Towards Visual Explanations for Movie Scenario Segmentation

Linning Xu
The Chinese University of Hong Kong
1155128851@link.cuhk.edu.hk

Abstract

In this paper, we revisit the class activation mapping (CAM) techniques [7], and shed light on how it help in understanding movie scenario segmentation task. We first propose the adaptive threshold strategy and several modified implementations of CAM derived from its vanilla version that are suitable for a wide range of visual explanation needs. Using a collected data set consisting of 30 movies with labeled scenario segmentations, we apply CAM to visualize the discriminative regions in the input sequential shots that determine whether there exist a scenario transition between the consecutive two shots. The experimental results demonstrate that our extended version of CAM provides novel perspective to understand movie scenario segmentation task, and serves as an indicator of the potential limitations of our current scenario segmentation approaches in order to bring future improvements.

1 Introduction

With the on-growing popularity of movie industry worldwide, there is growing interest in Deep Learning to analyze movie systematically. From images, short clips to long videos, movie provides a rich source of information and knowledge to analyze, yet has very limited exploration in the research area. *Movie Scenario Segmentation* aims to cut a movie into several *scenarios*. Compared to *frame* or *shot*, scenario contains high-level semantics elements, thus could be viewed as the basic functional unit in story-telling of a movie, where a wide range of high-level movie analysis tasks could be based on, such as the character recognition, human interaction graph generation, scenario retrieval, etc. Previous studies [1, 3] mainly focused on short video analysis considering the lack of large long video data sets. Several unsupervised [4, 5, 6] methods have been proposed for scenario segmentation tasks. It is noticed that existing scenario detection methods using low-level visual cues fail to capture high-level semantics in movie scenarios.

Movie Scenario Segmentation task is modeled as a binary classification problem in our experiments, where we use 1 to indicate a scenario transition. In our pilot studies, several networks have been designed to do this task and expect future improvements. To help understand the deep features in our movie scenario segmentation task, we propose to apply Class Activation Maps (CAM) for visual explanations. CAM is a powerful tool for identifying the class-discriminative regions with the merit of global average pooling, and the most activated regions are captured by the extracted weights from the final layer and displayed on the original image. The contribution of this paper mainly lies in two aspects. Firstly, we propose several adaptions of vanilla CAM for more informative visual explanations in static images. Then we extend its use to movie settings where an additional max pooling layer is crafted to tackle the context information in the sequential shot images. The experimental results demonstrate that our CAM is helpful in 1) finding out the informative regions in the sequential shot images in determining the existence of scenario transitions, and 2) interpreting the failure cases in our current segmentation results, shedding light on the potential directions for future improvements.

Paper Report of Final Project for ELEG5491 - Introduction to Deep Learning.
Copyright by author(s) 2019. The Chinese University of Hong Kong.

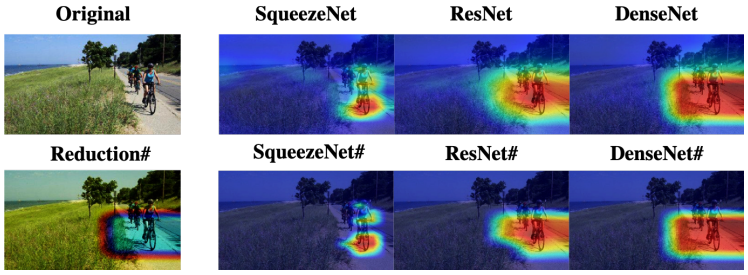

Figure 1: Illustration of different modifications of CAM for a *mountain bike* image. The classification task is based on ImageNet and all of the adopted models give Top 1 ranking to the class *mountain bike, all-terrain bike, off-roader* for the chosen image. Different shapes of the activated regions are captured by different network structures. The output images on the first row are the vanilla version CAM, showing several vague highlighted areas. The images on the second row indicated by hash tags(#) adopt *cutting-off* strategies, displaying much more compact activation regions that eliminate noisy non-informative regions. While most visualization techniques adopt *addition* to mask CAM on the raw images, it is also reasonable to use *reduction* in order to retain the original color display, as shown in the left bottom image.

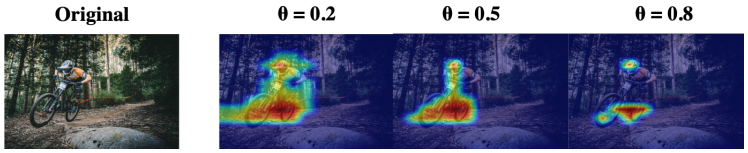

Figure 2: The distillation of the activation regions with increasing thresholds. It is noticed that only the *helmet* and the *bike* parts are preserved, where the non-relevant human figure for identifying class*mountain bike* is discarded. This kind of evolution may be interpreted as an alternative way for hierarchical feature representation for information extraction.

2 Adaptive Threshold Strategy for CAM

For a given image, let $f_k(x, y)$ represents the activation of unit k in the last convolutional layer at spatial location (x, y). According to [7], the class activation map M_c for class c is defined as,

$$M_c(x, y) = \sum_k w_k^c f_k(x, y), \tag{1}$$

which is a weighted linear sum of the presence of these visual patterns at different spatial locations. The activation map values need to be clamped into the range $(0, 255)$ in order to display RGB color properly, denoted as $M_c^*(x, y)$, and the output is given as below,

$$O_c(x, y) = \alpha M_c^*(x, y) + \beta R(x, y), \tag{2}$$

with default value $\alpha = \beta = 0.5$. However, it is noticed that this plain implementation could only give limited visual explanations for the classification task, since the activation region tends to be spread across pixels (see the first row in Figure 1) and was originally proposed for localization tasks. In order to get more informative visual explanations from CAM, we propose to set an adaptive threshold θ to cut-off the noisy activation regions. The $M_c^*(x, y)$ is updated as below,

$$M_c^*(x, y) = \begin{cases} M_c^*(x, y) & \text{if } M_c^*(x, y) > \theta, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

The merit of this method lies in its flexibility of choosing a θ that is task-oriented and could be tailored for any desired accuracy, with the minimum coverage on the less informative regions. A comparison between several versions of CAM is given in Figure 1. Sometimes it is desirable to see the distillation of information from a image, which could be achieved by choosing an increasing sequence of thresholds θs, as demonstrated in Figure 2.

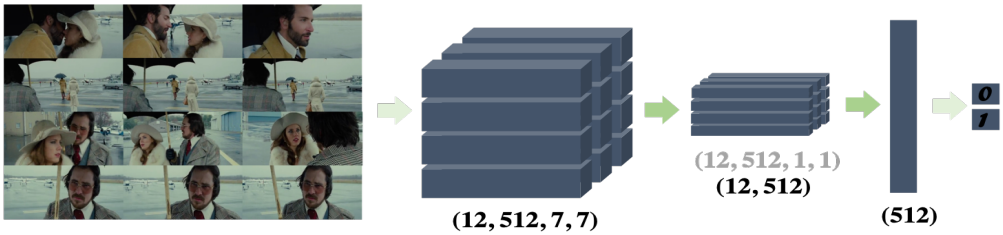

Figure 3: Network architecture adapted to CAM, using inputs from *American Hustle (2013)* . Two Max-pooling layers (indicated by green arrows) are adopted to extract out corresponding weights for classification. 0 and 1 stand for non-transition and transition respectively.

3 CAM on Movie Scenario Segmentation

3.1 Extension of CAM on the additional dimension

In order to visualize the informative regions in our movie scenario segmentation setting, we modified the original network structures and made it suitable for CAM, while more complicated networks and feature designs are adopted in the previous pure classification task. Part of the modified network architecture is illustrated in Figure 3. In movie setting, CAM is extended to an additional dimension, where a second max pooling is added to obtain the weights in the contextual information from dimension z. Assume there are $Z = 12$ shot images in sequence for single input, and $F = 512$ features are obtained after feeding into the backbone ResNet [2]. For each feature, we record the index z from 12 images that obtains the maximum value. For each image z, we count the total number of features C_z that achieve maximum values, and assign the weight $w_z = C_z/F$ to each image z. The overall calculation is modified as below,

$$M_c(x, y, z) = \sum_z w_z \sum_k w_k^c f_k(x, y, z). \tag{4}$$

3.2 Implementation Details

Our data set contains 30 movies, made of the train, validation and test sets each consists of 10 different movies. There are approximately 12000 cleaned annotations in each set after data pre-processing. As illustrated in Figure 3, the input is a sequence of 12 shot images from consecutive 4 shots, each with 3 frames. Our goal is to train a model that could automatically decide whether there is a scenario transition happens between the 2^{nd} and 3^{rd} shots in the selected window. The output is a binary number where 1 indicates the scenario transition and the two parts should be split apart. To remedy the severe data imbalance problem that 0 is dominated in the data set, the adaptive threshold is set as $\theta = 0.75$ and the loss function is $Loss(x_i) = -\alpha_1 Loss(x_i, class_0) - \alpha_2 Loss(x_i, class_1)$, a weighted between the two classes with α_1/α_2 = 1/10 considering the data proportion. To preserve the interpretability of 2d images, we keep the 7×7 feature maps with 512 channels from the output of the average pooling layer from ResNet18 [2], followed by the first max pooling layer extracting out the most activated region with kernel size 7×7. The second max pooling layer is applied to dimension z cross each channel, with kernel size 12×1.

There typically exits a trade-off between accuracy and interpretability. While CAM is easy for visual explanations, the modified network structure and the feature transformations result in the reduced classification accuracy to some extend. We use mAP as our evaluation metric. Let AP_k denotes the Average Precision (AP) of the kth movie in test set with total size K=10. Then the *mean Average Precision (mAP)* is evaluated as $mAP = \frac{1}{K}\sum_k AP_k$. While in the pure classification task where the cosine similarities between the first and the second part of the input sequential images have been incorporated as an important feature lead to the final mAP = 0.45 on test data set, our the CAM version only achieve test mAP = 0.39.

3.3 Result Analysis

Some special properties of the visualized CAMs are concluded as below, observed from Figure 4.

Figure 4: Activated maps using modified CAM, where the ground truths and the predictions are denoted as GT and PD respectively, showing both the successful and failure cases. The first and second rows are scenarios from *Paranorman (2012)*, the third and fourth rows are scenarios from *Shame (2011)*

***Spot the Difference* game.** In predictions of 0s, it is noticed that the activated maps cross images tends to have same shapes. In this sense, the scenario segmentation task could be viewed as a *Spot the Difference* game, asking our designed model whether there exist any difference (in space layout or tone of the color) between the corresponding regions in the input images.

Background-oriented and hue sensitive. The distribution of the activated regions seem to be spread out, less object-oriented and less class-discriminative. While the CAM that is applied on ImageNet with 1000 classes can detect objects accurately, our CAM mainly highlight on the background information, capturing the overall space alignments and the color tones from the shot images. The maps are hue sensitive, even small changes in hue and brightness that are hard be detected by human eyes would be exaggerated. These properties are responsible for many failure cases in our segementation task. As shown in fourth row in Figure 4, the consecutive shots have very similar space alignments and color tones, thus our model predict it as a 0 (non-transition), and the trivial content changes in the shots are hard to be detected by our current model.

One noticeable limitation of our current model indicated from the visualized result is the random transformations of input shot images. In traditional object classifications, we focus on object detection. As long as the object appears in any part of the image, the objects would be detected. Also, the shape of objects is the main focus while hues and colors are not that important. However, as inferred from our experimental results, movie scenario segmentation emphasizes the overall looking of the input shot images, capturing the trivial changes in color tones. Therefore, random crops and traditional image transformation may result in great information loss in our task.

4 Conclusion

In future works, it is desirable to keep as much information in original input with special treatment to the raw shot images. It is also expected to get more informative annotations from the collected data set by considering enlarging the classification categories in order to obtain more class-discriminative regions. More crafty features could be designed and incorporated with insights obtained from the observed special properties to further improve the segmentation performance.

References

[1] B. G. Fabian Caba Heilbron, Victor Escorcia and J. C. Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–970, 2015.

[2] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[3] M. Monfort, A. Andonian, B. Zhou, K. Ramakrishnan, S. A. Bargal, Y. Yan, L. Brown, Q. Fan, D. Gutfreund, C. Vondrick, et al. Moments in time dataset: one million videos for event understanding. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[4] Z. Rasheed and M. Shah. Scene detection in hollywood movies and tv shows. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, volume 2, pages II–343. IEEE, 2003.

[5] Y. Rui, T. S. Huang, and S. Mehrotra. Exploring video structure beyond the shots. In *Proceedings. IEEE International Conference on Multimedia Computing and Systems (Cat. No. 98TB100241)*, pages 237–240. IEEE, 1998.

[6] P. Sidiropoulos, V. Mezaris, I. Kompatsiaris, H. Meinedo, M. Bugalho, and I. Trancoso. Temporal video segmentation to scenes using high-level audiovisual features. *IEEE Transactions on Circuits and Systems for Video Technology*, 21(8):1163–1177, 2011.

[7] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.