Hybrid Approach for Logo Segmentation in Videos

Quentin Monnier*† *quentin.monnier@b-com.com*,
Tania Pouli* *taniapouli@gmail.com*,
Kidiyo Kpalma† *kidiyo.kpalma@insa-rennes.fr*,
Nicolas Ramin* *nicolas.ramin@b-com.com*

*IRT b<>com, 1219 Av. des Champs Blancs, Cesson-Sévigné, France
† Univ Rennes, INSA Rennes, CNRS, IETR-UMR 6164, Rennes, France

***Abstract*—This work focuses on the segmentation of fixed overlay logos in videos. Although state-of-the-art video semantic segmentation models achieve good results, they often rely on complex architectures with high memory and computational costs. These constraints are even more critical in video processing, where models must handle temporal consistency and efficiently process multiple frames. However, applications such as discriminating overlay content from scene elements have specific constraints and exploitable priors that make generalist models less suitable. We present a hybrid approach that addresses this task by combining the adaptability of deep learning with handcrafted spatio-temporal features. This hybrid architecture outperforms traditional models that process frames directly or rely solely on isolated cues, while maintaining competitive inference times. Extensive experiments confirm its effectiveness. Our code is available at https://gitlab.insa-rennes.fr/qmonnier/hybrid-approach-for-logo-segmentation-in-videos/.**

***Index Terms*—video semantic segmentation, overlay logo detection, hybrid deep learning, spatio-temporal features**

I. Introduction

Video content consumption constantly increases, both online and through traditional broadcast. Most content can be considered as "scene elements" as it is captured by a physical or virtual camera. On the other hand, content can be edited before distribution by superimposing "overlay" elements such as logos, graphics or text. Both types of content are different in nature, have different visual properties (textures, colors, shapes, motion), and serve different purposes.

Often, raw footage prior to the addition of overlay elements is not archived, meaning that later separation of the two types of content is no longer possible. Nevertheless, it is common for archived content to be reused later and to undergo further transformations (conversion to HDR, upscaling, compression, etc). Since different types of content do not respond to transformations in the same way and serve different purposes, one might want to re-separate them to process them differently.

This can be considered as a semantic segmentation task, which is a well-studied area. This field is dominated by deep learning approaches, which tend to address general tasks and thus require complex architectures. Applying such methods to video content leads to new constraints that often further increase complexity and execution time.

However, segmenting fixed graphical elements is a more constrained task. Features such as shape, position, texture, and motion might be useful in guiding an otherwise generic semantic segmentation model. In this work, we explore this hypothesis by proposing a hybrid approach where a set of carefully selected features capturing spatio-temporal information in a compact manner are provided to a segmentation model instead of the raw video frames. We show that this approach can outperform state-of-the-art models in this specific segmentation task, while offering a more compact architecture.

Existing methods for detecting and segmenting fixed overlay graphical elements in videos are discussed in Section II. We describe the features used as input and the subsequent neural network architecture in Section III. Finally, we evaluate the performance of the proposed hybrid approach relative to a traditional neural architecture, both in terms of accuracy and execution time.

II. Background

The goal of image segmentation is to assign a label to each input pixel depending on its content. "Object" or "Instance" segmentation aims to distinguish different object occurrences, while "semantic" segmentation labels their semantic nature (panoptic segmentation is the grouping of the two). Tasks are also classified according to the data available for learning (supervised, weakly supervised, unsupervised) or the degree of user interaction required for inference to produce a result (interactive, semi-interactive, automatic). Video segmentation extends the image segmentation problem by introducing new constraints in terms of temporal consistency and computational complexity [1]. In this paper, we propose an automatic method to perform a binary (scene element vs overlay) semantic segmentation task on videos.

Existing logo detection methods have been designed for various goals. Logo detection in text documents focuses on analyzing shapes in black-and-white images rather than spatio-temporal features [2]–[17]. Several works aim to identify known logos from a set [2], [9], [12]–[25]. In these cases, the goal is not to learn to distinguish logos from other content but to recognize specific patterns in possibly degraded media. Moreover, certain methods determine the rough position of a logo with a bounding box, which is a less demanding task than pixel-wise segmentation [2]–[17], [19], [20], [24]–[26].

In [27], a more precise segmentation of fixed graphical elements is proposed, where logo detection is performed by thresholding the differences between consecutive frames over the duration of the video. If no logo is detected, logo segmentation is performed patch-wise using a Bayesian classifier coupled with a neural network. In [28], logos are detected

by thresholding the sum of differences between consecutive frames within a time window. Outliers are then removed by applying a maximum a posteriori model to the spatio-temporal neighborhood of each pixel. Pixel-wise temporal variation can also be inferred from the difference between the minimum and maximum luminances in a time window, as is done for each video corner in [29]. Gradients can be used in different ways to provide additional spatial and chromatic information: they can be filtered to keep those that maintain a similar position and direction over time [30], or averaged along the time dimension to create a map used as a cue for logo detection [31], [32]. Alternatively, the 3D histograms of the YCbCr components of the video corners can be used to separate the colors of the logo from those of the background [33].

These works show that features such as pixel-wise temporal variations, persistent gradients, or logo appearance and color are effective cues for logo detection. However, these cues are not used simultaneously and spatial and temporal information is not considered together. More importantly, existing methods often rely on arbitrary thresholds to make their decisions, making them less flexible. The time window durations used by the aforementioned methods can also limit their potential applications, since they cause a greater delay when the currently processed frame is centered.

In contrast to our application, in logo removal tasks, false positives (falsely identified as logos) or imprecise masks are less critical because the inpainting step corrects these errors. In addition, most of these methods focus only on the corners of the frames. In contrast, we focus on applications where mask precision is crucial.

Although not specifically focused on the segmentation of fixed graphical elements, recent advances in deep learning have led to significant improvements in semantic segmentation for video. Unlike previous convolutional architectures, new methods that integrate transformers and attention mechanisms have become the new state of the art. One example is TMANet [34], a model that ranked first on both the cityscapes [35] and camvid [36] video segmentation benchmarks on paperwithcode: it uses an attention module to retain temporal memory without having to be fed multiple frames at a time. However, these methods are complex, which comes at the expense of inference time, ease of learning (including the amount of training data required), and compatibility of these models with frugal or embedded systems.

Inspired by both types of approach, in the next section, we describe our hybrid method that aims at finding a trade-off between efficiency and complexity with respect to the task of scene element and overlay content segmentation.

III. Proposed Method

The key idea of our approach is to take advantage of the specific features typically present in fixed graphical elements, which distinguish them from the underlying video content, and combine them with the flexibility and predictive power of CNN-based semantic segmentation. Our hybrid approach pre-

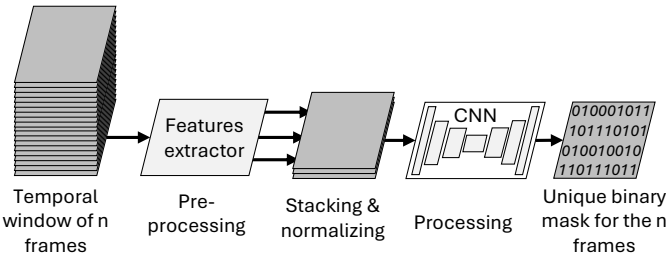

Fig. 1. Overview of the steps to calculate the mask of a video sequence

computes a set of hand-crafted features that serve as input to a neural network performing the segmentation.

To compute the segmentation of a video V, our method sequentially takes as input a temporal window of n frames (typically $n = 20$) sampled at a rate of r. The input tensor of size $h_v \times w_v \times c_v \times n$ (where h_v, w_v, c_v are the height, width, and number of channels of each frame) is fed into a pre-processing module that computes spatiotemporal features (see Section III-A) in the form of feature maps of the same spatial dimensions as the input frames.

As shown in Figure 1, the computed feature maps are then stacked into a single tensor, normalized to zero mean and unit standard deviation. The tensor is then fed to a small 2D encoder-decoder CNN, which is trained to generate a binary mask of any fixed graphic element in the sequence, based on the features described in Section III-A.

A. Spatio-temporal Feature Computation

Fixed overlay elements present certain characteristics that are easy to identify for a human observer, both in terms of their spatial and temporal aspects. Existing semantic segmentation networks target a wide variety of object categories and thus consider features that are not necessarily useful in this particular case. Even if such approaches are fine-tuned for this specific task, the architectures remain large and computationally intensive. Instead, we propose to use a set of "hand-crafted" feature maps, computed from a number of consecutive frames, as input to the segmentation network.

Based on previous work and our own experiments, we identified three main types of features that can be used by segmentation techniques to perform the the discrimination between the two contents: color cues, temporal variability cues, and shape cues. However, each cue family can be used in many different ways and implementations. For example, persistent colors can be retrieved by computing the temporal mean of the frames as well as their median. Temporal variability can be computed using the standard deviation of each pixel or the difference between its minimum and maximum luminance values over time. The shape information is included in the features just described, but our experiments showed that creating specific features for this cue yields better results. What seems to condense shape information best are spatiotemporal edges. To gather them into a single layer map, one can use the mean, multiplication, or median of the frames edges along the temporal dimension, use the "generalized gradient method"

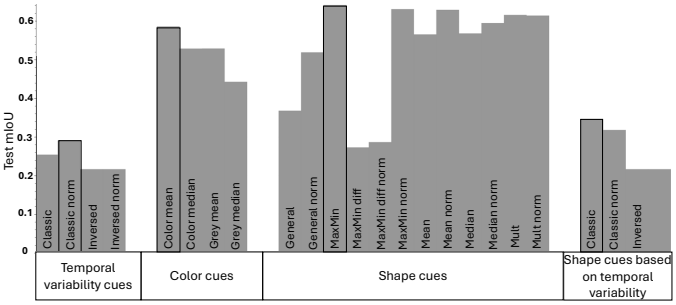

Fig. 2. A comparison of the features implementations that proved to be the most effective when used separately (based on the logo mIoU score). Features used in combination experiments are outlined in bold.

[30], or directly compute the edges of the temporal variation map discussed earlier. Finally, for each type of cue, some features might be better interpreted when inverted or instance-wise normalized, while others might not.

Since each cue type has multiple implementation variants, testing all combinations (with multiple runs for stability) would exceed our available time and GPU resources. To identify the best feature set, we did an ablation study by first training networks (see Section III-C) on individual variants of each feature family to determine the most effective implementation. As inference time showed little correlation with implementation choices, we selected the best variants based on mIoU. Figure 2 presents the results, and the retained implementations are detailed below.

1) Mean of the Frames: Logos are often designed to be easily distinguishable from the rest of the content visually, for example by using specific shapes and colors. To obtain a compact representation capturing this information, we consider a channel-wise pixel average of the frames in the sequence, which retains the shapes and colors of motionless areas. Formally, let V_k be the kth frame from the temporal window. $V_{k,i,j,l}$ is therefore the pixel located at (i, j) in channel l. The mean of the frames (MF) is therefore determined by:

$$MF_{i,j,l} = \frac{\sum_{k=0}^{n-1} V_{i,j,k,l}}{n} \tag{1}$$

2) Temporal Variation: When a video contains a fixed logo, its location is generally subject to little variation compared to other objects in the scene. Detecting the degree of temporal variation of each pixel is therefore a good cue for segmenting fixed logos. With f_{norm} being the min-max feature scaling (0 to 1), we compute the temporal variation map (TV) using:

$$TV_{i,j} = f_{norm}\left(\max_l\left(\max_k\left(V_{i,j,k,l}\right) - \min_k\left(V_{i,j,k,l}\right)\right)\right) \tag{2}$$

Gradients are the first derivatives of the signal. They are easy to compute and are used to detect the boundaries between contrasting objects in the same scene. For the problem of detecting fixed logos in video, gradients can be exploited in different ways. Gradients that remain fixed over time can indicate the presence of a logo. Thus, we compute $G\left(V_{i,j,k,l}\right)$ for each frame of the sequence using:

$$G\left(V_{i,j,k,l}\right) = \max_l\left(\sqrt{\left(\frac{\partial}{\partial i}V_{i,j,k,l}\right)^2 + \left(\frac{\partial}{\partial j}V_{i,j,k,l}\right)^2}\right) \tag{3}$$

Then, we can find the maximum and minimum value at each location, which yields a two layers feature map, called "spatial edges" (SE), and described by:

$$SE_{i,j} = \left\{\max_k\left(G\left(V_{i,j,k,l}\right)\right), \min_k\left(G\left(V_{i,j,k,l}\right)\right)\right\} \tag{4}$$

Finally, gradients can also extract shape information from the temporal variability map described above. Differences in temporal variability between different regions of the map give rise to edges that are complementary to those that appear directly in the frames. This temporal variability edges (TE) map can then be combined into a "spatiotemporal edges" (STE) map using:

$$STE_{i,j} = \left(1 + ME_{i,j}\right) \times \left(1 + G\left(TV_{i,j}\right)\right) - 1 \tag{5}$$

In this way, common edges between spatial and temporal edges are highlighted, without the edges detected with just one of the two methods being completely overlooked.

B. Dataset Creation

There is no public dataset to our knowledge for logo segmentation in videos, but existing logo recognition datasets provide images of brand logos. We create our own dataset by inserting these logos into unedited videos. Some datasets contain logos directly included in scenes (e.g., on soda bottles), which we avoid. Instead, we use databases available on www.kaggle.com with PNG logos with transparent backgrounds, allowing for easy masking via the alpha channel.

To ensure diversity and avoid over-fitting to specific brands, we filter out repetitive logos based on names and labels, leaving 1206 unique logos. We then obtain different raw videos from [37]–[40], ensuring sufficient variation and no pre-existing graphical elements (logos, subtitles, borders). A histogram comparison step was included to avoid scenes with similar characteristics.

To generate the dataset, we randomly paired videos with logos (80% of the time), resizing and positioning the logos within the frames before re-encoding. This produced 1480 unique sequences, split into train/validation/test sets, in which both scene elements and overlay elements are distinct. The dataset is available at https://gitlab.insa-rennes.fr/qmonnier/logos-segmentation-dataset/.

C. Network Architecture and Training

As described in Fig. 3, the neural network is a small encoder-decoder divided into "blocks", each consisting of two $2D$ convolutions with dropout, followed by batch normalization and ReLU activation. The encoder consists of three blocks interspersed with 2×2 max-pooling. The decoder also contains three blocks, but the max-pooling is replaced by 2×2 upconvolution to increase the feature size. Finally, the output of the network is a simple 2×2 convolution with sigmoid activation. This architecture allows the neural network

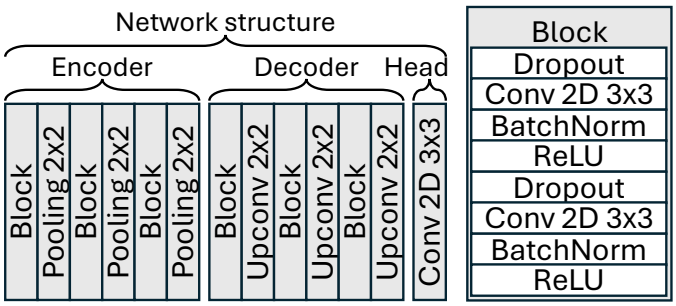

Fig. 3. The convolutional neural network structure

to accept images of any size as input, adapting to many types of broadcasting contents. This architecture is deliberately very simple, since our focus is on measuring the efficiency of the hybrid concept, rather than carefully tuning a network and its hyperparameters for a very specific task.

Our models are trained on the dataset for 20 epochs, using dice loss, and the Adam optimizer. Since we chose to make the output model size adaptive, a batch size of 1 is used.

IV. Experiments and Results

As described in Section III, we combined selected features and tested multiple models for segmentation. We compared them with a 3D encoder-decoder (using 3D convolutions to take the sequence of n frames directly as input), a 2D encoder-decoder with frame-wise processing (no temporal information), and TMANet [34], adapted to our dataset in CamVid format. Since TMANet requires fixed-size inputs (640x640), we created two alternative datasets (by resizing or cropping) and trained/tested all models under the same conditions (NVIDIA A100 GPU) for fair comparison. Performance was measured using mean intersection over union (mIoU). Test sequences contained unseen overlay elements to ensure unbiased evaluation. We also analyzed training/inference times (s) and network sizes, summarizing results averaged over multiple runs in Table I.

Results (also depicted in Figure 4) show that in both datasets, the frame-wise model performs poorly: relying only on shape and color leads to false detections in untextured areas of the frames. The 3D model effectively leverages spatio-temporal features, predicting logos in stable regions. TMANet results are correct for the cropped dataset; however, when applied to the resized dataset that has a higher data imbalance, it underperforms, likely because its temporal memory attention is suited for complex scene semantics rather than fine-detail segmentation as required by our task.

We retained only the best-performing feature combination: the temporal variation map (TV) with spatiotemporal edges (STE), which outperformed both the 3D model and TMANet in both datasets, validating our hybrid approach. Interestingly, although the temporal variation map alone performed the worst, its inclusion still had a positive impact on the best combined model. In contrast, the mean frame (MF) feature was effective individually but did not enhance performance when combined with other features, suggesting that some features

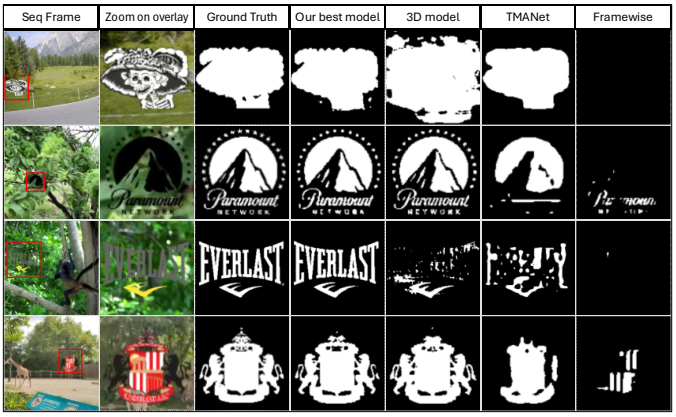

Fig. 4. Some representative results from the cropped test set. Because the display is cropped, some false positives are not visible in this figure

are complementary while others are redundant. Notably, the best model does not rely on color, challenging our initial hypothesis, and prior work focused on this aspect [33]. Further analysis of segmentation errors could clarify each feature's role.

Since frame-wise models (2D and TMANet) produce per-frame segmentations while others do so every 20 frames, we use FPS to compare inference times. Our best model not only outperforms classical methods but also has low inference time. The bottleneck in our approach is data loading, further demonstrating its efficiency.

TABLE I
Models comparison. Each dataset has two mIoU mesures: the overlay class (left) and the mean of the two classes (right)

Models	Params↓	Train time↓	Infer FPS↑	mIoU↑			
				Resized		Cropped	
Our best model	118 K	9 038	60	**0.736**	**0.865**	**0.774**	**0.880**
Spatiotemporal	118 K	8 809	**62**	0.682	0.840	0.706	0.849
3D model	341 K	10 866	50	0.670	0.832	0.659	0.821
Spatial edges	**117** K	9 058	**62**	0.656	0.826	0.715	0.848
TMANet [34]	31 000 K	14 140	20	0.594	0.796	0.729	0.863
Mean frames	118 K	8 911	**62**	0.520	0.758	0.628	0.810
Framewise	118 K	14 813	42	0.421	0.709	0.379	0.684
Temporal var	**117** K	**8 400**	60	0.280	0.637	0.425	0.668

Conclusion

In this study, we have shown that the semantic segmentation of fixed logos in videos is a particular problem due to the inherent characteristics of this type of content. On the one hand, fully neural architectures that use 3D convolution or attention mechanisms to learn relevant spatio-temporal features are computationally intensive. On the other hand, rule-based methods are less flexible, especially for combining different types of cues. Our hybrid approach combines the best of both worlds: the specificity of rule-based features with the flexibility and pooling power of neural networks. The proposed model outperforms its alternatives: It has flexible input size, fewer neurons, and shorter inference time. Future work may extend this concept to other specific segmentation problems. We may also explore the effect of different time windows or

sampling rates, try to quantize and distill our neural network to make it more compact, or change its architecture to better handle multimodal features [41], [42]. Finally, it might be interesting to investigate splitting the feature preprocessing and network inference tasks into different hardware components (e.g., CPU and NPU) to assess the gain in flexibility.

REFERENCES

[1] Q. Monnier, T. Pouli, and K. Kpalma, "Survey on fast dense video segmentation techniques," *Computer Vision and Image Understanding*, vol. 241, p. 103959, 2024.

[2] M. E. Abdulmunim and H. K. Abass, "Logo matching in arabic documents using region based features and surf descriptor," in *Annual Conference on New Trends in Information & Communications Technology Applications*, 2017, pp. 75–79.

[3] S. Hassanzadeh and H. Pourghassem, "A fast logo recognition algorithm in noisy document images," in *International Conference on Intelligent Computation and Bio-Medical Instrumentation*, 2011, pp. 64–67.

[4] H. Pourghassem, "A hierarchical logo detection and recognition algorithm using two-stage segmentation and multiple classifiers," in *International Conference on Computational Intelligence and Communication Networks*, 2012, pp. 227–231.

[5] T. A. Pham, M. Delalandre, and S. Barrat, "A contour-based method for logo detection," in *International Conference on Document Analysis and Recognition*, 2011, pp. 718–722.

[6] N. Sharma, R. Mandal, R. Sharma, U. Pal, and M. Blumenstein, "Signature and logo detection using deep cnn for document image retrieval," in *International Conference on Frontiers in Handwriting Recognition*, 2018, pp. 416–422.

[7] A. V. Nandedkar, J. Mukherjee, and S. Sural, "A spectral filtering based deep learning for detection of logo and stamp," in *National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics*, 2015, pp. 1–4.

[8] Z. Li, M. Schulte-Austum, and M. Neschen, "Fast logo detection and recognition in document images," in *International Conference on Pattern Recognition*, 2010, pp. 2716–2719.

[9] Y. Zhang, S. Zhang, W. Liang, and H. Wang, "Spatial connected component pre-locating algorithm for rapid logo detection," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2012, pp. 1297–1300.

[10] H. Wang and Y. Chen, "Logo detection in document images based on boundary extension of feature rectangles," in *International Conference on Document Analysis and Recognition*, 2009, pp. 1335–1339.

[11] D. L. C. Lu and D. Zeng, "Logo detection based on convolutional neural networks," in *IEEE International Conference on Signal, Information and Data Processing*, 2019, pp. 1–6.

[12] V. P. Le, M. Visani, C. De Tran, and J.-M. Ogier, "Improving logo spotting and matching for document categorization by a post-filter based on homography," in *International Conference on Document Analysis and Recognition*, 2013, pp. 270–274.

[13] S. Hassanzadeh and H. Pourghassem, "Fast logo detection based on morphological features in document images," in *IEEE International Colloquium on Signal Processing and its Applications*, 2011, pp. 283–286.

[14] A. Alaei and M. Delalandre, "A complete logo detection/recognition system for document images," in *IAPR International Workshop on Document Analysis Systems*, 2014, pp. 324–328.

[15] M. S. Shirdhonkar and M. Kokare, "Automatic logo detection in document images," in *IEEE International Conference on Computational Intelligence and Computing Research*, 2010, pp. 1–3.

[16] Z. Ahmed and H. Fella, "Logos extraction on picture documents using shape and color density," in *IEEE International Symposium on Industrial Electronics*, 2008, pp. 2492–2496.

[17] K. Paleček and J. Chaloupka, "Logo detection and identification in system for audio-visual broadcast transcription," in *International Conference on Telecommunications and Signal Processing*, 2021, pp. 357–360.

[18] S. Duffner and C. Garcia, "A neural scheme for robust detection of transparent logos in tv programs," in *Artificial Neural Networks*, 2006, pp. 14–23.

[19] G. Xiao, Y. Dong, Z. Liu, and H. Wang, "Supervised tv logo detection based on svms," in *IEEE InternationalConference on Network Infrastructure and Digital Content*, 2010, pp. 174–178.

[20] B. Guan, H. Ye, H. Liu, and W. A. Sethares, "Video logo retrieval based on local features," in *EEE International Conference on Image Processing*, 2020, pp. 1396–1400.

[21] F. Meng, H. Li, G. Liu, and K. N. Ngan, "From logo to object segmentation," *IEEE Transactions on Multimedia*, vol. 15, no. 8, pp. 2186–2197, 2013.

[22] D. Pan, P. Shi, Z. Qiu, Y. Sha, X. Zhongdi, and J. Zhoushao, "Tv logo classification based on convolutional neural network," in *IEEE International Conference on Information and Automation*, 2016, pp. 1793–1796.

[23] D. Ku, J. Cheng, and G. Gao, "Translucent-static tv logo recognition by susan corner extracting and matching," *International Conference on Innovative Computing Technology*, pp. 44–48, 2013.

[24] S. Y. Arafat, S. A. Husain, I. A. Niaz, and M. Saleem, "Logo detection and recognition in video stream," in *International Conference on Digital Information Management*, 2010, pp. 163–168.

[25] C. Zhao, J. Wang, C. Xie, and H. Lu, "A coarse-to-fine logo recognition method in video streams," in *IEEE International Conference on Multimedia and Expo Workshops*, 2014, pp. 1–6.

[26] J. Wang, Q. Liu, L. Duan, H. Lu, and C. Xu, "Automatic tv logo detection, tracking and removal in broadcast video," in *Advances in Multimedia Modeling*, 2006, pp. 63–72.

[27] W. Yan, J. Wang, and M. Kankanhalli, "Automatic video logo detection and removal," *Multimedia Syst.*, vol. 10, pp. 379–391, 2005.

[28] K. Meisinger, T. Troeger, M. Zeller, and A. Kaup, "Automatic tv logo removal using statistical based logo detection and frequency selective inpainting," in *European Signal Processing Conference*, 2005, pp. 1–4.

[29] J. Cózar, N. Guil, J. González-Linares, and E. Zapata, "Video cataloging based on robust logotype detection," in *International Conference on Image Processing*, 2006, pp. 3217–3220.

[30] J. Wang, L. Duan, Z. Li, J. Liu, and H. Lu, "A robust method for tv logo tracking in video streams," in *IEEE International Conference on Multimedia and Expo*, 2006, pp. 1041–1044.

[31] A. Albiol, M. Ch, F. Albiol, and L. Torres, "Detection of tv commercials," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, 2004, pp. iii–541.

[32] N. Ozay and B. Sankur, "Automatic tv logo detection and classification in broadcast videos," in *European Signal Processing Conference*, 2009.

[33] A. Ekin and R. Braspenning, "Spatial detection of tv channel logos as outliers from the content," in *Visual Communications and Image Processing*, vol. 6077, 2006, p. 60770X.

[34] H. Wang, W. Wang, and J. Liu, "Temporal memory attention for video semantic segmentation," in *IEEE International Conference on Image Processing*, 2021, pp. 2254–2258.

[35] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, " The Cityscapes Dataset for Semantic Urban Scene Understanding ," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3213–3223.

[36] G. J. Brostow, J. Fauqueur, and R. Cipolla, "Semantic object classes in video: A high-definition ground truth database," *Pattern Recognition Letters*, vol. 30, no. 2, pp. 88–97, 2009, video-based Object and Event Analysis.

[37] S. Caelles, J. Pont-Tuset, F. Perazzi, A. Montes, K.-K. Maninis, and L. Van Gool, "The 2019 davis challenge on vos: Unsupervised multi-object segmentation," *arXiv:1905.00737*, 2019.

[38] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: a large video database for human motion recognition," in *Proceedings of the International Conference on Computer Vision*, 2011.

[39] A. Stergiou and R. Poppe, "Adapool: Exponential adaptive pooling for information-retaining downsampling," *IEEE Transactions on Image Processing*, vol. 32, pp. 251–266, 2021.

[40] J. Qi, Y. Gao, Y. Hu, X. Wang, X. Liu, X. Bai, S. Belongie, A. Yuille, P. Torr, and S. Bai, "Occluded video instance segmentation: A benchmark," *International Journal of Computer Vision*, 2022.

[41] Y. Sun, Z. Fu, C. Sun, Y. Hu, and S. Zhang, "Deep multimodal fusion network for semantic segmentation using remote sensing image and lidar data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–18, 2022.

[42] Z. Guo, X. Li, H. Huang, N. Guo, and Q. Li, "Deep learning-based image segmentation on multimodal medical imaging," *IEEE Transactions on Radiation and Plasma Medical Sciences*, vol. 3, no. 2, pp. 162–169, 2019.