Video Summarization using LSTMs

Kunal Lad
University of Texas at Austin
2317 Speedway, Austin, TX 78712
kunal.lad@utexas.edu

Manu Agarwal
University of Texas at Austin
2317 Speedway, Austin, TX 78712
manuagarwal@utexas.edu

Abstract

In this paper, we present a Deep Learning framework to generate a summary of the most vital parts of the video. We use a combination of LSTM (Long Short Term Memory) modules and CNN (Convolutional Neural Network) to accomplish the task. The approach is motivated by the success of Sequential Determinantal Point Process(DPP) for supervised video summarization. It is also inspired by sequence to the sequence video to text(s2vt) approach where a sequential modeling of video frames leads to significantly better results.

1. Introduction

With the advent of the internet, a lot of videos are being created, shared and consumed every day. However, users are faced with the problem of which videos to watch given the limited amount of time they have. Generating informative captions or text summarization of videos does solve the problem to some extent but does not take into account how appealing a video is. For instance, two videos of the same task - one very appealing and the other one not so appealing might be labeled with the same caption or summary. This motivates us to think of some kind of video summarization or an appealing GIF creation.

Video summarization can also be helpful in generating highlights of sports matches or synopses of TV serials. In this project, given a video as input, we propose to output a short video or a GIF summarizing the whole video. In essence, we wish to keep only the interesting frames or those frames whose uniqueness score is high. This shall give us a set of still images from the video or continuous scenes. We use TVSum50 dataset that contains videos along with importance scores per frame given by Mechanical Turkers specifying how important a frame is for the task described by the video. We try to maximize the cumulative score of all the frames in the video summary our system creates.

Change-point detection has been recently employed by many [11] for this task. We use LSTMs to take into account the temporal information contained in videos. We wish to compare our approach to the performance of change-point detection on this task. We shall also show the performance of Gaussian and random sampling of frames from videos on this task and how much of an improvement our approach gives.

2. Related Work

Video Summarization has been an important area of Computer Vision research and a lot of work has been done in this field. Until recently most of this was focused on generating summaries using key frame detection based on some selection criteria. For eg [15] uses color, texture and shape based features whereas [14] uses optical flow to compute key-frames where the change in the optical flow is the highest.

Another technique for generating summaries is to identify key segments rather than key frames. Eg [9] uses kernel temporal segmentation to generate a summary consisting of multiple video segments. [3] presents a generic motion based video segmentation technique that takes into account cinematographic rules and hence works for all types of videos (static, egocentric or moving) These technique generate more visually aesthethic summaries than key frame based approach which lacks motion information.

A number of approaches [12, 1, 7] have also been developed for specific categories of video like sports videos, wearable cameras etc. [12] uses unsupervised deep learning to summarize sports videos. In [5, 1] authors propose a method which uses region cues indicative of high level saliency and temporal event detection for egocentric video summarization. [7] generates a k subshot summary by using a metric which captures how visual objects contribute to progression of events.

User generated videos are generally poor in quality and most of the above techniques dont generalize well to them.

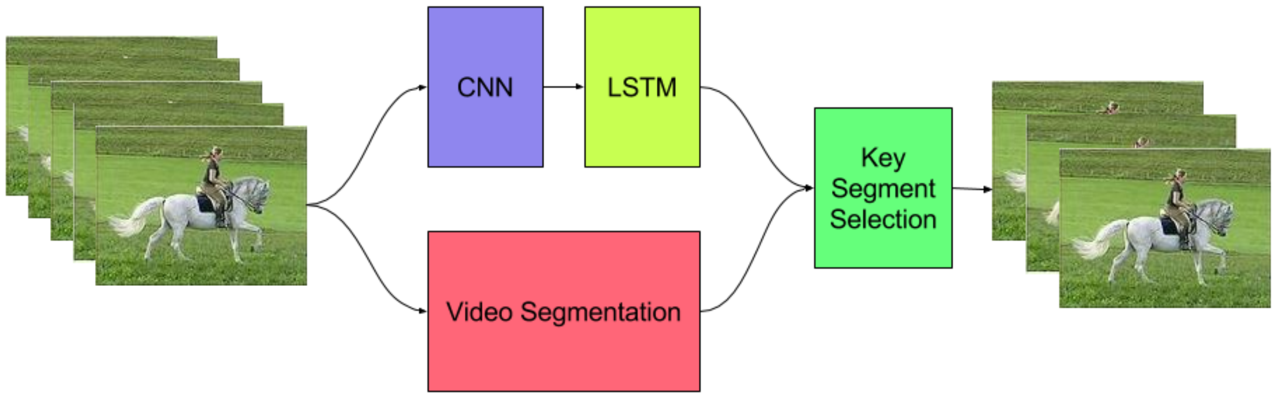

Figure 1. Overview of our approach

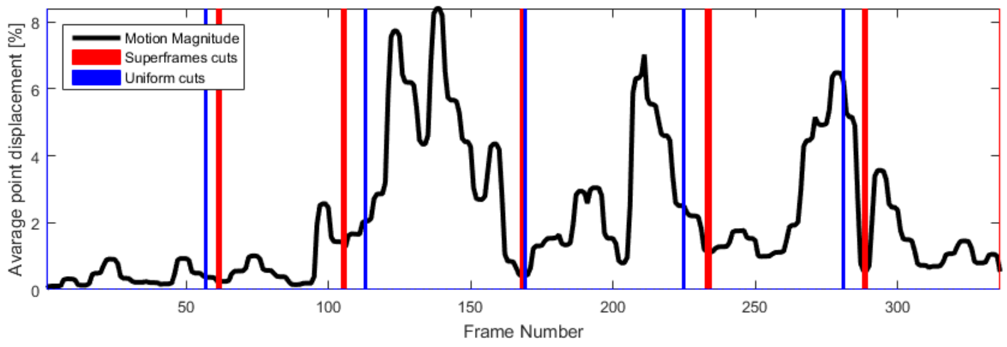

Figure 2. An illustration of superframe segmentation

To counter this problem [5] proposes a technique in which web images are used as prior information in unsupervised way to generate summaries. Similarly [11] proposes an unsupervised framework which uses title based image search results for video summarization.

Several approaches have been proposed which exploit sequential structure of videos. In [2] authors propose a probabilistic model for video summarization which uses temporal structure and visual features for diverse sequential subset selection. Recently LSTMs have shown to be extremely useful for tasks involving sequential structure and long term dependencies. For eg [13] uses a 2 layer LSTM network for generating video captions. But there hasnt been much work using Deep Learning in this area. [8] proposes a key frame based summarization framework which combines Convolutional Neural Networks with Restricted Bolztman Machines. Motivated from these successes we plan to build a framework similar to [13] which uses CNN features with LSTM network for video summarization.

3. Technical Approach

A video is a set of frames $V = \{S_1, S_2, \dots S_N\}$. A video summary is defined as a subset of these frames $\{S_{i_1}, S_{i_2}, \dots S_{i_M}\}$ such that $M = N$ where is the fraction of frames of the original video we want to keep in the summary.

We plan to use the Superframes framework from [3] for generating initial segments of the video. These superframes have their boundaries aligned with positions appropriate for a cut and hence give an appealing summary when combined.

Convolutional Neural Networks(CNNs) have been used recently for a variety of vision tasks. However, CNNs suffer from a major problem - they dont have persistence. For instance, if we want to classify the kinds of events happening at different points in a movie, we need to reason about previous events in the movie to inform later ones. This problem is addressed by Recurrent Neural Networks(RNNs) which have loops in them for information to persist.

For the video summarization problem, our design should capture representativeness(frames should depict main content of the videos), diversity(should not be redundant), interestingness (should have salient motion/appearance or trackable objects) and importance (should contain imp objects that drive visual narrative. We can capture things such as diversity and interestingness only when we visualize the video summarization problem as a temporal modeling problem. Hence, using RNNs for this project makes more sense.

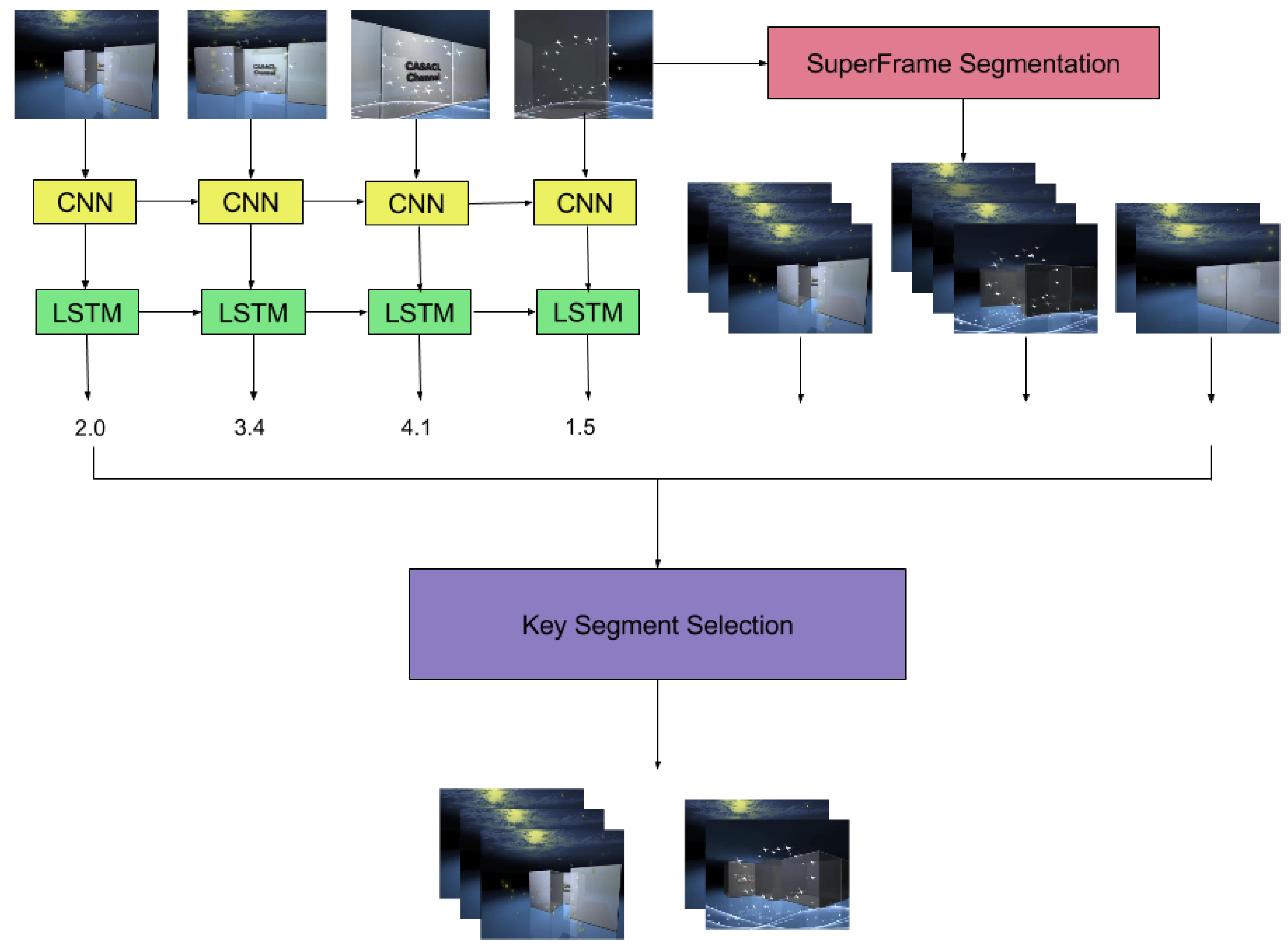

Figure 3. End-to-end architecture of our framework

Now for video summarization, we need to model longterm dependencies as well since two shots of a video long time apart need to be weighed with respect to each other in terms of importance while creating a summary for the video. Long Short Term Models(LSTMs) are a special kind of RNN capable of learning such long term dependencies. They were introduced by Hochreiter and Schmidhuber [4]. They have been refined by a lot of people since then and work well on problems that require modeling time as a dimension.

For an input x_t at time step t, the LSTM computes a hidden state h_t and a memory cell state c_t which is an encoding of whatever the cell has seen until time t:

$$i_t = \sigma(W_{x_i}x_t + W_{h_i}h_{t_1} + b_i)$$
$$f_t = \sigma(W_{x_f}x_t + W_{h_f}h_{t_1} + b_f)$$
$$o_t = \sigma(W_{x_o}x_t + W_{h_o}h_{t_1} + b_o)$$
$$g_t = \phi(W_{x_g}x_t + W_{h_g}h_{t_1} + b_g)$$
$$c_t = f_t \odot c_{t_1} + i_t g_t$$
$$h_t = o_t \odot c_t$$

Here ϕ represents the hyperbolic tangent non-linearity, σ is the sigmoidal non-linearity, $\odot$ represents the elementwise product with the gate value, and the weight matrices denoted by W_{ij} and biases b_j are the trained parameters. We plan to combine CNNs and LSTMs for this task. The basic architecture of our framework is shown in Figure 3. It consists of four components:

3.1. Video Segmentation

We will be using Superframe Segmentation framework of [3] for segmenting the videos. It uses motion cues and principles from video editing theory to compute segment boundaries which are visually aesthetic. The main advantage of this approach is that it works well even for raw videos.

3.2. Convolutional Neural Network

We are using the 19 layer VGGNet trained on ImageNet to extract features from frames of the videos. VGGNet consists of 16 convolutional layers followed by 3 fully connected layers. The width of convolutional layers is quite small, starting from 64 in the first layer and then increasing by a factor of 2 after each max-pooling layer, until it reaches 512.

3.3. Long Short Term Memory Model

During the training phase, we shall supply the LSTM with a features extracted by the CNN from input frames in a video. The LSTM shall learn to compute a sequence of hidden states $(h_1, \ldots, h_n)$ and shall output an importance score for every frame in the video. The LSTM shall be trained to minimize the loss as explained in the next section.

During test time, the scores generated by our model are used to determine whether or not a particular frame would be included in our summary. Only frames with importance scores above a certain threshold are included in the summary. We can vary the threshold to indicate what percentage of the video the summary captures.

3.4. Key Segment Selection

After the frame level importance scores produced by LSTM model and segment boundaries produces by Super frame Segmentation framework, video summarization can be formulated as the 0/1 Knapsack Problem:

Given a time budget t , maximize the total importance score of the shots included in a summary. The solution will be a vector of size K (the number of frames in a video), whose values are either 0 (not in summary) or 1 (include in summary). We use the framework provided along with TVSum50 dataset for solving this problem. It uses Dynamic Programming to compute the optimal solution.

4. Loss functions

We experiment with 3 loss functions for training our models:

1. Mean Squared Error Loss:

$$L(x, y) = \frac{1}{n}\Sigma_{i=1}^{n}(x_i - y_i)^2$$

2. Absolute Loss:

$$L(x, y) = \frac{1}{n}\Sigma_{i=1}^{n}|(x_i - y_i)|$$

3. Smooth L1 Loss:

$$\begin{aligned} L(x, y) &= \frac{1}{n}\Sigma_{i=1}^{n}0.5 * (x_i - y_i)^2 \ \text{ if } \ |(x_i - y_i)| \leq 1 \\ &= \frac{1}{n}\Sigma_{i=1}^{n}|(x_i - y_i)| - 0.5 \quad \text{otherwise} \end{aligned}$$

where n is the batch size, x is ground truth frame importance score and y is predicted frame importance score.

5. Dataset

We evaluate our framework on a real world dataset: TVSum50 (Yahoo! Title-based video summarization dataset).

5.1. TVSum50

TVSum50 contains 50 videos of various genres (e.g., news, how-to, documentary, vlog, egocentric) and 1,000 annotations of shot-level importance scores obtained via crowdsourcing (20 per video). Each shot is 2 seconds long and is annotated with a score from 1 (not important) to 5(very important). These videos have been collected from YouTube and come with the Creative Commons CC-BY (v3.0) license. The shot-level importance scores are annotated via Amazon Mechanical Turk Each video was annotated by 20 crowd-workers.

6. Evaluation Metric

TVSum50 dataset has 20 annotations per video whereas SumMe has 15 annotations per video. Following [11] and [3] we use the $F1$ score defined below to evaluate the quality of our models.

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{1}$$

Precision and Recall are computed by comparing the summary vector generated by our model with the annotated summary. Overall score is average of score computed for each annotated summary. We will use the evaluation framework provided along with TVSum50 dataset for this.

7. Experiments

We compared the performance of following models with existing techniques on benchmark dataset TVSum50.

1. Convolutional Neural Network (fine-tuned from BVLC VGG Net) which predicts frame relevance score for each frame.
2. LSTM model which uses VGG CNN features to predict frame relevance score for each frame.
3. A bidirectional LSTM model which uses CNN features to predict frame relevance score for each frame.

7.1. Baselines

We compare our models against baselines and other approaches mentioned in [11]

1. **Sampling (SU and SR):** In this method video summary is generated by selecting shots either uniformly (SU) or randomly (SR) such that the summary length is within the budget.

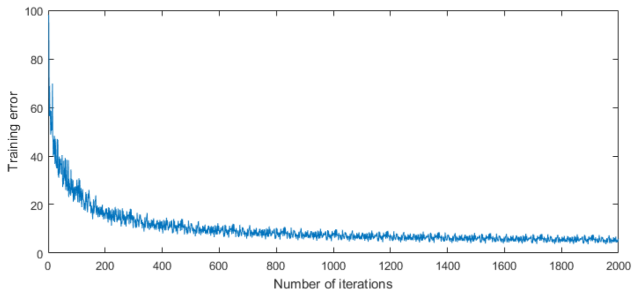

Figure 4. Decrease in the value of training error of LSTM with iterations

2. **Clustering (CK and CS):** We compare the performance with respect to 2 clustering based approaches: k-means (CK) and spectral clustering (CS). In these approaches video frames are clustered using the methods mentioned above with $k = 100$ clusters. After clustering the frames, shot level distances are computed by taking average distance of only those frames that belong to the most frequently occurring cluster within each shot. Most representative shots are then selected from each cluster on the basis of their closeness to centroids of top k clusters to generate a summary of length within the budget l.

3. **LiveLight(LL):** This approach generates a video summary by identifying and removing redundant shots over time. A dictionary of shots is updated online and used to measure the redundancy. Since this approach is not able to control the summary length, it selects shots with highest reconstruction error that fit within the budget l.

4. **Web Image Prior (WP):** [5] proposed an approach based on web image priors. This approach produces keyframes as a summary. Similar to the clustering based method (above) summary is generated by computing the shot-level distances and selecting those shots that are most representative of the top k largest clusters and fit within length budget l.

5. **Archetypal Analysis (AA1, AA2 and CA):** We compare with three versions of archetypal analysis: (AA1) which learns archetypes from video data only, (AA2) which uses a combination of video and image data and (CA) which uses co-archetypal analysis. For all three version number of archetypes is set to 200.

7.2. Implementation Details

We implemented our models in torch using the Sequential module and Fast LSTM implementation from RNN library [6]. We select 4 videos from each category for our training set and remaining 1 video from each category comprises our test set. Although training set has only 40 videos, each of them is quite long (5-7 mins). So we further divide each video into frame sequences (with 1 in 10 sampling) of length 128. This trick helps us in tackling the challenge of small dataset and also reduces the training time since each training eg has atmost 128 frames now. We chose the hyperparameters empirically. Figure 4 shows the plot of error in frame importance score for sequence of 128 frames with number of training iterations. We started with a learning rate of 0.0001 and reduce it by 0.8 after every 500 iterations.

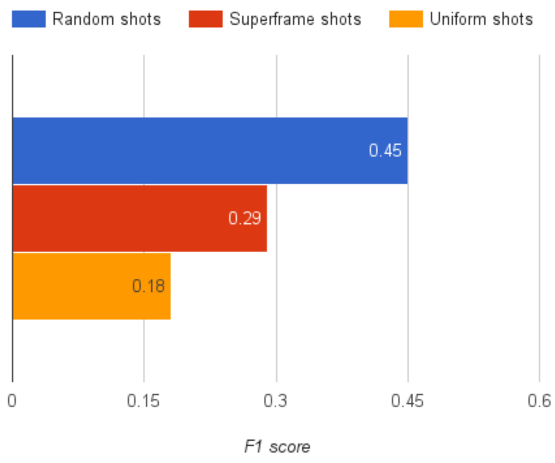

Figure 5. Comparison of different video segmentation techniques

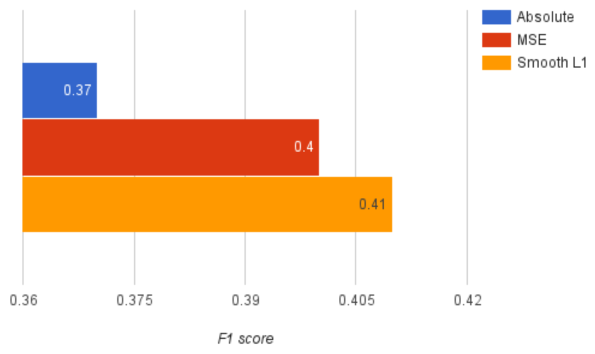

Figure 6. Comparison of different error criteria

7.3. Results

This section presents the results of various experiments mentioned above. We also investigate how the performance of our models vary with various parameters like loss function and type of video segmentation.

CNN vs LSTM model:
From experimental results in Figure 7 we can conclude

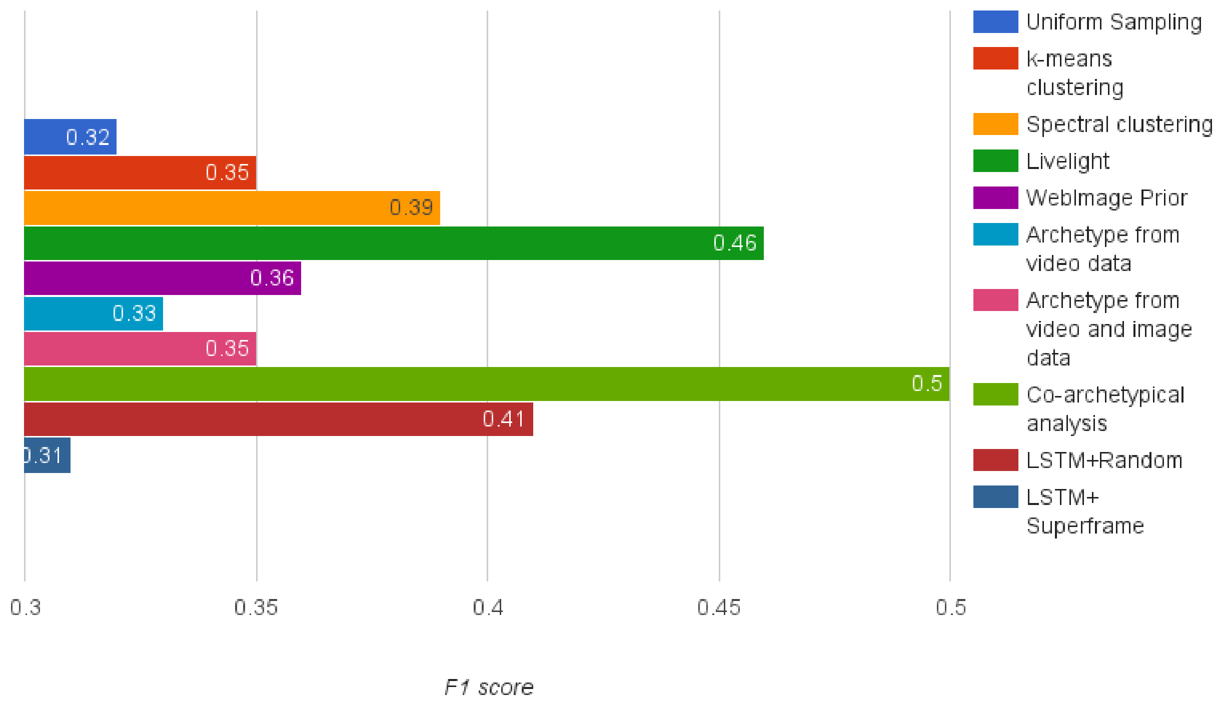

Figure 8. Comparison of our approach with existing baselines

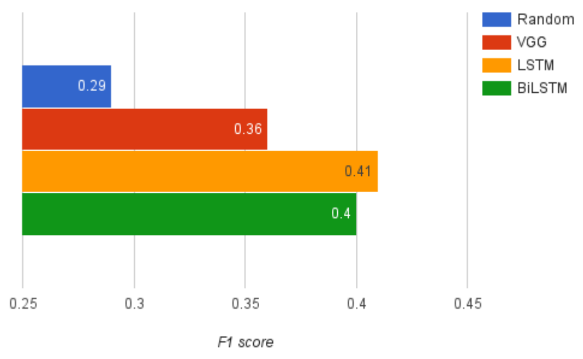

Figure 7. Comparison of different approaches tried by us

that both LSTM model performs better (F1 score of 0.41 vs 0.36) than VGG CNN model. This demonstrates that LSTM models which capture temporal information within the video are more effective in predicting the frame importance score for task of video summarization.

LSTM vs Bidirectional LSTM: This experiment was motivated from superior results obtained for bidirectional model for Sequence to Sequence Video to Text. Thus we expected that bidirectional LSTM to perform better. Results from Figure 7 suggest that both LSTM and Bi-LSTM give almost similar performance (F1 score: 0.41 vs 04). A possible reason for this could be the fact that for Sequence to Sequence Video to Text task, the backward dependencies in text (which is more structured than visual data) are very useful for the model but for our task backward dependencies in visual data are not important.

Random vs Uniform vs Superframe Segmentation: In this experiment we evaluated how the segmentation technique affects the performance of our model. Results from Figure 5 show that random segmentation performs better than superframe and uniform segmentation. This was surprising since we expected superframe segmentation to yield the best results. We believe that the main reason for this is that we were not able to configure the various parameters in superframe segmentation framework for TvSum50. Thus we used random segmentation in rest of the experiments.

MSE vs Abs vs Smooth L1: In this experiment we compared the performance of models trained with different loss functions. Results in Figure 6 show that absolute error performs the worst with F1 score of 0.27 Smooth L1 performs slightly better than MSE loss (0.41 vs 0.4) because it is less sensitive to outliers and prevents exploding gradients.

Comparison with existing models: Figure 8 presents the results of comparison of our model against the several baselines discussed in previous section. NOTE: We are not sure if the test-train split and segmentation technique used for our experiments and the ones on which baseline algorithms were tested are same. Hence it is not fair to directly compare our results with them, but we believe that this comparison still gives us some sense of how well our model is doing. We found that our best performing model (LSTM + Random Segmentation) gives an F1 score of 0.45 which surpasses all baselines except LiveLight(0.46) and Co-archetypal(0.5).

Qualitative Analysis: We also tried to qualitatively evaluate the summaries generated by our approach. We found that for most of the videos summary by our best performing model (LSTM + random segmentation) is able to capture the gist of the original video. However, since we are using random segmentation, we could observe that some segments included in our summaries are cut at inappropriate places. We could get more visually aesthetic summaries by using a better segmentation technique. Some sample results can be found here

LSTM Quantitative Analysis: We also tried to measure how good our LSTM framework is at predicting the frame importance scores. We compared the predicted scores with average of 20 user annotations. We got an average error of 0.46 with std deviation of 0.39. Since the number of frames per video is quite large and percentage of frames which can be scored high or low is pre-defined, this distribution is very similar to Gaussian. Hence mean + 2 x std deviation covers 90 percent of data. Thus for 90 percent of frames our error is less than 1.24 This shows that our LSTM model is quite good at predicting frame importance score.

8. Conclusion

We tried an approach which uses LSTMs to exploit the temporal information within the videos for the task of video summarization. Although we could not surpass the state of the art on this task, our model performs better than most of the existing techniques even with random segmentation. This suggests that our model is learning to predict very good frame importance scores. We believe that with better segmentation techniques and availability of larger dataset of annotated videos this approach might be able to surpass the state of the art techniques.

9. Future Work

A simple extension of our approach would be to use shifted segments similar to shifted grids approach used in [10]. Thus importance score of every frame would be predicted by taking average of shot scores of all shifted segments it lies in. We believe that this approach will reduce the outliers since it will smoothen the frame score transitions on segment boundaries.

References

[1] J. Ghosh, Y. J. Lee, and K. Grauman. Discovering important people and objects for egocentric video summarization. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1346–1353. IEEE, 2012.

[2] B. Gong, W.-L. Chao, K. Grauman, and F. Sha. Diverse sequential subset selection for supervised video summarization. In *Advances in Neural Information Processing Systems*, pages 2069–2077, 2014.

[3] M. Gygli, H. Grabner, H. Riemenschneider, and L. Van Gool. Creating summaries from user videos. In *Computer Vision–ECCV 2014*, pages 505–520. Springer, 2014.

[4] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[5] A. Khosla, R. Hamid, C.-J. Lin, and N. Sundaresan. Large-scale video summarization using web-image priors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2698–2705, 2013.

[6] N. Léonard, S. Waghmare, and Y. Wang. rnn: Recurrent library for torch. *arXiv preprint arXiv:1511.07889*, 2015.

[7] Z. Lu and K. Grauman. Story-driven summarization for egocentric video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2714–2721, 2013.

[8] O. Morere, H. Goh, A. Veillard, V. Chandrasekhar, and J. Lin. Co-regularized deep representations for video summarization. In *Image Processing (ICIP), 2015 IEEE International Conference on*, pages 3165–3169. IEEE, 2015.

[9] D. Potapov, M. Douze, Z. Harchaoui, and C. Schmid. Category-specific video summarization. In *Computer Vision–ECCV 2014*, pages 540–555. Springer, 2014.

[10] A. Recasens, A. Khosla, C. Vondrick, and A. Torralba. Where are they looking? In *Advances in Neural Information Processing Systems*, pages 199–207, 2015.

[11] Y. Song, J. Vallmitjana, A. Stent, and A. Jaimes. Tvsum: Summarizing web videos using titles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5179–5187, 2015.

[12] S. Sripada, V. G. Kasturi, and G. K. Parai. Summarization of sports videos based on unsupervised deep learning.

[13] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko. Sequence to sequence-video to text. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4534–4542, 2015.

[14] W. Wolf. Key frame selection by motion analysis. In *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*, volume 2, pages 1228–1231. IEEE, 1996.

[15] H. J. Zhang, J. Wu, D. Zhong, and S. W. Smoliar. An integrated system for content-based video retrieval and browsing. *Pattern recognition*, 30(4):643–658, 1997.