Online Video Summarization: Predicting Future To Better Summarize Present

Shamit Lal* Shivam Duggal* Indu Sreedevi
Delhi Technological Univeristy
shamit_bt2k13@dtu.ac.in, shivamduggal.9507@gmail.com, s.indu@dce.ac.in

Abstract

Automatically generating the summary of a video is a challenging problem due to its subjective nature. Most of the previous works in the field consider the entire video to extract out the important frames. Unlike them, our paper presents MerryGoRoundNet, a supervised learning approach to solve this problem in an online fashion. We observe that to effectively summarize a video, one needs to take into account both the spatial and temporal relations between video frames. MerryGoRoundNet utilizes encoder-decoder style architecture and convolutional LSTM to establish spatiotemporal relationship and generates the summary on the fly, thereby being more efficient than non-autoregressive counterparts in terms of time and memory. In order to make summary more diverse and complete, we augment our network with unsupervised task of next frame prediction and a supervised task of scene start detection and propose a loss function that explicitly focuses on achieving the right balance between continuity and diversity in the produced summary. Ablation study performed affirms the architecture and learning objective of our approach. Evaluation of MerryGoRoundNet on different datasets demonstrates superior performance among online summarization approaches and competitive performance when compared with offline approaches as well.

1. Introduction

Lately, the internet has been inundated with videos and video streaming services. The most popular video streaming website, YouTube, witnesses upload of more than 100 hours of video content every minute that is available for its billion-plus users. This glut of videos forces the users to rely on various metadata, like title, thumbnail, description or comments, to find the one they desire to watch. Even this metadata information may not be an accurate indicator of the semantic content of the corresponding video, which leaves users with the only option of skimming through the video to get a gist of it.

*Equal contribution

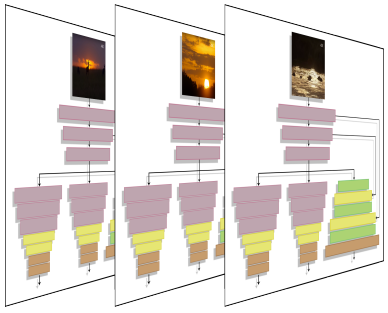


Figure 1: MerryGoRoundNet architecture. Each vertical entity shown in the figure represents an instance of the model at a timestep. Model instance at timestep i is connected to the model instance at timestep $i+1$ via the corresponding purple-colored recurrent blocks. Each component of the model is shown in detail in Figure 2.

Summary: Diverse, representative, epigrammatic and comprehensive synopsis of any entity (text, speech, image, video).

Thus, video summarization is a viable solution to this problem. Video summarization is inherently a sequential problem, where the decision to mark a frame as summary worthy or not depends on neighbouring frames as well. Therefore, if a particular frame has been assigned high importance by the summarizer system, the neighboring frames should also be given higher importance to achieve a continuous summary. However, selecting too many neighboring frames will increase the summary's length, thereby degrading its effectiveness. Also, waiting until the end of the video to initiate the summary generation task can be immensely time-consuming. Moreover, as the length of the video grows, summary generation can become ever more tedious. We propose an architecture that can generate crisp summary for a potentially very long video in real time as the video progresses.

Deep learning has proven to be quite effective when dealing with sequential problems. Long Short-Term Memory networks (LSTM) [11] have been shown to be effective in capturing long-range temporal dependencies, and at

the same time not suffering from optimization hurdles that simple recurrent neural networks (SRN) face. Convolutional LSTM [29], which extends fully connected LSTMs by using convolutions for both state-to-state and input-to-state transitions, allows one to incorporate both spatial and temporal information directly into the sequential model, thereby effectively learning meaningful features from the spatiotemporal data.

The paper proposes a sequential, autoregressive, end-to-end trainable, fully convolutional deep neural network solution to video summarization.

"Watching the FIFA world cup and missed the build up for the goal?". Our approach provides a solution to this problem, by generating the summary before the video ends, capturing the key-frames in a real-time fashion. The problem of automatic video summarization deals with automatically selecting important frames from a video and stitching them together to create a summary video. Unlike the previous state-of-the-art models in video summarization which predict the importance of a frame using energy-functions which require the global context, our approach can predict the relevance of a frame immediately after observing the current frame. [36, 12, 18] enhances the diversity of the summary by evaluating the inter-frame similarity either by using cosine-similarity or using the Determinantal point process (DPP). Such a process has a squared time complexity with respect to the number of frames in a video. Rather than applying any similarity function on the top of the frame features, the proposed network inherently ensures that the summary created is diverse enough, as explained in section 3.3, with time complexity being linear with respect to the number of frames.

The network takes as input the frames of a video and returns importance scores for them, which represents the likelihood of them being selected as part of the summary. It uses convolutional LSTMs for maintaining sequential relationships among frames as well as spatial relationships within a frame. The network is named *MerryGoRoundNet*, which does justice to its architecture. MerryGoRoundNet, shown in figure 1, is an augmented network, which uses multitask learning to perform better at each timestep on its primary task of video frame importance scoring. The multiple tasks have been chosen to complement the primary task at hand. This multitask learning helps network in learning meaningful and rich intermediate layers, and at the same time achieving the objective of creating a smooth and *complete* summary.

We augment our network with an unsupervised secondary task of predicting the next frame in the sequence. This branch enables the network to generate diverse and crisp summary, as explained in detail in section 3.3. This also restricts the model's capacity by providing a prior to the system in form of a cost function of the next frame prediction, thereby preventing overfitting. Domain adaption is one of the side-effects of using this unsupervised branch. The third task being performed is scene start detection. The supervised scene start branch helps the network to identify the start of new scenes, which generally have high likelihood of being part of the summary. Moreover, it also helps in restricting the parameters of the model to a confined space by forcing it to perform multiple tasks simultaneously. As part of this research, we also augmented the SumMe [8], TVSum [30] and Youtube [6] datasets with segment boundary labels.

To the best of our knowledge, convolutional LSTM based architectures and multitask learning have never been explored before for video summarization problem. The contributions in this paper are:

1. Designing a suitable loss function that learns an objective, resulting in a summary that is the right balance between continuity and diversity and makes the network learn more meaningful and descriptive features.

2. Exploring the usage of convolutional LSTMs to solve the problem of video summarization.

3. Demonstrating, through ablation studies, that next frame prediction task helps in achieving diverse and complete summary. Predicting next frame with high accuracy can mean that this frame is not summary worthy since network already had all information to correctly predict it.

4. Achieving better results than online approaches and competitive results when compared to offline methods on various datasets.

2. Related work

2.1. Video summarization

Video summarization involves detection of important frames automatically. Both supervised and unsupervised learning approaches have been used lately to tackle it. Supervised ones [36, 35, 9, 28, 13] learn to predict important frames or segments by leveraging human-generated summaries seen during training. However, due to lack of abundant annotated data, several unsupervised solutions have been proposed for this problem. Traditional unsupervised approaches [30, 22, 19, 38, 34] rely on low-level indices, hand-crafted features and manually determined criteria to determine frame importance.

Some of the earlier approaches [2, 26, 23] focused on certain categories or genres of videos, thereby restricting their systems to focus on particular domains. For example, summarizing football games poses less difficulty. The system can directly make use of domain knowledge and structure of the game to select important segments.

State-of-the-art approaches [36, 20, 18, 12] used recurrent neural networks or their LSTM counterparts to identify the temporal dependencies between the video's atomic elements. Zhang *et al*. [36] proposed LSTM for modelling temporal-dependency among video frames. They further augmented it with Determinantal Point Process (DPP), resulting in diverse summaries. Otani *et al*. [20] used features extracted by deep recurrent neural networks to cluster the video segments in a semantic space. The points in the semantic space, that correspond to the center of clusters, were then sampled to produce a summarized video.

Ji *et al*. [12] used an encoder-decoder architecture, in which the encoder uses bidirectional LSTM and the decoder is a stack of two attention-based LSTMs. The recent success of the attention-based encoder-decoder systems, particularly in the field of NLP, reaffirmed the state-of-the-art results of this paper. However, unlike our approach, this paper encodes the complete video before predicting the importance of each frame. Moreover, the paper mentioned the need for more annotated supervised data in its future-works.

Mahasseni *et al*. [18] proposed a complex adversarial framework to train a deep neural network to minimize the distance between the distribution of summarizations and corresponding training videos. They presented multiple ablations of their proposed model: with and without GAN loss, using DPP, adding sparsity loss and diversity regularizers. Similar to Ji *et al*., they used bidirectional LSTMs, thereby being non-autoregressive in frame importance prediction task.

Talking about online methods, [38] proposed a network dubbed as LiveLight, which maintains a dictionary which is further used to guide the reconstruction of previously unseen frames using sparse coding. Moreover, they extract low-level histogram of gradient features out of their segments. They divide the video into equal segments of 50 frames, considering each video as a single scene. Compared to them, our approach uses deep recurrent networks to extract out high-level features and operates on much more atomic elements (frames), compared to segment of frames.

Zhang *et al*. [37] proposed an unsupervised summary generation method based on identifying key-object motions and used an online motion auto-encoder method to represent the dictionary of past frames. Unlike them, we propose a method that generates the next frame to better learn the temporal dependencies.

2.2. Next frame prediction

Associating auxiliary classifier to the intermediate hidden layers adds not only to the transparency of the latent variables but also enhances the effectiveness of the network to learn without having vanishing or exploding gradients. Lee *et al*. [15] introduced Deep Supervised Nets (DSN) in which not only the last layer but also the intermediate layers directly learn to predict the target variable using the squared hinge loss of the SVM. Adding classifiers to the hidden layers leads to simpler gradient back-propagation through these layers.

Rasmus *et al*.'s [24] work on Ladder Network, which combines supervised learning with unsupervised learning, eases this need for layer-wise training. Their semi-supervised Ladder Network can be considered as a stack of Denoising Autoencoders (DAE), where the denoising cost function associated with each layer acts as a prior for the layers to learn.

Pezeshki *et al*. [21] investigated multiple variants of the Ladder network and postulated the need of having lateral connections. Video Ladder Network [4] used this Ladder Network for video next frame prediction.

Unlike the previous works in which the unsupervised and the supervised tasks minimize the same cost function, MerryGoRountNet augments the supervised video summarization task with unsupervised next frame prediction task.

3. Proposed approach

3.1. Architecture

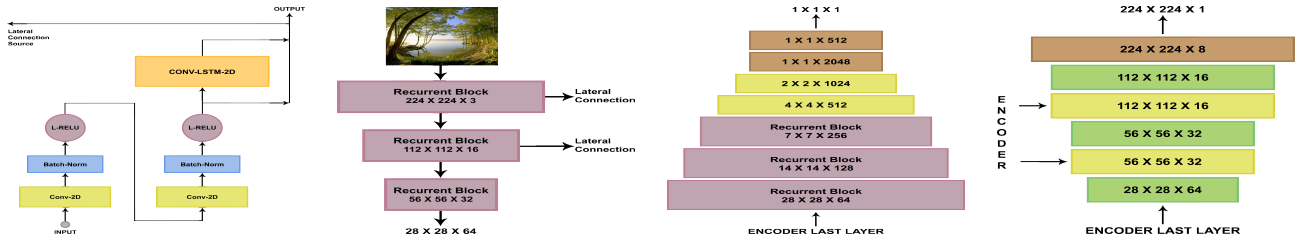
The complete system, dubbed as MerryGoRoundNet, takes as input a sequence of frames, and for each frame in the sequence provides 3 outputs - likelihood of the frame being in the summary, a binary value representing whether the current frame is a scene start or not, and a dense output representing the next frame in the sequence. At each timestep, frame importance is predicted based on the information from the current and the past frames. It learns both the temporal and spatial relationships within the data.

The proposed architecture makes use of the temporal information embedded within the frame sequence using a recurrent block as shown in Section 3.1.1. The architecture of MerryGoRoundNet is inspired by the success of multi-task learning based approaches [14, 7, 5, 27, 3]. It can be viewed as a single encoder, multiple decoder system (one upsampling-decoder and two down-sampling decoders), where every task is performed by the shared encoder and the corresponding decoder. Section 3.3 provides the intuition about how these subsystems fit in the overall picture and how they complement each other in achieving better performance on the primary task.

The following subsections describe the recurrent block, the shared encoder, and all the three decoders in detail.

3.1.1 Recurrent block

The proposed recurrent block is represented in Fig 2a. This is the basic building block for the MerryGoRoundNet. It consists of a stack of two-dimensional convolution layers, followed by a uni-directional convolutional LSTM layer. We used batch-normalization after each convolution layer, followed by Leaky-Relu activation. In each recurrent block,



(a) Recurrent Block (b) Encoder (c) Down-Sampling Decoder (d) Up-Sampling Decoder

Figure 2: Main components of MerryGoRoundNet. Purple boxes represent the recurrent blocks. Brown and yellow boxes respectively represent the 1×1 and 3×3 convolutions. Transposed convolutions are shown by green boxes. The dimensions written inside boxes represent the dimensions of input feature maps to that block. (c) 3×3 convolutions have stride 2. (d) 3×3 convolutions have stride 1.

the topmost convolution layer has a stride of 2, rest all the layers have stride 1. Each frame serves as an input to the recurrent block at different timesteps. Moreover, the convolutional LSTM layer receives its hidden state feature map, representing the previously seen frames, from the previous timestep. The outputs from all the layers are sent to the upsampling decoder as lateral connections.

3.1.2 Encoder

The encoder $\mathcal{E}$, shown in Fig 2b, takes as input the current frame in the sequence and outputs a lower-dimensional feature map, which is operated on by the decoders. It can be represented as a stack of recurrent blocks, defined in Section 3.1.1.

Each recurrent block in the encoder has an open lateral connection, which is used by the up-sampling decoder. These lateral connections act as a prior to the encoder, forcing it to learn semantically rich and meaningful features representing the video frame sequence.

3.1.3 Down-sampling decoders

Two down-sampling decoders are used, one for predicting the frame importance (decoder $\mathcal{D}_1$) and the other for scene boundary detection (decoder $\mathcal{D}_3$). The decoders used for these tasks share the same architecture, which is shown in Fig 2c.

Decoder, just like the encoder, is represented as a stack of recurrent blocks, followed by a stack of convolution layers to completely down-sample the input. No fully connected layers are used for down-sampling, rather 1×1 convolution layers are used as first introduced in [17] and used in [31]. After each convolution layer, batch-normalization followed by Leaky-Relu activation function are applied.

3.1.4 Up-sampling decoder

Up-sampling decoder $\mathcal{D}_2$, shown in Fig 2d, is used for next frame prediction. This decoder and the encoder together represents a ladder network [25], with lateral connections coming into the decoder from the encoder. These lateral connections help the network converge faster by providing better gradient support to the intermediate layers. Moreover as mentioned in [25], this helps the higher layers focus on more abstract, invariant features, leaving the details for the lower layers. The upsampling in the decoder is performed using transposed convolution layers [33].

As the input to the MerryGoRoundNet is a normalized image with the value of each pixel between 0 and 1, to get a similar output, sigmoid activation is used post the last 1×1 convolutional layer in the up-sampling decoder.

3.2. Learning objective

Training of MerryGoRoundNet requires jointly training the encoder $\mathcal{E}$ and decoders $\mathcal{D}_1$, $\mathcal{D}_2$ and $\mathcal{D}_3$. For this, we propose a unified loss function comprimising of several terms, weighted appropriately by constants λ_1, λ_2, λ_3 and λ_4.

$$\mathcal{L}^i_{total} = \lambda_1 \mathcal{L}^i_{d1} + \lambda_2 \mathcal{L}^i_{d2} + \lambda_3 \mathcal{L}^i_{d3} + \lambda_4 \mathcal{L}^i_{div} \quad (1)$$

where $\mathcal{L}^i_{total}$ is total loss at timestep i. The first loss $\mathcal{L}^i_{d1}$ (sigmoid cross entropy loss) is the loss incurred by the network when it wrongly predicts the importance of a frame.

$$\mathcal{L}^i_{d1}(\hat{s^i}, s^i) = -(s^i \log_2 \hat{s}^i_j + (1 - s^i) \log_2 (1 - \hat{s}^i)) \quad (2)$$

where

$$\hat{s^i} = Sigmoid(\mathcal{D}_1(\mathcal{E}(v^i))) \quad (3)$$

v^i is the i^{th} frame of the video and s^i and $\hat{s}^i$ are respectively the true and predicted frame importance scores.

The loss term $\mathcal{L}^i_{d2}$ represents the error in predicting the next frame of the video. Decoder $\mathcal{D}_2$ predicts the grayscale next frame which is compared with the grayscale version of original next frame to calculate the loss.

$$\mathcal{L}^i_{next\_frame}(\hat{G^{i+1}}, G^{i+1}) = \frac{\sum_{h,w}(G^{i+1}_{h,w} - \hat{G}^{i+1}_{h,w})^2}{H \times W} \quad (4)$$

$$\mathcal{L}^i_{d2}(B^{i+1}, \mathcal{L}^i_{next\_frame}) = B^{i+1} \times \mathcal{L}^i_{next\_frame} \quad (5)$$

$$\hat{G}^{i+1} = Sigmoid(\mathcal{D}_2(\mathcal{E}(v^{i+1}))) \quad (6)$$

where H, W are dimensions of the frame, G^{i+1} is the original normalized grayscale next frame and $\hat{G}^{i+1}$ is the predicted grayscale next frame image of the same size. B^{i+1} is the true *new scene start* label, which is 0 when next frame is the start of a new scene, otherwise 1. Since network won't be able to predict next frame when there is a change of shot, this prevents wrongly optimizing the network. $\mathcal{L}^i_{d2}$, being a pixel wise difference between predicted and ground truth next frames, may lead to blurring of the predicted frames. However, this loss works well for our problem since we are not interested in exact prediction of next frame and absolute value of this loss term. We only want this loss term to guide frame importance prediction and to be relatively less when the network predicts next frame having a structure similar to the original next frame than when the network is not able to predict the next frame correctly at all.

$\mathcal{L}^i_{d3}$ is the loss for video shot boundary prediction.

$$\mathcal{L}^i_{d3}(\hat{B^i}, B^i) = -(B^i \log_2 \hat{B^i} + (1 - B^i) \log_2 (1 - \hat{B^i})) \quad (7)$$

$$\hat{B^i} = Sigmoid(\mathcal{D}_3(\mathcal{E}(v^i))) \quad (8)$$

where $\hat{B^i}$ is the predicted new scene start label.

$\mathcal{L}_{div}$ loss term helps in increasing the diversity by giving higher importance to frames which would result in summary being more *complete*. The intuition behind this loss term is provided in Section 3.3.

$$\mathcal{L}^i_{div}(s^i, \hat{s^i}, \mathcal{L}^{i-1}_{next\_frame})) = |s^i - \hat{s^i}|(s^i \mathcal{L}^{i-1}_{next\_frame} + (1 - s^i)(1 - \mathcal{L}^{i-1}_{next\_frame})) \quad (9)$$

3.3. Intuition behind MerryGoRoundNet

Harri Valpola's ladder network [32] learns the unknown latent variables by optimizing the cost function in the same way as in the stochastic gradient descent of supervised network. More specifically, their network compares the input/layers reconstructed from the noisy input/layers (as in hierarchical auto-encoder), with the true input/layers performing the supervised task. This would ensure that the unsupervised network won't force the supervised network to learn any input specific representations not needed for the supervised task at hand, rather would help in better selection of the features that correlate to the principal components of the supervised task.

Unlike their architecture, MerryGoRoundNet supports the unsupervised learning of the latent variables through a task (next frame prediction) different from the supervised video frame importance prediction. Executing this unsupervised task alongside helps to learn the video representations and complements the primary task of frame importance prediction. This can be visualized as follows: Summary of a video refers to such a subset of frames which can completely represent the video. Using next frame prediction as an auxiliary task helps in deciding whether appending the next frame to the summary adds to its value or not. If MerryGoRoundNet can predict the next frame with high confidence, then selecting that next frame as part of the summary doesn't help much in diversifying the summarized video (though it does lend some continuity to the overall summary).

The loss term in equation 9 brings this intuition into the learning objective. It doesn't impact the learning objective when predicted and ground truth importance scores are equal. In other cases, the network is penalized when it deviates from the above argument. For example, when the ground truth importance is high and predicted importance is low, network will be penalized heavily when next frame prediction loss is high. This is in line with our reasoning as higher next frame loss means that the frame may be bringing new information that may be critical for the summary.

Secondly, [23] uses DPP over their LSTM network to enhance the diversity within the predicted subset of important frames. Using the above-defined visualization, we argue that the next frame prediction task will also help to generate a diverse summary, achieving the objective of DPP. This can further be understood by considering the following analogy between the above-mentioned statement and space spanned by some basis vectors.

Suppose there are n basis vectors. Then, any other vector a in the same space will be a linear combination of these n basis vectors and wouldn't provide any new information about the space. Similarly, if the network can predict the next frame with high confidence, that next frame can be represented as some non-linear combination of the weights and past frames seen by the network and will most likely lie in the same video scene that network is currently processing. Including this frame in the summary is unlikely to provide much additional information. Hence, the network is penalized when it gives high importance to frames when next frame prediction decoder $\mathcal{D}_2$ at previous timestep already predicted that frame with high confidence.

Also, this architecture is well structured to adapt to various domains of videos. Fine-tuning and unsupervised pretraining [10] are the common approaches for domain adaption. Auto-encoders have been used in the past for performing the task of domain adaption. Moreover, as mentioned

in [32], continuing the unsupervised training along-side supervised training, rather than just restricting the unsupervised training part to the pre-processing stage, helps the network identify better-correlated features for the supervised task. Thus, the augmented next frame prediction branch helps the network to learn domain invariant features and hence fulfills the requirement for larger annotated datasets for video-summarization. Particularly, the lateral connections between the upsampling decoder and the encoder enable the higher layers to focus on invariant features, leaving the task of learning fine details for the low-level layers.

The aim of augmenting the scene-start branch into the network was never to generate state-of-the-art results in scene boundary detection task but to augment the network's capability of predicting the important frames. The primary intuition behind this branch was to assist MerryGoRoundNet in differentiating between clearer scene frames and blurry scene transition frames. Transition refers to the short duration as focus moves from one scene to another. Thus, augmenting such a branch helped the network to focus on the main scenes and better compress the video.

4. Inference

MerryGoRoundNet is auto-regressive in nature since the importance scores can be predicted and summary can be generated while processing the video. During inference, a sequence of frames is given as input to the network. Decoder $\mathcal{D}_1$ assigns the importance score (between 0 and 1) to each frame, which is the likelihood of that frame being part of the summary. The loss term in Equation 9 ensures that during inference, network will tend to assign relatively higher importance to frames as long as they provide new information, providing the right balance between continuity and diversity. For generating the summary, we found that setting the threshold $\theta = 0.7$, i.e. selecting frames having score greater than 0.7 resulted in the best summary.

The model provides various tuning points that can be used to alter the summary generated during inference. θ can be used to control the length of the summary generated during inference, which will be inversely proportionate to θ. To obtain the summary of a specific length, say $y\%$ of the video's length, we used $0-1$ Knapsack problem. The video was divided into segments using KTS algorithm. The *weight* of each segment will be the number of *important* frames, i.e. frames having score greater than θ. The *price* of each segment will be the average score of important frames. The *weight* that can be accommodated by the *knapsack* will be $y\%$ of the video's length.

Solving this $0-1$ Knapsack problem will provide the most optimal segments. Selecting important frames out of these segments and stitching them together will be the summary. Unlike previous works, that select entire segment as part of the summary, our approach selects frames that capture the gist of that segment. By not selecting the unimportant frames of the segment, that *weight* can be used to incorporate frames from other segments. As mentioned above, our learning objective ensures that important frames will be continuous. Hence, we don't need to select entire segment to achieve continuity and the final summary will be complete and without abrupt scene changes.

5. Experiments

5.1. Datasets

MerryGoRoundNet was evaluated on TVSum [30] and SumMe [8] datesets. For training, the two datasets were further augmented with Youtube [6] dataset. TVSum contains 50 videos from YouTube, each being 1-5 minutes in length. These videos are distributed equally among 10 categories defined in the TRECVid Multimedia Event Detection (MED). SumMe dataset consists of 25 videos, capturing various events like holidays and cooking. YouTube dataset also includes 50 videos. These videos are collected from websites and their lengths vary between 1 to 10 minutes. All these datasets were augmented with scene start label.

5.2. Evaluation metric

For fair comparison with recent works [36, 18], instead of using the inference approach described in Section 4, key shot generation method explained in [36] is utilized. Evaluation is done using the keyshot based metric proposed in [36]. Let A be the predicted keyshots and B be the keyshots annotated by users. A's duration is restricted to be less than 15% of the original video. Precision and recall can then be defined as follows:

$$P = \frac{\text{overlapped time duration of A and B}}{\text{A's duration}} \tag{10}$$

$$R = \frac{\text{overlapped time duration of A and B}}{\text{B's duration}} \tag{11}$$

Their harmonic mean F-score, given by:

$$R = \frac{2P \times R}{P + R} \times 100 \tag{12}$$

is then used as the evaluation metric. The steps specified in [36, 18] are followed to convert between frame scores, keyframes, and keyshot summaries and to generate ground truth keyshot summaries for datasets which provide frame scores.

5.3. Training details

We started with training the MerryGoRounNet only for next-frame prediction initially, as it is common to perform unsupervised pre-training [10] before the supervised tasks. We trained it for 10 epochs, with an initial learning rate of 0.2 and momentum of 0.9. After 10 epochs, simultaneous

Dataset	Method	Supervised/Unsupervised	Online/ Offline	F-Score
Training: 80% SumMe + YouTube Evaluation: 20% SumMe	vsLSTM [36]	supervised	offline	37.6
	Online Motion-AE [37]	**unsupervised**	**online**	**37.7**
	LSTM-VS(w/o attention) [12]	supervised	offline	38-39
	dppLSTM [36]	supervised	offline	38.6
	SUM-GAN$_{dpp}$ [18]	unsupervised	offline	39.1
	Gygli *et al*. [9]	supervised	offline	39.7
	MerryGoRountNet(ours)	**supervised**	**online**	**39.7**
	Zhang *et al*. [35]	supervised	offline	40.9
	SUM-GAN$_{sup}$ [18]	supervised	offline	41.7
	Li *et al*. [16]	supervised	offline	43.1
	A-AVS [12]	supervised	offline	43.9
	M-AVS [12]	supervised	offline	44.4
Training: 80% TVSum + YouTube Evaluation: 20% TVSum	**LiveLight [38]**	**unsupervised**	**online**	**46.6**
	LSTM-VS˙(w/o attention) [12]	supervised	offline	49-50
	TVSum [30]	unsupervised	offline	51.1
	Online Motion-AE [37]	**unsupervised**	**online**	**51.5**
	SUM-GAN$_{dpp}$ [18]	unsupervised	offline	51.7
	Li *et al*. [16]	supervised	offline	52.7
	MerryGoRountNet(ours)	**supervised**	**online**	**53.1**
	vsLSTM [36]	supervised	offline	54.2
	dppLSTM [36]	supervised	offline	54.7
	SUM-GAN$_{sup}$ [18]	supervised	offline	56.3
	A-AVS [12]	supervised	offline	59.4
	M-AVS [12]	supervised	offline	61.0

Table 1: Performance comparison with state-of-the-art methods using F-Score evaluation metric. Online systems use only past and current frames whereas offline systems utilize even future frames to evaluate the importance of the current frame. All online approaches have been highlighted. MerryGoRoundNet achieves best performance among online methods and surpasses most of the offline approaches as well, while being more efficient in time and memory.

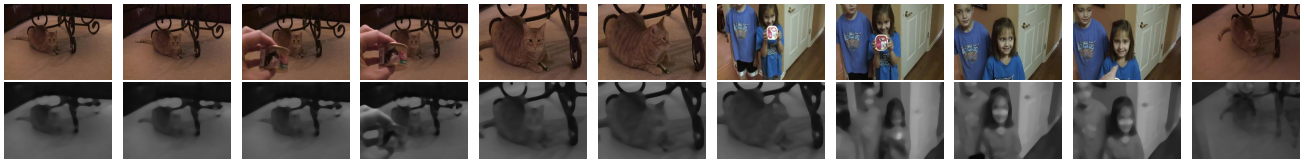


Figure 3: First row represents the summary and the next row represents the next-frame predicted at the timestep just before the corresponding summary frame.

training of all the MerryGoRoundNet branches was initiated. The same learning rate was used for training the MerryGoRoundNet. Learning rate was decayed by a fixed factor of 0.94 after every 500^{th} iteration. Combined training was carried out for another 20 epochs. Each frame was normalized before training. This particularly helped in training the next-frame prediction ladder network.

The LSTM state-state recurrent weights were initialized with orthogonal matrices, as orthogonal initialization helps to maintain the gradients across timesteps [1]. The upsampling layer weights were initialized to perform bilinear interpolation. All the other weights were drawn randomly from a zero-mean gaussian with standard deviation 0.02. MerryGoRoundNet was trained with a mini-batch size of 8 video sequences, where each sequence contained 50 frames. Each video was broken at 3 frames per second.

Initially, while unsupervised pre-training all the lambda values were initialized to 0, except λ_2, which was set as 100. Then, as the combined training of MerryGoRoundNet initiated, λ_2 was initialized to 30 and was iteratively downgraded by a factor of 0.8 after every epoch. λ_1 and λ_3 were set to 1 and 0.8 respectively throughout the training. λ_4 was set to 70, as this loss term guides multiple branches together. The motivation behind iteratively decreasing λ_2 is

Dataset	System	F-Score
SumMe	A	36.8
	B	36.9
	D	**39.7**
TVSum	A	49.2
	B	49.5
	C	52.6
	D	**53.1**

Table 2: Comparison of multiple systems mentioned in ablation study. Best results are highlighted.

to allow the gradient to completely focus on the improvement of frame importance branch and consider the next-frame branch output as another true source of information to train the network against.

5.4. Ablation study

To back the intuition with results, it was necessary to carry out ablation of different components of the MerryGoRoundNet. Following are the four systems evaluated:

- System A: Frame importance prediction alone.
- System B: System A with scene start detection.
- System C: System A with unsupervised next-frame prediction branch.
- System D (MerryGoRoundNet): System B with unsupervised next-frame prediction branch.

Table 2 shows that augmentation of the scene start detection branch to the frame importance prediction branch results in slight enhancement in the F-Score. Benefits were particularly observed for the videos with large transition period between scenes. Significant improvements were observed when augmenting the network with next frame prediction branch. Moreover, System D's superiority over the other two systems proves the intuition behind the architecture of MerryGoRoundNet.

6. Results and comparison

We compare our approach with previously used approaches. Table 1 displays the performance of various approaches in terms of evaluation metric defined in Section 5.2. To exhibit the domain adaptiveness of our approach, the results table contains a column specifying the datasets used for training the approaches. Most of the approaches used OVP, Youtube, and SumMe/TVSum datasets for the training of their systems, however, MerryGoRoundNet was trained only using Youtube and Summe/TVSum. Backed up by results, MerryGoRoundNet justifies the intuition it was built upon, generating a continuous, diverse, crisp summary.

The state-of-the-art approaches [18, 12, 36] used bidirectional LSTM to capture frame features and evaluated the frame importance by taking into consideration their long-term dependency with all the frames in both the directions (past and future). Moreover, [12] used global attention mechanism over all the encoded frames in their decoder. MerryGoRoundNet, on the other hand, leverages autoregressive framework to produce summary in an online fashion. This extra information carried by the future frames accounts for the slight gap in the F-Scores between our approaches. Also, the summary generated by MerryGoRoundNet is at par with the one generated by [36]. This verifies our hypothesis that the next-frame prediction branch is capable enough in ensuring diversity within the summary, without even observing all the frames. We operate at a linear rate with respect to the video frame count, compared to the quadratic rate of [18, 12, 36], thereby justifying the accuracy-efficiency trade-off. As shown in Table 1, we achieved better F-score metric in both TVSum and SumMe datasets compared to other online methods.

Figure 3 contains the summary and output of the next-frame prediction decoder for first 30 seconds of the video *https://www.youtube.com/watch?v=-esJrBWj2d8*. The video was processed at 3 FPS. Each output of next-frame decoder corresponds to the frame generated at the previous timestep for the summary frame. As can be seen from the figure, MerryGoRoundNet could not correctly predict the 7^{th} frame of the summary, and hence included it in the summary. Moreover, the summary generated correlates highly with human-generated summary. This highlights the fact that behavior of a human while generating a summary is similar to the intuition behind the MerryGoRoundNet. Additional summaries for some videos from TVSum test-set are provided in the supplementary material as GIFs.

7. Conclusion

Proposed MerryGoRoundNet explores the application of convolutional LSTM for real-time video summarization, taking into account both the spatial and temporal relations among data, both of which are essential to capture to provide a meaningful summary. We showed that augmenting the network with ladder network based next frame prediction branch and a scene start detection branch provides multiple benefits and complements the primary task. Not only do these additional tasks help in domain adaption, they also aid in generating a diverse and crisp summary. They also help in preventing over-fitting of the model, despite less amount of annotated data. Also, instead of selecting either keyframes (more diversity, less continuity) or keyshots (less diversity more continuity), we devised an inference method which provides us with the best of both worlds. Ablation study and results showed that MerryGoRoundNet achieves best performance among online approaches and competitive results overall, while being much more efficient than the previous approaches.

References

[1] M. Arjovsky, A. Shah, and Y. Bengio. Unitary evolution recurrent neural networks. *CoRR*, abs/1511.06464, 2015.

[2] N. Babaguchi, Y. Kawai, T. Ogura, and T. Kitahashi. Personalized abstraction of broadcasted american football video by highlight selection. *IEEE Transactions on Multimedia*, 6(4):575–586, 2004.

[3] R. Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, Jul 1997.

[4] F. Cricri, X. Ni, M. Honkala, E. Aksu, and M. Gabbouj. Video ladder networks. *CoRR*, abs/1612.01756, 2016.

[5] J. Dai, K. He, and J. Sun. Instance-aware semantic segmentation via multi-task network cascades. *CoRR*, abs/1512.04412, 2015.

[6] S. E. F. De Avila, A. P. B. Lopes, A. da Luz Jr, and A. de Albuquerque Araújo. Vsumm: A mechanism designed to produce static video summaries and a novel evaluation method. *Pattern Recognition Letters*, 32(1):56–68, 2011.

[7] S. Duggal, S. Manik, and M. Ghai. Amalgamation of video description and multiple object localization using single deep learning model. In *Proceedings of the 9th International Conference on Signal Processing Systems*, ICSPS 2017, pages 109–115, New York, NY, USA, 2017. ACM.

[8] M. Gygli, H. Grabner, H. Riemenschneider, and L. Van Gool. Creating summaries from user videos. In *European conference on computer vision*, pages 505–520. Springer, 2014.

[9] M. Gygli, H. Grabner, and L. Van Gool. Video summarization by learning submodular mixtures of objectives. In *Proceedings CVPR 2015*, pages 3090–3098, 2015.

[10] G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

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

[12] Z. Ji, K. Xiong, Y. Pang, and X. Li. Video summarization with attention-based encoder-decoder networks. *arXiv preprint arXiv:1708.09545*, 2017.

[13] H. Jiang, Y. Lu, and J. Xue. Automatic soccer video event detection based on a deep neural network combined cnn and rnn. In *Tools with Artificial Intelligence (ICTAI), 2016 IEEE 28th International Conference on*, pages 490–494. IEEE, 2016.

[14] S. Lal, V. Garg, and O. P. Verma. Automatic image colorization using adversarial training. In *Proceedings of the 9th International Conference on Signal Processing Systems*, ICSPS 2017, pages 84–88, New York, NY, USA, 2017. ACM.

[15] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu. Deeply-supervised nets. In *Artificial Intelligence and Statistics*, pages 562–570, 2015.

[16] X. Li, B. Zhao, and X. Lu. A general framework for edited video and raw video summarization. *IEEE Transactions on Image Processing*, 26(8):3652–3664, 2017.

[17] M. Lin, Q. Chen, and S. Yan. Network in network. *CoRR*, abs/1312.4400, 2013.

[18] B. Mahasseni, M. Lam, and S. Todorovic. Unsupervised video summarization with adversarial lstm networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[19] O. Morère, 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.

[20] M. Otani, Y. Nakashima, E. Rahtu, J. Heikkilä, and N. Yokoya. Video summarization using deep semantic features. *CoRR*, abs/1609.08758, 2016.

[21] M. Pezeshki, L. Fan, P. Brakel, A. C. Courville, and Y. Bengio. Deconstructing the ladder network architecture. *CoRR*, abs/1511.06430, 2015.

[22] D. Potapov, M. Douze, Z. Harchaoui, and C. Schmid. Category-specific video summarization. In *European conference on computer vision*, pages 540–555. Springer, 2014.

[23] D. Potapov, M. Douze, Z. Harchaoui, and C. Schmid. Category-specific video summarization. In *European conference on computer vision*, pages 540–555. Springer, 2014.

[24] A. Rasmus, H. Valpola, M. Honkala, M. Berglund, and T. Raiko. Semi-supervised learning with ladder network. *CoRR*, abs/1507.02672, 2015.

[25] A. Rasmus, H. Valpola, M. Honkala, M. Berglund, and T. Raiko. Semi-supervised learning with ladder network. *CoRR*, abs/1507.02672, 2015.

[26] A. Raventos, R. Quijada, L. Torres, and F. Tarrés. Automatic summarization of soccer highlights using audio-visual descriptors. *SpringerPlus*, 4(1):301, 2015.

[27] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015.

[28] A. Sharghi, B. Gong, and M. Shah. Query-focused extractive video summarization. In *European Conference on Computer Vision*, pages 3–19. Springer, 2016.

[29] X. Shi, Z. Chen, H. Wang, D. Yeung, W. Wong, and W. Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *CoRR*, abs/1506.04214, 2015.

[30] 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.

[31] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014.

[32] H. Valpola. From neural PCA to deep unsupervised learning. *ArXiv e-prints*, Nov. 2014.

[33] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus. Deconvolutional networks. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2528–2535, June 2010.

[34] 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.

[35] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman. Summary transfer: Exemplar-based subset selection for video

summarization. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, pages 1059–1067. IEEE, 2016.

[36] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman. Video summarization with long short-term memory. In *European conference on computer vision*, pages 766–782. Springer, 2016.

[37] Y. Zhang, X. Liang, D. Zhang, M. Tan, and E. P. Xing. Unsupervised object-level video summarization with online motion auto-encoder. *arXiv preprint arXiv:1801.00543*, 2018.

[38] B. Zhao and E. P. Xing. Quasi real-time summarization for consumer videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2513–2520, 2014.