Semi-Supervised Approach To Video Summarization

Anonymous WACV submission

Paper ID ****

Abstract

Automatically generating the summary of a video is a challenging problem due to its subjective nature. The paper presents MerryGoRoundNet, a semi-supervised learning approach to solve this problem. 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 temporal relationship. Apart from just predicting the likelihood of a frame belonging to the summary, we augment our network with unsupervised task of next frame prediction and a supervised task of scene start detection. These tasks help in domain adaption and making the summary smooth and diverse. Finally, during inference, we incorporate the best of both worlds, i.e. continuity and diversity into the final summary. Ablation study performed affirms the architecture and learning objective of our approach. Evaluation of MerryGoRoundNet on different datasets demonstrates competitive performance when compared with recent works.

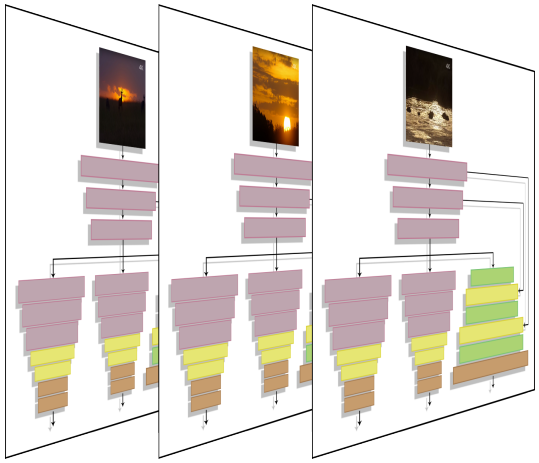


Figure 1: MerryGoRoundNet architecture.Each vertical entity shown in the figure represents an instance of the model at a timestep. Model instance at timestep i is connected to the model instance at timestep $i+1$ via the corresponding purple-colored recurrent blocks. Each component of the model is shown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

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.

Video summarization is a viable solution to this problem. It aims at providing a compact and descriptive version of the original video which successfully captures the essence of the same. Generating a summary for a video in a domain-independent manner can be a daunting task for computers, owing to the subjective nature of the problem. The high level of inter-dependency involved among frames adds to the complexity of the task. Humans, however, can tackle the problem rather easily by making use of the high-level semantic information present in the video. This helps in deciding whether a particular frame or shot would be suitable for the summary. Since there is a strong inter-dependency among frames, video summarization is inherently a sequential problem, where the decision to mark a frame as summary worthy or not depends on frames preceding as well as succeeding it. 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.

Deep learning has proven to be quite effective when dealing with sequential problems. Long Short-Term Memory networks (LSTM) [12] have been shown to be quite ef-

fective 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 [31], which extend 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, end-to-end trainable, fully convolutional deep neural network solution to video summarization. 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. The proposed learning objective ensures that the *right number* of similar frames are selected as part of the summary to achieve the correct balance between continuity and diversity of the summarized video. 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 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. It uses an encoder-decoder framework with one encoder and 3 decoders.

At each timestep, the encoder $\mathcal{E}$ accepts as input the current frame of the video and a vector representing the spatiotemporal information of previously seen frames, and produces an encoded representation of the same. This encoded representation is consumed by 3 decoders. These decoders correspond to video frame importance score prediction decoder $\mathcal{D}_1$, next frame prediction decoder $\mathcal{D}_2$, and start of a new segment (video shot boundary) prediction decoder $\mathcal{D}_3$. These encoders and decoders and the intuition behind them are explained in detail in Section 3. The loss functions corresponding to the three decoders are suitable weighted and optimized to train the entire network end-to-end by jointly training all the subsystems of MerryGoRoundNet.

Previous methods like [24] use KTS to identify optimal scene boundaries within a video and then identify the key frames within the segment. Our approach is based on interleaving the two tasks of optimal boundary detection and key frame detection. Moreover, to efficiently train the system despite the insufficient amount of annotated data, we augment our network with an unsupervised branch for predicting the next frame in the sequence. This branch not only helps in the better convergence of the system but also restricts the model capacity by providing a prior to the system in form of the cost function of the next frame prediction, thereby preventing overfitting. This also helps us to summarize previously unseen video genres. As part of this research, we also augmented the SumMe [9], TVSum [32] and Youtube [6] datasets with segment boundary labels.

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

1. Designing a suitable loss function that
   (a) learns an objective, resulting in a summary that is the right balance between continuity and diversity,
   (b) makes the network learn more meaningful and descriptive features and
   (c) does everything end-to-end without external processing steps.
2. Exploring the usage of convolutional LSTMs to solve the problem of video summarization.
3. Achieving competitive results on various datasets.

The rest of the paper is organized as follows. Section 2 covers the related work of video summarization. Section 3 describes the proposed MerryGoRoundNet, it's architecture, learning objective and the intuition behind it. Inference details at test time are provided in Section 4. We delineate the implementation details and ablation studies performed in Section 5. Section 6 shows the results achieved and comparison with some of the previous approaches. Finally, Section 7 concludes the paper.

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 [38, 37, 10, 30, 14] learn to predict important frames or segments by leveraging human-generated summaries seen during training. Unsupervised approaches [32, 25, 21, 39, 36], on the other hand, makes use of low-level indices or manually determined criteria to determine frame importance.

Earlier approaches focused on certain categories or genres of videos. 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. N. Babaguchi *et al*. [2] generated personalized abstractions of American football broadcast videos by automatically detecting significant game events with the

help of metadata and structure associated with the game. A. Raventos *et al*. [28] generated highlights of soccer games by further incorporating audio information with visual features. Danila Potapov *et al*. [24] leveraged multiple SVMs trained on different video categories. At test time, they used Kernel Temporal Segmentation (KTS) to segment their video into segments, score the segments using SVM for that video category and select highest scored segments to form a summary.

Recently, with the success of deep learning, several video summarization approaches using deep neural networks have been proposed. Zhang *et al*. [38] proposed LSTM for modelling temporal-dependency among video frames. They further augmented it with Determinantal Point Process (DPP), resulting in better summaries. Otani *et al*. [22] used deep neural networks to map video segments to a semantic space, which, they argued can encode information vital to make segment importance decisions. The points in the semantic space, that correspond to the center of clusters, were then sampled to produce a summarized video.

Several approaches utilizing unsupervised learning methods have also shown appreciable results when generating meaningful summaries. Mahasseni *et al*. [19] proposed an adversarial framework to train a deep neural network to minimize the distance between the distribution of summarizations and corresponding training videos. Song *et al*. [32] generated summarized video by using title-based results for image search. They argued that the title is generally highly descriptive of the crux of the video. Therefore, images that are related to the title can be used as a proxy for principal visual concepts of the key topic.

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*. [16] 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 [26] work on Ladder Network, which combines supervised learning with unsupervised learning, eases this need for layer-wise training. Their 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*. [23] investigated multiple variants of the Ladder network and postulated the need of having latent 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. Moreover, adding an extra branch of scene start detection to the MerryGoRoundNet helps to train the model efficiently and also enables the network to perform both key-frame and key-segment detection simultaneously using shared-convolution features.

2.3. Segment boundary detection

Selecting segments having highest importance results in a much more continuous summary than selecting highest scored frames. Potapov *et al*. [24] proposed KTS for change point detection in a video. The algorithm takes a frame-to-frame similarity matrix as input and outputs a list of change points corresponding to the segment boundaries. KTS makes use of dynamic programming to optimize its underlying objective.

Mas and Fernandez [20] detected shot boundaries using color histograms. Their system is capable of detecting both abrupt boundaries by analyzing color histogram differences between frames and utilizes temporal color variations for smooth boundary detection. Gygli [8] proposed a convolutional neural network architecture, which is end-to-end trainable, for detecting shot boundaries. Their model is fully convolutional in time, enabling the network to make boundary decision based on large temporal context.

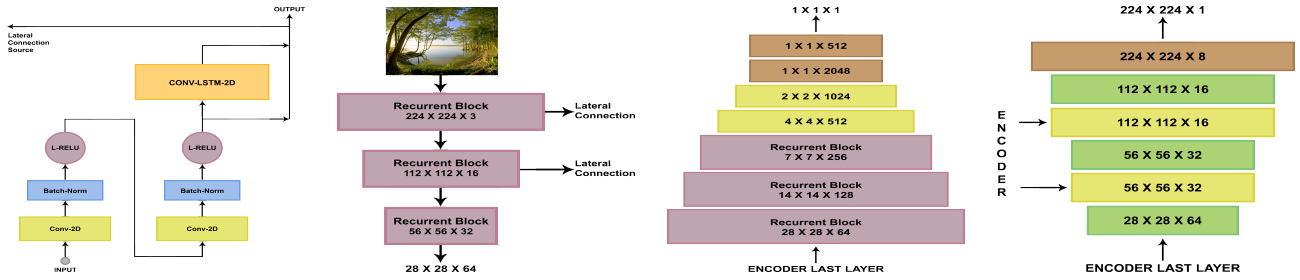
3. Proposed Approach

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.

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 [15, 7, 5, 29, 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 all the three decoders and the shared encoder in detail.



(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 layer. (c) 3×3 convolutions have stride 2. (d) 3×3 convolutions have stride 1.

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, 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 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, 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 and the other for scene boundary detection. The decoders used for these tasks share the same architecture, which is shown in Fig 2c. The benefits of sharing the encoder between these tasks are discussed in Section 5.4.

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 [18] and used in [33]. After each convolution layer, batch-normalization followed by Leaky-Relu activation function are applied.

3.1.4 Up-Sampling Decoder

Up-sampling decoder, shown in Fig 2d, is used for next frame prediction. This decoder and the encoder together represents a ladder network [27], 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 [27], 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 [35].

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 $\mathcal{E}_1$ 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

$$\lambda_1 + \lambda_2 + \lambda_3 = 1 \quad (2)$$

and

$$\lambda_1, \lambda_2, \lambda_3, \lambda_4 \in \mathcal{R} \tag{3}$$

$\mathcal{L}^i_{total}$ 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.

$$\mathcal{L}^i_{d1}(\hat{s^i}, s^i) = -(s^i \log_2 \hat{s}^i_j + (1 - s^i) \log_2(1 - \hat{s}^i)) \tag{4}$$

where

$$\hat{s^i} = Sigmoid(\mathcal{D}_1(\mathcal{E}(v^i))) \tag{5}$$

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.

$$\mathcal{L}^i_{next\_frame}(\hat{G^{i+1}}, G^{i+1}) = \frac{\sum_{h,w}(G^{i+1}_{h,w} - \hat{G}^{i+1}_{h,w})^2}{h \times w} \tag{6}$$

$$\mathcal{L}^i_{d2}(B^{i+1}, \mathcal{L}_{next\_frame}) = -B^{i+1} \times \mathcal{L}_{next\_frame} \tag{7}$$

$$\hat{G}^{i+1} = Sigmoid(\mathcal{D}_2(\mathcal{E}(v^{i+1}))) \tag{8}$$

where 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.

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

$$\mathcal{L}^i_{d3}(\hat{B}^i, B^i) = -(B^i \log_2 \hat{B}^i + (1 - B^i) \log_2(1 - \hat{B}^i)) \tag{9}$$

$$\hat{B}^i = Sigmoid(\mathcal{D}_3(\mathcal{E}(v^i))) \tag{10}$$

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.

$$\mathcal{L}^i_{div}(\hat{s^i}, \mathcal{L}^{i-1}_{d2})) = -(\hat{s}^i \times \mathcal{L}^{i-1}_{next\_frame}) \tag{11}$$

3.3. Intuition behind MerryGoRoundNet

Harri Valpola's ladder network [34] 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. The loss term in equation 11 penalizes assigning a high score to a frame if that frame can be predicted with high confidence at previous timestep. 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).

Secondly, [24] 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 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. We ignore this loss term when the current frame, according to decoder $\mathcal{D}_3$, is the start of a new scene. Since network will not be able to predict next

frame when the current frame is the last frame of the ongoing video scene, optimizing the network for this loss at the end of a scene will be wrong.

Lastly, this architecture is well structured to adapt to various domains of videos. Fine-tuning and unsupervised pre-training [11] 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 [34], 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.

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. Decoder $\mathcal{D}_3$ predicts whether the current frame is the start of a new segment or not. The loss term in Equation 11 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, as determined by decoder $\mathcal{D}_3$. 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 the final summary will be complete and without abrupt scene changes.

5. Experiments

5.1. Datasets

MerryGoRoundNet was evaluated on TVSum [32] and SumMe [9] 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 [38, 19], instead of using the inference approach described in Section 4, key shot generation method explained in [38] is utilized. Evaluation is done using the keyshot based metric proposed in [38]. 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:

$$P = \frac{\text{overlapped time duration of A and B}}{\text{A's duration}} \tag{12}$$

$$R = \frac{\text{overlapped time duration of A and B}}{\text{B's duration}} \tag{13}$$

Their harmonic mean F-score, given by:

$$R = \frac{2P \times R}{P + R} \times 100 \tag{14}$$

is then used as the evaluation metric. The steps specified in [38, 19] 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 [11] before the supervised tasks. We trained it for 5 epochs, with an initial learning rate of 0.2 and momentum of 0.9. After 5 epochs, simultaneous

Dataset	Method	Supervised/Unsupervised/ SemiSupervised	F-Score
Training: 80% SumMe + YouTube Evaluation: 20% SumMe	vsLSTM [38]	supervised	37.6
	dppLSTM [38]	supervised	38.6
	SUM-GAN$_{dpp}$ [19]	supervised	39.1
	Gygli *et al*. [10]	supervised	39.7
	Zhang *et al*. [37]	supervised	40.9
	SUM-GAN$_{sup}$ [19]	supervised	41.7
	MerryGoRountNet(ours)	**semi-supervised**	**42.8**
	Li *et al*. [17]	supervised	43.1
	A-AVS [13]	supervised	43.9
	M-AVS [13]	supervised	44.4
Training: 80% TVSum + YouTube Evaluation: 20% TVSum	TVSum [32]	unsupervised	51.1
	SUM-GAN$_{dpp}$ [19]	unsupervised	51.7
	Li *et al*. [17]	supervised	52.7
	vsLSTM [38]	supervised	54.2
	dppLSTM [38]	supervised	54.7
	MerryGoRountNet(ours)	**semi-supervised**	**55.1**
	SUM-GAN$_{sup}$ [19]	supervised	56.3
	A-AVS [13]	supervised	59.4
	M-AVS [13]	supervised	61.0

Table 1: Performance comparison with state-of-the-art methods using F-Score evaluation metric.

Dataset	System	F-Score
SumMe	A	38.4
	B	37.2
	C	42.8
TVSum	A	49.7
	B	44.3
	C	55.1

Table 2: Comparison of multiple systems mentioned in ablation study

training of all the MerryGoRoundNet branches was initiated. The same learning rate was used for training the MerryGoRoundNet. Learning rate for decayed by a factor of 0.94 after every 1000^{th} iteration. 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. Values of λ_1, λ_2, λ_3, λ_4 were set to 0.5, 0.2, 0.3 and 0.65 respectively.

5.4. Ablation Study

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

- System A: Different Encoders for Video Summarization and Scene Start Detection
- System B: Shared Encoder for Video Summarization and Scene Start detection
- System C(MerryGoRoundNet) : System B with unsupervised next-frame prediction branch.

As can be seen from Table 2, sharing weights between shot boundary detection and frame importance prediction leads to better results. Moreover, System C's superiority over the other two systems prove the intuition behind the architecture of MerryGoRoundNet.

6. Results

We compare our semi-supervised approach with previously used approaches. Table 1 displays the performance of various approaches in terms of the 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

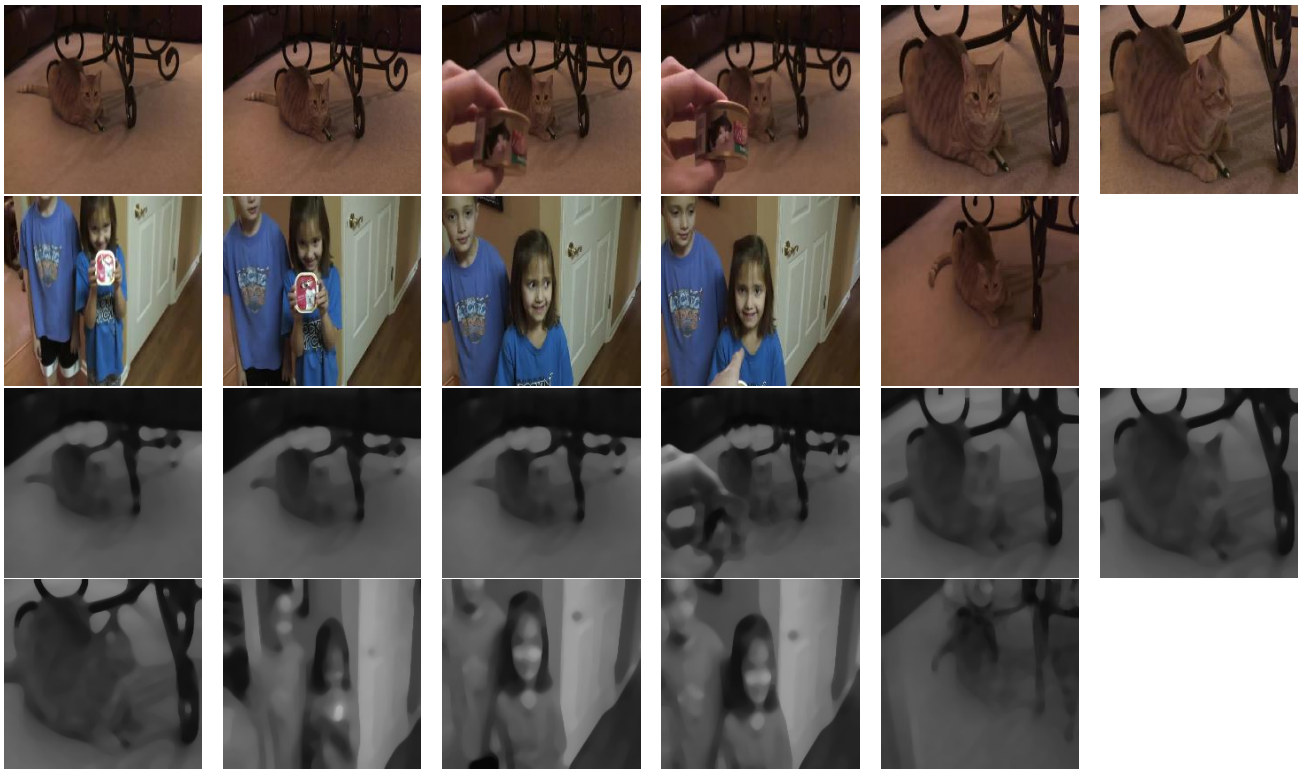


Figure 3: First two rows represent the summary and the next two represent the next-frame predicted for the frame just before the corresponding summary frame.

for the training of their systems, however, MerryGoRoundNet was trained only using Youtube and Summe/TVSum. Backed up by the results, MerryGoRoundNet justifies the intuition it was built upon, generating a continuous, diverse, crisp summary.

Figure 3 contains the summary and the 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 i.e. 90 frames were generated out of the 30 seconds video. Each output of the next-frame decoder corresponds to the frame generated relative to the summary frame. As can be seen from the figure, MerryGoRoundNet could not correctly predict the 6th frame of the summary, and hence included that frame in the summary. Moreover, the summary generated correlates highly to the human-generated summary. This highlights the fact that general behavior of a human while generating a summary is similar to the intuition behind the MerryGoRoundNet.

7. Conclusion

Our proposed work, MerryGoRoundNet, explores the application of convolutional LSTM for video summarization. Convolutional LSTM's are devised to take into account both the spatial and temporal relations among data, both of which are essential to capture to provide a meaningful summary. We further 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, but they also aid in generating a diverse and continuous summary, thereby achieving the objective of DPP augmented LSTM. 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 experiments showed that the final augmented MerryGoRoundNet performed better than [38, 19], and that additional tasks did help as expected in achieving a better summary.

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. Ridiculously fast shot boundary detection with fully convolutional neural networks. *arXiv preprint arXiv:1705.08214*, 2017.

[9] 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.

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

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

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

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

[14] 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.

[15] 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.

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

[17] 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.

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

[19] 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.

[20] J. Mas and G. Fernandez. Video shot boundary detection based on color histogram. *Notebook Papers TRECVID2003, Gaithersburg, Maryland, NIST*, 15, 2003.

[21] 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.

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

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

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

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

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

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

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

[29] 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.

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

[31] 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.

[32] 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.

[33] 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.

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

[35] 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.

[36] 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.

[37] 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.

[38] 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.

[39] 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.