Unsupervised video summarization using ITL-Autoencoder

Gijung Lee
University of Florida

***Abstract*— In a video, there are so many frames that are not important to see or check contents. These unimportant frames make us waste the time. We can solve this problem by detecting important objects in a video and making video time shorter automatically. Detecting objects is performed well in computer vision. However, this good performance is for not only important objects but also unimportant objects in the video. Detecting only important objects is a challenging problem in computer vision. If we can detect only important objects in a video, there will be many applications we can apply in various fields. For example, in underwater circumstances, we can check what is happening by recording a video. However, it is difficult to sort out which parts are important and unimportant in a video. Moreover, it is waste of time to check every frame in a long video to sort the important parts. To detect important parts in a frame, the autoencoder is used for this project. Using this model, we can extract the important parts in a frame and make a video time shorter which includes only import events in a video. We can apply this project in various fields. With an unsupervised approach, we have the advantage that there is no requirement for human annotations to learn the important event in a video. With this method, the evaluation shows that the process for video summarization has two summarized videos that are an important event and an unimportant event.**

Index Terms*—Video summarization, Saliency object detection, Unsupervised learning, K-means clustering.

I. Introduction

Video has become one of the most important forms of visual data. Due to the huge amount of video data and long play video time in each video, it is unrealistic for humans to watch these videos and identify useful information. However, the video summarization that I propose allows for generating a concise synopsis that conveys the important parts of the full-length video; based on this, viewers can have a quick overview of the whole story without having to watch the entire content. As you can see in Fig. 1, video summarization is the process of compacting a video having only important contents in the video. To do a video summarization, several approaches are suggested. In one of the video summarizations works, Barbieriet al. (2003) [1] sort out the coherent bibliography according to several ways of the summarization process, targeted scenario, the type of visual content, and the characteristics of the summarization approach. [2]

This work as presented as the project for EEL 6825 – Pattern Recognition and Intelligent Systems course

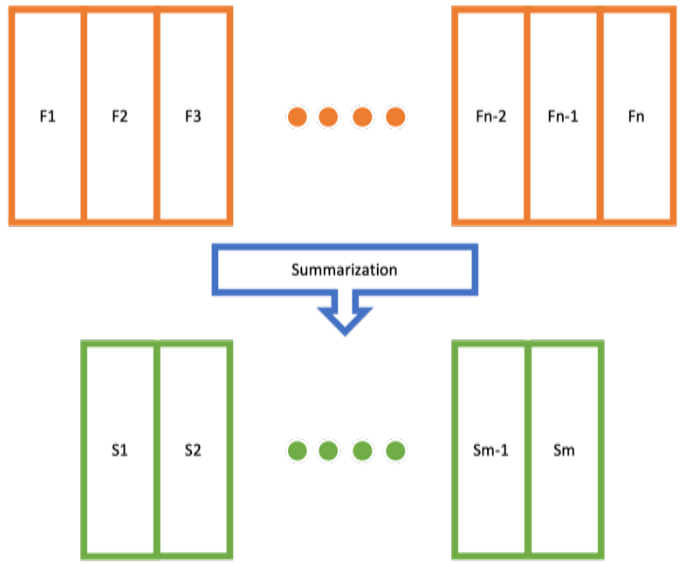

Fig. 1. The process of video summarization F: Frame, n: number of frames in a video, S: Summarized frame, m: number of frames in a summarized video, m is smaller than n.

In another early work, Li et al. (2006) [3] analyze the existing summarization approaches into utility-based methods that apply attention models to distinguish the salient objects and scenes and structure-based methods that build on the video shots and scenes. [2] Jiang et al. (2009) [4] suggest a few characteristic video summarization approaches, that involve the extraction of low-level visual features for calculating frame similarity or operating clustering-based key-frame selection. The motion descriptors are used to detect the main events of the video. The eigen-feature is used to identify the video structure. [2] After the introduction of deep learning algorithms, a comparison of the summarization performance shows that most deep-learning-based methods outperform the above methods [1][3][4] and stand for the state of the art in video summarization. I also apply deep learning-based methods for the project. For this project, in Fig. 2, there are four steps to generate a summarized video. The first step is to extract frames in a video. The second step is generating pseudo labels for the frames. For this step, I investigate Information-Theoretic Learning-Autoencoder (ITL-AE) [5] for better clustering to generate pseudo labels. The third step is to classify actual frames by comparing with reconstruction scores and pseudo labels. The fourth step is generating a summarized video.

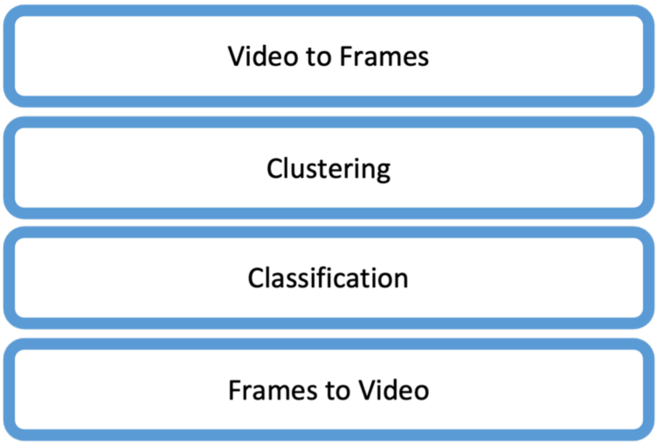

Fig. 2. Steps for the project.

The difficulty of video summarization is deciding and classifying important contents in the video. The frame that has important content is classified by a clustering method using the latent space in Autoencoder.

II. Related Works

A. Anomaly detection

Anomaly detection is one of the most challenging in computer vision. Recently, [6, 7] use deep learning-based autoencoders to learn the model of common behavior and apply reconstruction loss to detect anomalies. This project reaches to know common behaviors and anomalous behaviors for anomaly detection which can be used to sort out important and unimportant frames.

B. Autoencoder

An autoencoder is a type of deep neural network that is to learn lower-dimensional latent space and reconstructs input data. The encoder and decoder are concurrently trained. The encoder transforms the input data into latent space while the decoder reconstructs the input data from latent space. The loss function errors between the input data of the encoder and the output data of the decoder by comparing how well reconstructed the output data: $\mathcal{L} = |x - \hat{x}|$. The latent space is a denoised form of input data that helps classification. [5]

C. Information-Theoretic Learning (ITL) regularization

Using only reconstruction loss in a simple autoencoder can make biasing the network to learn only mapping data points to specific points on the latent space: i.e., the spatial structure of the latent space is meaningless. There are two problems with the bias. First, it will learn a discontinuous latent space when we sample an adjacent point to an encoded data point and pass it through the decoder (a meaningless representation). Second, it has the disadvantage of clustering because the distance between points does not mean similarity.

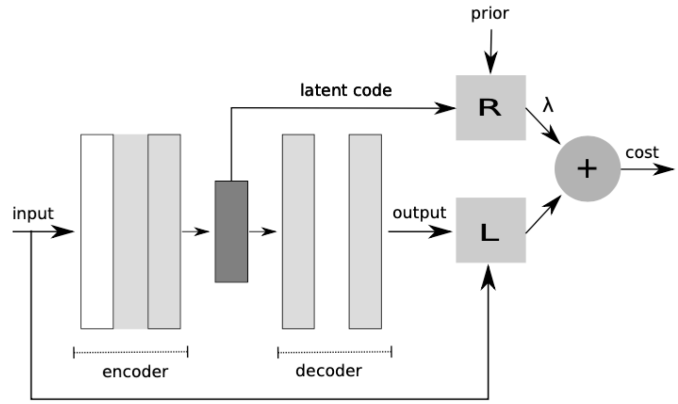

Fig. 3. Diagram for ITL autoencoder. L is the reconstruction cost function and R is the regularization that uses information-theoretic measures. [5]

To solve those issues, we apply the Information Theoretic Learning (ITL) regularization that calculates the Cauchy-Schwarz divergence (CSD) of the latent space with respect to a prior distribution. The reconstruction loss function has this divergence with some multiplier λ.

$$\mathcal{L} = |x - \hat{x}| + \lambda\, CSD(q(z|x)\,||\,p(z))$$

Where $q(z|x)$ is represented the encoder and $p(z)$ is represented an applied prior distribution on the latent space.

a. A parzen window for probability density estimation

To estimate the probability density function we use parzen window method. The estimation is made by centering a Gaussian Kernel $K_\sigma(x - x_i)$ with size σ at each data point x_i and summing the Gaussian Kernels to estimate the probability density function. The kernel size σ is the hyperparameter that we optimize for.

$$\hat{p}(x) = \frac{1}{N}\sum_{i=1}^{N} K_\sigma(x - x_i)$$

b. Renyi's 2nd order entropy and cross-entropy estimation

Renyi's entropy of a probability distribution $p(x)$ is an important method in Information Theoretic Learning. The second-order entropy is given by

$$H_2(X) = -\log \int p^2(x)dx$$

and calculated using the parzen window to be

$$\hat{H}_2(X) = -\log \frac{1}{N^2}\sum_j \sum_i K_{\sigma\sqrt{2}}(x_j - x_i) \equiv -\log V(X)$$

The cross-entropy between two distributions is calculated as

$$\hat{H}_2(X, Y) = -\log \frac{1}{N_X N_Y}\sum_j \sum_i K_{\sigma\sqrt{2}}(x_i - y_j) \equiv -\log V(X, Y)$$

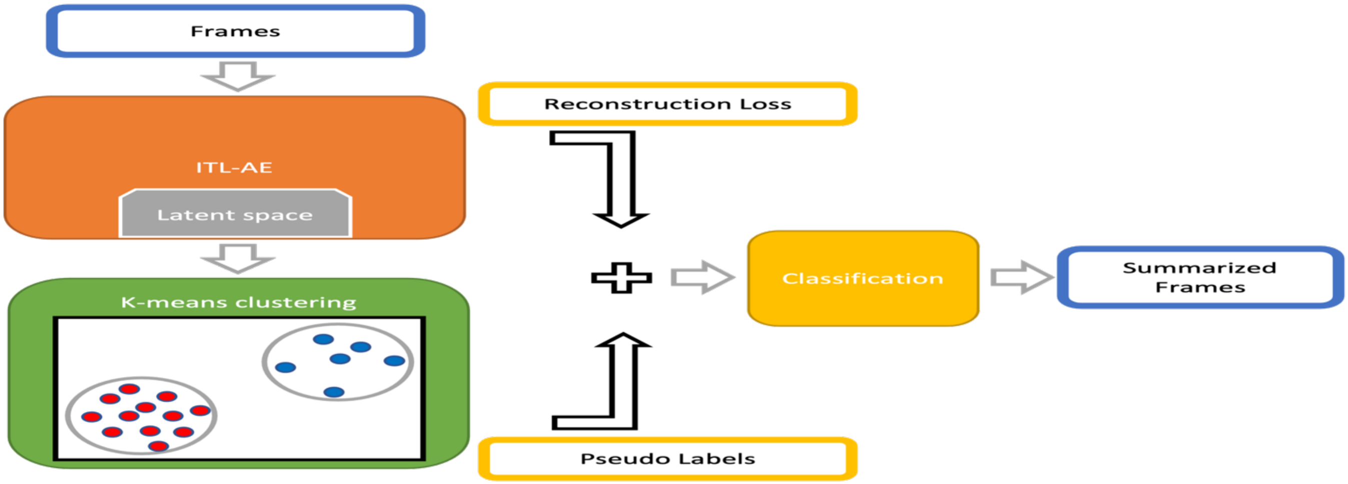

Fig. 4. Unsupervised Video Summarization Framework. 1. Training ITL-AE to get latent space. 2. Generating pseudo labels by clustering data points in latent space by K-means clustering. 3. Classification of the important frame and unimportant frame by using pseudo label and reconstruction loss.

Where $V_2(X)$ is the information potential and $V_2(X,Y)$ is the cross-information potential. In terms of the information potentials, the Cauchy Schwartz divergence is given by

$$CSD(p_X || p_Y) = \log \frac{V(X)V(Y)}{V^2(X,Y)}$$

We impose the model to minimize the information potential that makes samples from $p(x)$ to spread out and to maximize the cross-information potential that makes samples from $p(x)$ to move toward samples from $p(y)$. [5]

A. K-means clustering

The K-means clustering is an unsupervised clustering method. K-means clustering aims to cluster data points into separated subgroups by making each data point belonging to one of these subgroups. It aims to make data points in a cluster as similar as possible while maintaining clusters as distinct as possible. K-means clustering allocates the cluster's center so that the sum of the squared distances between the data points in the cluster is as little as possible. We use K-means clustering on the latent space to sort out frames (important and unimportant) into different clusters. Then we use these clustered labels as pseudo labels for classification.

III. Experiments

A. Dataset

For this project, I used the Brackish dataset [8]. The brackish dataset contains 89 videos are provided with annotations in the AAU Bounding Box, YOLO Darknet, and MS COCO formats. Fish are annotated in six coarse categories. Categories: Big fish, Small fish, Crab, Shrimp, Jellyfish, Starfish.

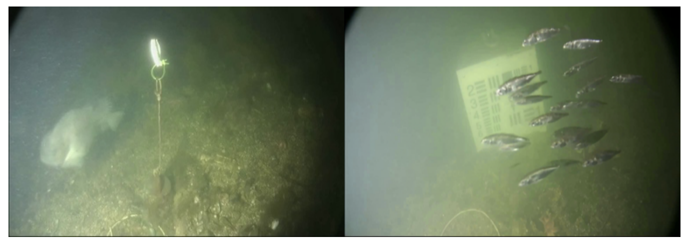

Fig. 5. Brackish data. Left: Big fish Right: Small fish

I used only fish class for this project. Individual video is used in the same category for video summarization. Frames that are extracted in a video are used for the video summarization process.

B. Frames

We extract frames in a video since not all videos have the same length and FPS, we define a parameter to adjust how many frames we want to extract and save per second. If a video of duration of 30 seconds, saves 10 frames per second = 300 frames are saved in total. For this project, we set the 15 frames per second which are the original video's FPS.

C. Latent space

In this part, I tried to generate pseudo labels by using the ITL-AE architecture described in the previous section. First, I trained autoencoders with 2-dimensional latent space. I used a Gaussian mixture distribution for the prior.

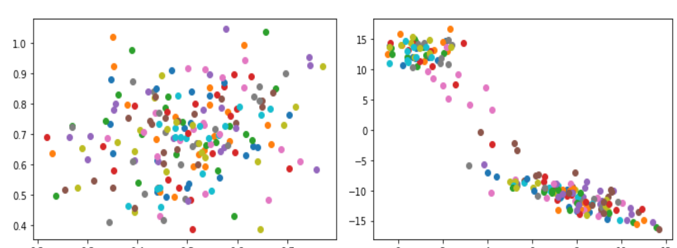

Fig. 6. Latent space. Left: without training Right: with training

As you see in the left image in Fig. 6., the input frames are spread out in the latent space while the right image shows that the input frames are clustered in the latent space. Using the ITL-AE gives latent space that makes frames be sorted out. With this method, we can generate pseudo labels that can be important frames and unimportant frames. Frames that have similar information are placed in one area in the latent space after training by ITL-AE.

D. Clustering

We use the K-means clustering method to generate pseudo labels by clustering data in the latent space. In this case, label “0” means an important frame that includes fish, and label “1” means an unimportant frame that mostly includes background.

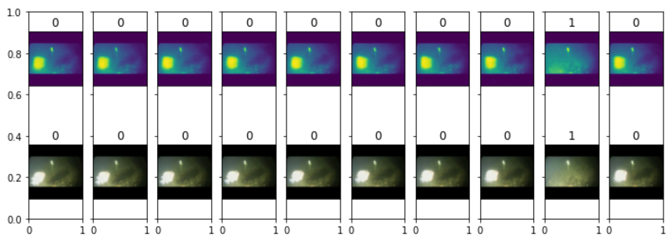

Fig. 7. Proper pseudo labels with images.

When I trained ITL-AE with epochs 200, as you can see in Fig. 7., I could get proper pseudo labels.

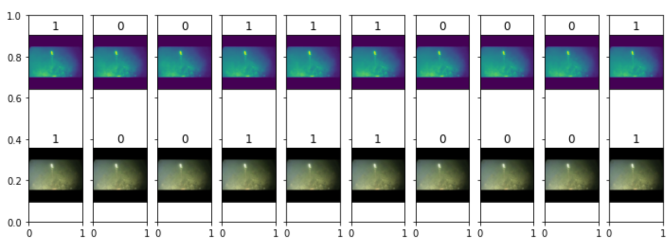

Fig. 8. Improper pseudo labels with images.

However, as you can see in Fig. 8., some frames have the wrong label. To solve this problem, we approach anomaly detection by reconstruction loss.

E. Reconstruction Loss

This time we apply reconstruction loss to find anomaly behaviors in frames.

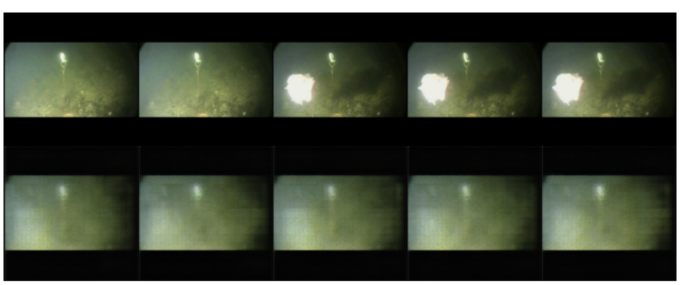

Fig. 9. Images after train model with epochs 10. Top: original frames. Bottom: generated frames by model.

As you can see in Fig. 9., the model with a few epochs of training generates frames that have common behaviors in a frame. The frames that have only background are calculated to get less reconstruction loss values while the frames that have fish are calculated to get greater loss values.

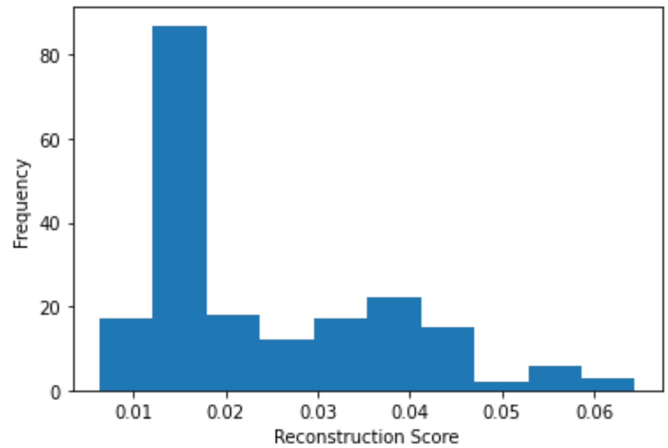

Fig. 10. Histogram of reconstruction score.

As you can see in Fig. 10., frames that are over 0.02 in reconstruction score are abnormal. We can decide that a frame that has a reconstruction loss value of more than 0.02 is an important frame that has fish in this case. Specifically, we can set the threshold by $threshold = M + S - 0.01$. (M: mean of reconstruction scores, S: standard deviation of reconstruction scores)

F. Classification

Finally, we can get only important frames that have a fish in a frame by applying both methods pseudo labels and reconstruction loss. The frames that are unproperly labeled are solved combine the reconstruction loss. The frames mislabeled as “0” are checked with the threshold of the reconstruction score. Only the frames that are over the threshold are classified as label “0”.

IV. RESULTS

After I performed every step of the video summarization frameworks, I could get proper results. I used the Brackish dataset and accuracy score (It represents the ratio of the sum of true positives and true negatives out of all the predictions). I visualized the results by a confusion matrix.

Methods	Accuracy
PL	65.33%
PL + RL	87.44%

Table 1. Results of video summarization. PL: Pseudo labels, RL: Reconstruction loss

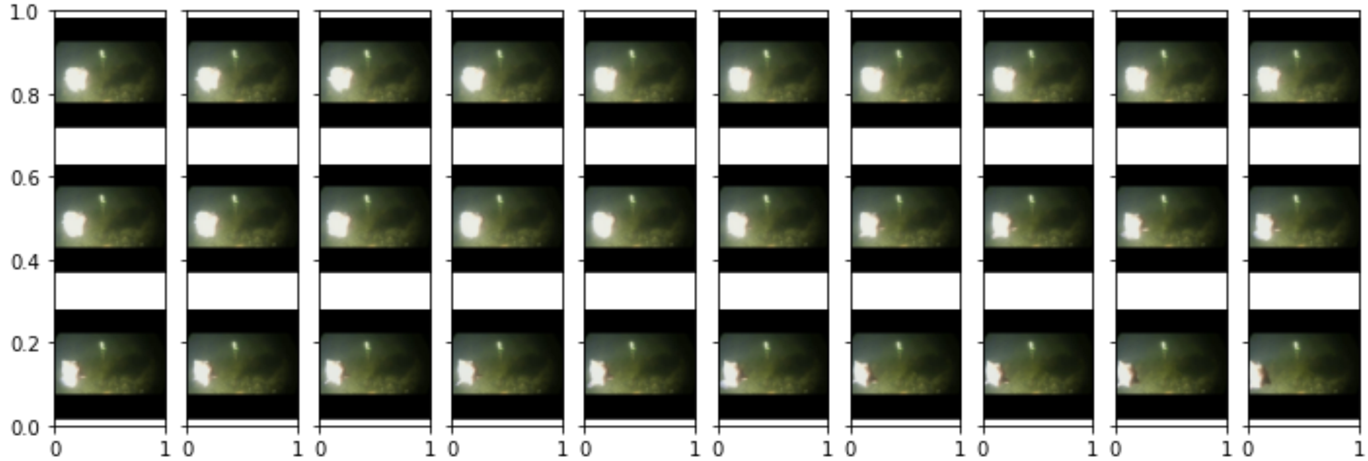

Fig. 11. Summarized Frames.

As you can see the table 1, we can get 87.44% accuracy with both pseudo labels and reconstruction loss methods. In table 1, we can get only 65.33% with only pseudo labels. We can check the reconstruction loss helps to improve the performance by solving the pseudo labels method's problem.

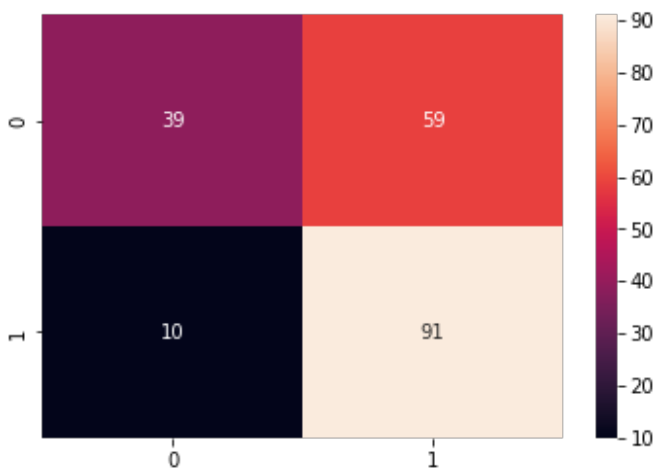

Fig. 12. Confusion matrix using the pseudo labels method.

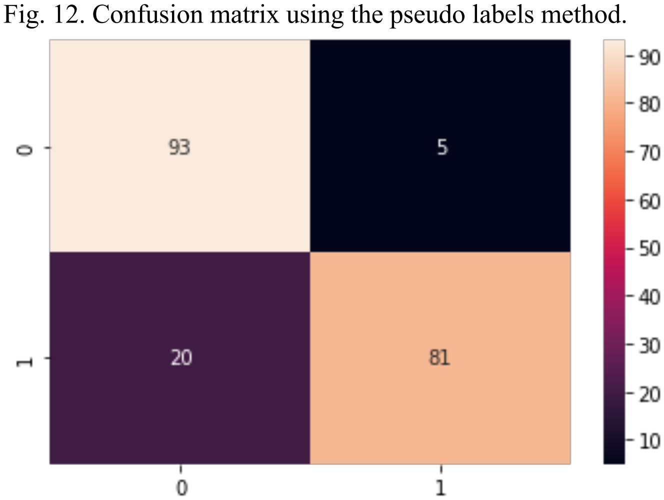

Fig. 13. Confusion matrix using the pseudo labels and reconstruction loss methods.

I. Conclusion

The video summarization task is a challenge if we do it manually. It wastes our time a lot. With my proposed method, we can get proper a video that has shorter playtime without human efforts. We can easily summarize the video that includes only important frames. With this framework, there are many applications that we can apply in various fields. We could check only one (pseudo labels) method cannot affect video summarization. We can harmony with other methods (i.e., reconstruction loss) to get a better result. We get 87.44% accuracy. We have future work to get a summarized video that fully has important frames.

Acknowledgment

I would like to express special thanks to Dr. Wu for his wonderful class.

References

[1] M. Barbieri, L. Agnihotri, and N. Dimitrova, "Video summarization: methods and landscape," in *Internet Multimedia Management Systems IV*, J. R. Smith, S. Panchanathan, and T. Zhang, Eds., vol. 5242, International Society for Optics and Photonics. SPIE, 2003, pp. 1 – 13.

[2] Apostolidis, E., Adamantidou, E., Metsai, A. I., Mezaris, V., & Patras, I. (2021). Video summarization using Deep Neural Networks: A survey. *Proceedings of the IEEE*, *109*(11), 1838–1863.

[3] Ying Li, Shih-Hung Lee, Chia-Hung Yeh, and C.-J. Kuo, "Techniques for movie content analysis and skimming: tutorial and overview on video abstraction techniques," *IEEE Signal Processing Magazine*, vol. 23, no. 2, pp. 79–89, 2006.

[4] R. M. Jiang, A. H. Sadka, and D. Crookes, *Advances in Video Summarization and Skimming*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 27–50.

[5] Santana, E., Emigh, M., & Principe, J. C. (2016). Information theoretic-learning auto-encoder. *2016 International Joint Conference on Neural Networks (IJCNN)*.

[6] D. Xu, E. Ricci, Y. Yan, J. Song, and N. Sebe. Learning deep representations of appearance and motion for anomalous event detection. In *BMVC*, 2015.

[7] M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis. Learning temporal regularity in video sequences. In *CVPR*, June 2016.

[8] Pedersen, M.; HAurum, J.B.; Gade, R.; Moeslund, T.B.; Madsen, N. Detection of Marine Animals in a New Underwater Dataset with Varying Visibility. In Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Long Beach, CA, USA, 15–20 June 2019.