video resources into two input forms, video frames and video clips. Video clips generally consist of several video frames. The part of input module transforms the raw video resources into another form which is suitable for data classification. Then, two different parts of classification module mainly further processes the two input video streams, and then outputs values which can represent every frame and clip's information. In the end of the module, we get two types of value which can be combined in load-reduction module to produce a new key feature parameter. Load-reduction module is the final processing step for deciding whether and how to drop a frame.

B. Classification Module

Perceptron is usually used to solve image classification problem. Before transmission, raw images are transformed into raw pixel data, and then transmission network correctly classifies these pixel data by dynamically learning its weights and biases. The dynamic learning process is a tough work, because only a small change happened on transmission network (such as weighs or biases of any single perceptron), the output of perceptron is tend to have a completely flip. Due to the change in the perceptron, the rest part of transmission network will completely change in some way. So it's not immediately obvious how we can get a transmission network of perceptron to learn video information. To overcome this problem, an improved method is proposed.

In this paper, we use a time dimensionality to connect several neighboring frames, so the form of output with time dimensionality can be written as:

$$f_j^{i,j,k} = \sum_{k} \sum_{j=1}^{J} \sum_{i=1}^{I} f_i * w_{i,j,k}. \tag{1}$$

Different from researchers using two dimensions to represent images, we add a time dimension k to represent videos. By doing so, the complexity of our proposed algorithm just add one more dimension calculation and the increased calculation for video analytics is acceptable. Next, we consider equation (1) as our 3-dimension convolution kernel to compute feature maps pixel by pixel.

Then, we introduce our algorithm in detail. First, we consider an image which is a $P \times Q$ greyscale image, then we have $P \times Q$ input neurons, with the intensities scaled appropriately between 0 and 1. We define the cost function as:

$$J(W, b) = \frac{1}{2n} \| h_{w,b}(x) - \alpha \|^2. \tag{2}$$

The inputs are 3-dimensionality matrixes, while the output dimension is determined by actual neural network conditions. The pseudo-code of the classification module is presented in **Algorithm 1**.

In the cost function of classification module, we achieve the goal of our algorithm to minimize the cost $J(w, b)$ as a function of the weights and biases. In other words, we try to find a set of weights and biases which makes the cost as small as possible ($J(w, b) = 0$).

Algorithm 1: Classification Module

```
Initialize $w$, $b$ randomly
While cost function $<$ threshold do
    Compute feature maps pixel by pixel according
     to the 3-dimension convolution kernel (1), and
     feed forward
    Calculate cost function (2)
    If cost function $>$ threshold then
        Back propagation using method Stochastic
         Gradient Descent (SGD)
    End if
End while
```

Theorem 1. The cost function J can be guaranteed always to decrease.

Proof: According to the basic conception of calculus, we can change J as the form of equation (2), and our algorithm is associated with vectors, so we denote the gradient vector by ∇J. The change of cost function is analyzed as follows:

$$\Delta J \approx \frac{\partial J}{\partial w}\Delta w + \frac{\partial J}{\partial b}\Delta b, \tag{3}$$

$$\nabla J = (\frac{\partial J}{\partial w}, \frac{\partial J}{\partial b})^T, \tag{4}$$

$$\Delta J = \nabla J \cdot (\triangle w, \triangle b), \tag{5}$$

$$(\triangle w, \triangle b) = -\eta \nabla J = -\eta (\frac{\partial J}{\partial w}, \frac{\partial J}{\partial b})^T, \tag{6}$$

$$\Delta J = -\eta \|\nabla J\|^2 = -\eta \|(\frac{\partial J}{\partial w}, \frac{\partial J}{\partial b})^T\|^2 \leq 0. \tag{7}$$

According to equations (5) and (6), we can find a suitable parameter which is called the leaning rate that can be used to simplify the expression (4), and then we can guarantee the J always decrease.

Theorem 2. A factor $\frac{1}{n}$ or $\frac{1}{m}$ in cost function helps to train data in real time.

Proof: Separating w and b, we get two equation (8) and (9). We set a small number m to randomly choose training inputs.

$$w'_k = w_k - \varrho \frac{\partial J}{\partial w_k}, \tag{8}$$

$$b'_k = b_k - \varrho \frac{\partial J}{\partial b_k}, \tag{9}$$

$$\frac{\sum_{j-1}^{m} \nabla J_{xi}}{m} \approx \frac{\sum_x \nabla J_x}{n} = \nabla J, \tag{10}$$

$$w'_k = w_k - \frac{\varrho}{m} \frac{\partial J}{\partial w_k}, \tag{11}$$

$$b'_k = b_k - \frac{\varrho}{m} \frac{\partial J}{\partial b_k}. \tag{12}$$

In Equation (2), we scale the overall cost function by a factor $\frac{1}{n}$. People sometimes omit $\frac{1}{n}$, and instead of averaging, people prefer to sum over the costs of individual training data. However, it is particularly helpful when we cannot know the number of training data in advance. The situation may happen if large number of training data generate in real time. Also, in a similar way, the mini-batch update rule (11) and (12) sometimes omit the $\frac{1}{m}$ term out the front of the sums. Conceptually this makes little difference, since it is equivalent to rescaling the learning rate ϱ.

C. Load-Reduction Module

1) Keyframes and Video Skim: After classification module labels every video frame and clip's information value, we consider these information values as load-reduction module's input. In this paper, we use the concept of the two video abstract mechanisms [11], and instead of using original video as input, we change the input to the information value output from classification module. We introduce keyframe and video skim in our paper as follow.

When dealing with video frames, we use key frames which can be also called representative frames, R-frames. R-frames are a set consisting of a collection of salient images extracted from the information values. Hence, the key frame set K is defined as follows:

$$K = F_{k-frames}(Video) = f_{r_1}, f_{r_2}, \cdots, f_{r_\sigma}. \tag{13}$$

where σ is the total number of keyframes and $F_{k-frames}$ denotes the concrete extraction operation. In equation (13), we use extractive label f_{r_i} to form a key frame set to represent the input video.

When dealing with video clips, we use video skim, to help to recognize the importance of each frame. Video skim consists of a collection of video clips extracted from the original video. These video clips are significantly shorter duration. The video skim S is defined as follows

$$S = F_{skim}(Video) = f_{r_1} \bigodot f_{r_2} \bigodot \cdots \bigodot f_{r_\sigma}. \tag{14}$$

where $\bigodot$ is the excerpt assembly and integration operation.

2) Key Frame Ratio: In important feature extraction mechanism, we set a ratio, θ, over the number of the video frames or video clips as a constraint to guarantee a suitable output number in order to guarantee storage. This method is very suitable for resource shortage environment.

The following problem is how to set the ratio. Actually, the problem can be viewed as an optimization problem of finding a suitable set $R = r_1, r_2..., r_\sigma$, which can represent the video using the least frames or clips. We summarize the optimization problem as:

$$r_1, r_2, \cdots, r_\sigma = arg\ min_{r_i}\{D(R, F) | 1 \leq r_i \leq n\}, \tag{15}$$

$$\sigma = \theta \cdot n. \tag{16}$$

where n is the number of frames or clips in the original video sequence, σ is the total number of key frames or clips, D is a dissimilarity measure and F is the output in classification module.

3) Minimum Correlation method: Next, we begin to analyse how to find a suitable set which can represent the video to use the least frames or clips. Because the inputs, frames or clips, are always sequential elements, we consider to make use of correlation among them to complete key elements extraction. Therefore, we introduce minimal correlation to solve the problem. The minimal correlation method is select frames or clips that are dissimilar to each other and can represent the video with the least elements. Introducing the concept of minimal correlation, we can rewrite the equation (15) as:

$$r_1, r_2, \cdots, r_\sigma = arg\ min_{r_i}\{Corr(f_{r_1}, f_{r_2}, \cdots, f_{r_\sigma})\}. \tag{17}$$

where $Corr$ is a correlation computing operation.

When we consider pairs of frames or clips to simplify the entire set, the equation (17) can be formulated as:

$$Corr(f_{r_1}, f_{r_2}, \cdots, f_{r_\sigma}) = \{\sum_{i=1}^{k-1} \sum_{j=i+1}^{k} corr(f_{r_i}, f_{r_j})^2\}^{1/2}. \tag{18}$$

where $Corr(f_{r_i}, f_{r_j})$ is the correlation coefficient of any two frames or clips (f_{r_i}, f_{r_j}).

In this paper, we take sequential elements into consideration, and the equation (18) can be written as:

$$\{r_1, r_2, \cdots, r_\sigma\} = arg\ min_{r_i}\{\sum_{i=1}^{\sigma-1} corr(f_{r_i}, f_{r_{i+1}})\}. \tag{19}$$

However, the extraction of key frames or clips based on these equations endeavor to maximize the difference of each frame or clip rather than simply reduce the total number.

```
Algorithm 2: Load-reduction Module — Step 1
Inputs $f_{r_i}$, $A$
Procedure:
 Begin
     While $(i < n)$
         If $(Corr(f_{r_i}) < A)$
             enter into scene+1
         Else
             still in scene
         End if
     End while
```

```
Algorithm 3: Load-reduction Module — Step 2
Input $f_{r_i}$
Procedure:
 Begin
     $\eta = \alpha \sum f/k$
     While (scene has not change)
         If $(f_{r_i} > \eta)$
             drop the frame
         Else
             store the frame in set S
         End if
     End while
```

In the model's load-reduction module, a reliable key frame extraction mechanism is applied to recognize the importance of each frame or clip. Specific implementation process of the mechanism is presented in **Algorithm 2 and 3**. Algorithm 2 mainly judges whether the scene changes or not and Algorithm 3 emphasizes the load-reduction. In the pseudo-codes, we first recognize the scene change, because different scenes have different thresholds, through correctly recognizing scene change or not, we can improve the final performance. What's more, if continuous frames are all in the same scene, we classify these frames into a group and distribute each group a threshold η.

$$\eta = \frac{\sum Importance_{frame}}{number}. \tag{20}$$

where $Importance_{frame}$ is processed frame's importance which ranges from 0 to 1, and $number$ refer to several correlative frames' number.

III. EXPERIMENTAL EVALUATIONS

In this section, the performance of hybrid-stream big data analytics model is verified. In the first part, we introduce the data set which is used in this model. Then the performance of hybrid-stream model is analyzed and compared to the existing related models. Finally, performance comparisons with Basic CNN, Temporal stream ConvNet and Two-stream model are demonstrated.

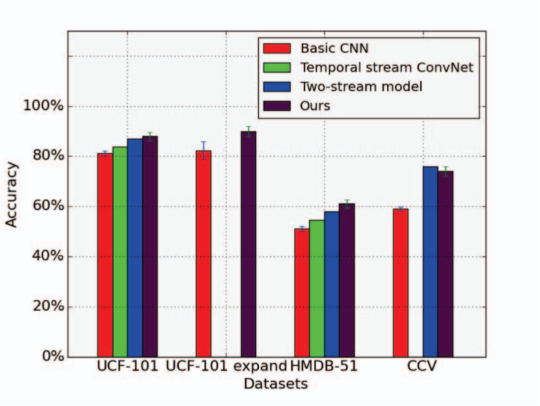

Fig. 2: Comparisons among different models under different data set.

A. Simulation Setup

The inputs are fixed to the size of 214×214. Different from two-dimension feature map in conventional layer, we add a time dimension based on the two-dimension feature map to process real-time videos. Therefore, we have two different inputs. The first inputs are video frames (single images), and the second inputs are video clips (several continuous frames).

In the following simulation, we change several parameters in classification algorithm to compare the algorithm's classification precision in different architectures. We also compare the model's performance in different data sets. Finally, we compare our model with other existing models.

B. Simulation Results

We analyse the model's performance in different data sets. UCF-101, UCF-101 Expand, HMDB-51 and CCV [12] are four different data set which we use to make comparisons. As Fig. 2 shows, our method has a good performance on the whole, and basically superior to other models on the same data set.

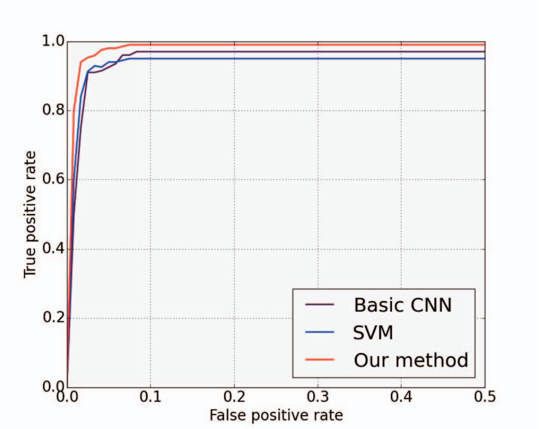

Fig. 3: Comparisons among basic CNN, SVM and our method under UCF-101.

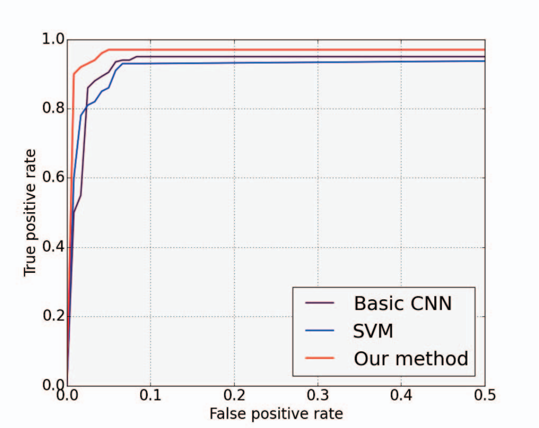

Fig. 4: Comparisons among basic CNN, SVM and our method under UCF-101 Expand.

Then, we compare our model with other existing models or algorithms and show our model's performance.

Receiver operating characteristic (ROC) is a kind of image which used to describe sensitivity. In the simulation, based on the data set UCF-101 and UCF-101 Expand, we use true positive rate (TPR) and false positive rate (FPR) of ROC to compare our method with basic CNN model and SVM (Support Vector Machine).

Fig. 3 and Fig. 4 show the classification performance of our methods, Basic CNN and SVM. As can be seen, for the same data set, our method has a better approach to the coordinate (0, 1) which is usually called as a perfect classifier. Meanwhile, the approaching speed of our method obviously faster than that of basic CNN and SVM. However, our method achieves an improvement on classification accuracy.

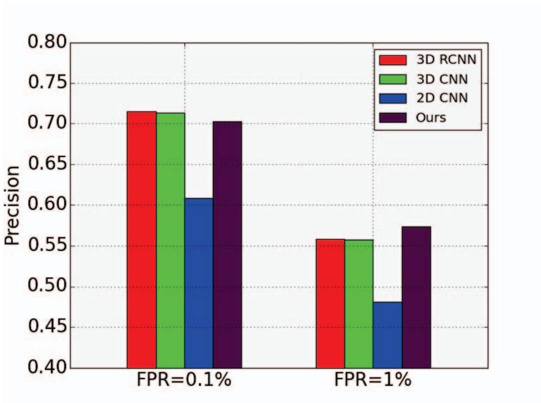

Fig. 5: Average performance (Precision) comparison of the four methods under different FPR.

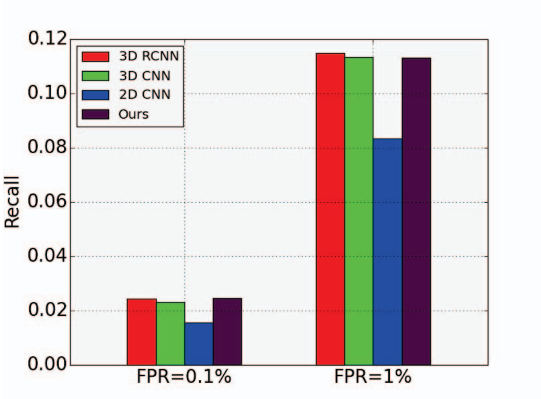

Fig. 6: Average performance (Recall) comparison of the four methods under different FPR.

The performance measures we used are precision, recall and scaled AUC (Area Under the Curve) at different values of false positive rates (FPR). The average performance is shown in Fig. 5, Fig. 6 and Fig. 7. Under lower FPR, we can see that higher average performance on Precision and lower average performance on Recall and AUC. When under the same FPR, our method outperforms the 2D-CNN, and has a

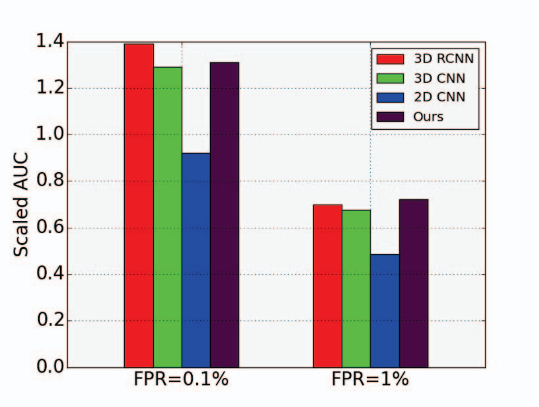

Fig. 7: Average performance (Scaled AUC) comparison of the four methods under different FPR.

similar performance with 3D Recursive-CNN (RCNN) and 3D CNN.

The performance of load-reduction module is shown in Fig. 8 and Fig. 9. The figures are both a 10-second video clip cutting from two different videos. The main difference between them is that the first video has an obvious scene change and the other only has a single scene. In Fig. 8, we use two colors to distinguish two scenes. Since we use different dynamic thresholds η to represent different scene, it is important to first recognize scene change with threshold η. From two figures, we can visually observe the drop frames.

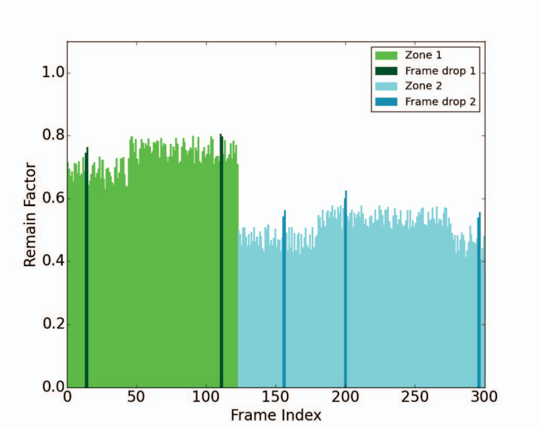

Fig. 8: Two scenes in a 10-second video and its drop frames.

IV. Conclusion

We present a novel model which enhances the performance of multimedia transmission and storage. The model, named hybrid-stream big data analytics model, is good at recognizing connections among frames and clips, and then do operation on them to improve transmission speed and reduce storage contents. Different from conventional deep learning methods to address image analysis problem, we improve the method to deal with video analysis. We formalize the video transmission and storage problem and shows a practical algorithm over a large-scale of real-time data. The conducted simulations show that our model performs well in most of the data sets, in particular for UCF-101 and UCF-101 Expand. Besides, proposed hybrid-stream big data analytics model and the improved video frames and clips recognized algorithm can lead to a fairly good video stream transmission and storage.

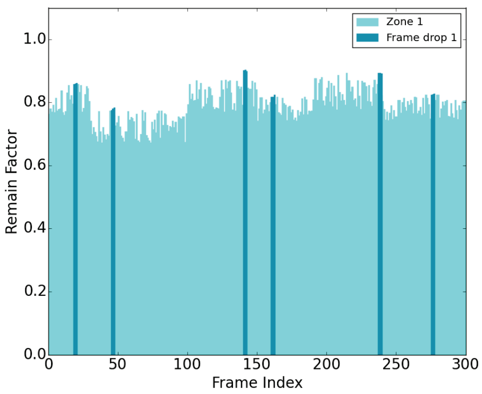

Fig. 9: One scene in a 10-second video and its drop frames.

References

[1] C. Chang, G. Huang, B. Lin, and C. Chuah, "LEISURE: Load-Balanced Network-Wide Traffic Measurement and Monitor Placement," *IEEE Transactions on Parallel and Distributed Systems,* vol. 26, no. 4, 2015, pp. 1059 - 1070.

[2] T. Jiang, H. Wang, and Y. Zhang, "Modeling Channel Allocation for Multimedia Transmission over Infrastructure based Cognitive Radio Networks," *IEEE Systems Journal,* vol. 5, no. 3, 2011, pp. 417 - 426.

[3] G. Nan, Z. Mao, M. Li, Y. Zhang, S. Gjessing, H. Wang, and Mohsen Guizani, "Distributed Resource Allocation in Cloud-based Wireless Multimedia Social Networks," *IEEE Network Magazine,* vol. 28, no. 4, 2014, pp. 74 - 80.

[4] X. Zhang, R. Yu, Y. Zhang, Y. Gao, M. Im, L. Cuthbert, and W. Wang, "Energy-Efficient Multimedia Transmissions through Base Station Cooperation over Heterogeneous Cellular Networks Exploiting User Behavior," *IEEE Wireless Communications,* vol. 21, no. 4, 2014, pp. 54 - 61.

[5] C. Sun, W. Wang, G. Cui, and X. Wang, "Service-aware bidirectional throughput optimisation route-selection strategy in long-term evolution-advanced networks," *IET Networks,* vol. 3, no. 4, 2014, pp. 259 - 266.

[6] S. Kiranyaz, T. Ince, and M. Gabbouj, "Real-Time Patient-Specific ECG Classification by 1-D Convolutional Neural Networks," *IEEE Transactions on Biomedical Engineering,* vol. 63, no. 3, 2016, pp. 664 - 675.

[7] X. Chen, S. Xiang, C. Liu, and C. Pan, "Vehicle Detection in Satellite Images by Hybrid Deep Convolutional Neural Networks," *IEEE Geoscience and Remote Sensing Letters,* vol. 11, no. 10, 2014, pp. 1797 - 1801.

[8] C. Yan, F. Coenen, and B. Zhang, "Driving posture recognition by convolutional neural networks," *IET Computer Vision,* vol. 10, no. 2, 2016, pp. 103 - 114.

[9] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," *in Proceedings of the Conference on Neural Information Processing System (NIPS),* 2014, pp. 568 - 576.

[10] H. Ye, Z. Wu, and R. Zhao, "Evaluating Two-Stream CNN for Video Classification," *in Proceedings of the 5th ACM on International Conference on Multimedia Retrieval (ICMR),* 2015, pp. 435 - 442.

[11] B. Truong and S. Venkatesh, "Video abstraction: a systematic review and classification," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM),* vol. 3, no. 3, Article 3 (February 2007).

[12] Y. Jiang, G. Ye, S. Chang, D. Ellis, and A. C. Loui, "Consumer Video Understanding: A Benchmark Database and an Evaluation of Human and Machine Performance," *in Proceedings of ACM International Conference on Multimedia Retrieval (ICMR),* 2011, pp. 29:1 - 29:8.