Jestr

Journal of Engineering Science and Technology Review 17 (5) (2024) 55-61

Research Article

JOURNAL OF
Engineering Science and Technology Review

www.jestr.org

Adaptive Video Text Tracking Based on Pixel-level Feature Extraction

Bangguo Zhang[1,*] and Fenggang Liu[2]

[1]*School of Computer Science, Hubei University of Technology, Wuhan 430000, Hubei, China*
[2]*School of Artificial Intelligence, Wuchang University of Technology, Wuhan 430223, Hubei, China*

Received 19 December 2023; Accepted 30 August 2024

Abstract

The increasing utilization of video text has made text detection and tracking in videos an important research direction in the field of computer vision. To address the issue of existing methods being inaccurate at detecting curved text, an adaptive video text tracking (ATDTF) model based on pixel-level feature extraction was proposed. First, pixel-level information was used for text detection, and then multi-directional prediction and connected component analysis were performed to achieve the accurate detection of text in complex scenes. Next, the tracking algorithm was used to associate and track text targets in consecutive frames. Finally, the target motion model or appearance model, in combination with Kalman filtering, was used to associate and predict the trajectory to achieve accurate tracking of the target. Experimental results show that, the ATDTF model improves the F1 value by 0.6% over the current method, the detection speed is at least 21 frames per second faster, and the running speed is increased twofold. The proposed model achieves performance improvement in video text detection and tracking tasks, providing an effective end-to-end solution for video text information processing.

Keywords: Video text, Text detection, Text tracking, Feature extraction, End-to-end

1. Introduction

Video text refers to text embedded in video frames, usually containing rich semantic information such as logos, announcements, advertisements, and subtitles. Video text is widely used in fields such as autonomous driving and intelligent monitoring. It helps users better understand video content by providing contextual information or additional explanations, making it valuable for certain applications. However, due to the diverse and dynamic nature of video text, traditional text detection and tracking methods often struggle to achieve ideal results.

Video text detection and tracking tasks mainly include detection and tracking. The goal of detection is to accurately identify the text area in the video frame, while that of tracking is to associate the above text areas across frames to achieve continuous tracking of video text. Existing methods usually adopt complex processes, but they fail to fully capture the semantic connections between consecutive video frames and ignore the real-time requirements of video text tracking [1]. In detection, methods based on deep learning have made significant progress in recent years. For example, the pixel-based scene text detector (PSENet) [2] is an advanced text detection algorithm that detects pixel-level information and can handle complex scenes with high accuracy. In tracking, the multiple objects tracking re-identification (MOTR) algorithm achieves accurate tracking of multiple targets by leveraging target re-identification technology.

Although the above methods perform well in their respective fields, in video text detection and tracking tasks, using detection or tracking methods alone often does not achieve the best results. To solve this problem, this study proposes an adaptive video text tracking (ATDTF) model based on pixel-level feature extraction. This model uses the PSENet algorithm to perform efficient text detection and the MOTR algorithm to achieve continuous tracking of detected text targets, thereby improving the real-time effectiveness of video text information processing.

2. State of art

Text detection and tracking is an important research direction in the field of computer vision, with applications spanning scene understanding, autonomous driving, augmented reality, intelligent monitoring, and many more. Existing studies mainly focused on text detection and tracking in static image, these involving edge detection, feature extraction, and classification methods [3]. For example, traditional edge detection methods identify text edges by detecting the mutation area of grayscale value in the image, and feature extraction describes the shape and structure of the text area by extracting local features, finally classifying the text area using classifiers such as support vector machine [4]. However, static image-based methods have many limitations in dealing with the continuity and dynamic changes of text in videos.

Video text usually moves, deforms, or becomes partially occluded as video frames change, posing challenges for traditional static image processing methods [5]. To overcome these issues, researchers have begun to explore video text detection and tracking methods based on deep learning in recent years. Text detection methods based on deep learning design deep convolutional networks to learn more robust feature representations from large volumes of annotated data, enabling them to cope with text detection tasks in complex scenes [6]. Among them, the efficient and

*E-mail address: BigEar111@163.com
ISSN: 1791-2377
doi:10.25103/jestr.175.07

accurate scene text (EAST) detector is a text detection method based on deep learning. It directly predicts the bounding box and score of the text area through a fully convolutional network, thereby achieving efficient text detection [7]. However, it still has certain limitations when dealing with dynamic text detection tasks in videos.

In response to the challenges of video text detection, researchers have proposed methods that combine temporal information. Recurrent neural networks (RNNs) and long short-term memory (LSTM) networks are applied to video text detection tasks owing to their superior performance in processing time series data. Eshwarappa et al. proposed a text detection method based on LSTM, which can better capture text changes between video frames by combining the temporal information of consecutive frames [8]. However, these methods still face certain challenges when dealing with significant changes between video frames, such as fast motion, severe deformation, and so forth. To further improve the performance of video text detection and tracking, recent studies have begun to explore multi-task learning and end-to-end training methods [9]. Although these methods have performed well in experiments, they still need further optimization in practical applications due to the complexity and diversity of video scenes.

Target association and trajectory prediction are key technologies to solve the problem of continuous video text tracking [10]. Target association identifies different video frames and associates the same text target to achieve continuous tracking of the target. Trajectory prediction predicts the target's position in future frames by analyzing its motion trajectory in past frames [11]. The combination of these two technologies can significantly improve the robustness and accuracy of video text tracking.

In the context of target association, traditional methods primarily rely on motion models and appearance models. The motion model assumes that the target's movement between adjacent frames is continuous and predictable, with common techniques including Kalman filtering and particle filtering [12]. These methods construct a motion model for the target, predict its position in the subsequent frame, and achieve target association by minimizing the error between the predicted position and the actual detected position. However, these approaches may encounter significant prediction errors when processing text targets that exhibit rapid motion or complex motion patterns.

The appearance model, in contrast, facilitates the matching and association of different frames by extracting the appearance features of the target, such as color, texture, and shape. Commonly utilized appearance features include histogram features, gradient features, and depth features. In recent years, advancements in deep learning have led to significant improvements in appearance feature extraction methods based on convolutional neural networks (CNNs), enhancing the efficacy of target association tasks [13]. For example, DeepSORT (Simple Online and Real-time Tracking with a Deep Association Metric) achieves real-time, high-precision target association by integrating a deep learning-based appearance feature extractor with Kalman filtering [14].

In the realm of trajectory prediction, traditional methods primarily rely on linear motion models, such as the commonly used uniform linear motion model [15]. However, in practical applications, target movement is often nonlinear and complex, making it challenging for a simple linear model to accurately forecast the target's future position. To address this issue, researchers have begun to investigate trajectory prediction methods based on deep learning. RNNs and LSTM are widely employed in trajectory prediction tasks due to their exceptional performance in processing time series data.

Although target association and trajectory prediction have demonstrated significant potential in video text tracking tasks, these methods still encounter challenges in complex scenes, such as text occlusion and variations in illumination. To enhance the robustness of target association and trajectory prediction, several studies have proposed methods for fusing multi-modal information. For instance, Kong et al. introduced a multi-modal fusion target association method that successfully achieved robust target association in complex environments by integrating visual and motion features [16]. Furthermore, research utilizing the multi-object tracking (MOT) framework has also made notable advancements in target association and trajectory prediction. The MOTR (Multiple Object Tracking Re-Identification) framework enables efficient and accurate tracking of multiple targets by combining target detection and re-identification technologies [17].

In summary, although video text detection and tracking have made significant progress in previous research, they still encounter numerous challenges when addressing text targets in complex video scenes. By integrating detection and tracking methods grounded in deep learning and employing target association and trajectory prediction technologies, the robustness and accuracy of video text tracking can be further enhanced. Consequently, this study proposes an adaptive video text tracking model based on pixel-level feature extraction, aimed at addressing these challenges. The model has been experimentally validated on multiple public datasets, demonstrating substantial performance improvements.

3. Methodology

This study proposes an ATDTF model based on pixel-level feature extraction, which combines PSENet and MOTR algorithms to achieve the efficient detection and continuous tracking of text objects in videos. The ATDTF model mainly includes several modules, including text detection, video text tracking, continuous multi-frame processing, and loss function. The structure and function of each module are introduced in detail below.

3.1 Text Detection

The text detection module is an important part of the ATDTF framework and is responsible for locating and identifying text objects in video frames. This module adopts the PSENet text detection algorithm based on deep learning, extracts feature from video frames through convolutional neural networks (CNNs), and uses multi-directional prediction and connected component analysis (CCA) to locate and identify text regions in images.

The text detection module uses a pre-trained deep CNN (e.g., ResNet) to extract features from input video frames. Multi-level convolution operations [18] are conducted to extract multi-scale image features that can capture information such as edges, textures, and shapes of text regions. PSENet uses a multi-directional head network to predict the text boundary of each pixel. Head networks in different directions can accurately locate text areas of complex shapes by integrating the prediction results of multiple directions. Based on the multi-directional prediction,

PSENet uses the CCA algorithm to merge and classify the detected text areas. This algorithm connects discrete text pixels into complete text instances by analyzing the connection relationship at the pixel level, thereby accurately separating closely adjacent text instances.

The advantage of PSENet lies in its pixel-level detection capability and progressive scale expansion algorithm. Through the efficient feature extraction and precise detection capabilities of PSENet, the text detection module can accurately locate text targets in complex video scenes, providing high-quality detection results for subsequent video text tracking.

3.2 Video Text Tracking

The video text tracking module aims to achieve continuous tracking of text in videos. This module continuously tracks text targets by combining target detection, feature extraction, and tracking algorithms.

The video text tracking module begins by using the PSENet algorithm to detect text regions in each frame. It then extracts image features through CNNs and uses multi-directional head prediction and CCA to locate text regions [19]. Next, feature vectors capturing the appearance information of the text are extracted from the detected regions using CNNs. Finally, the MOTR algorithm is used for target association and tracking. It combines target detection and target re-identification techniques and associates and matches the detected text targets through Kalman filtering and the Hungarian algorithm, thereby achieving continuous tracking of the target.

Through the above steps, the video text tracking module can achieve efficient detection and continuous tracking of text targets in videos. This module combines the high-precision text detection capability of PSENet and the robust target-tracking capability of MOTR, enabling it to accurately locate and continuously track text targets in complex video scenes.

3.3 Continuous Multi-frame Processing

The continuous multi-frame processing module aims to solve the problem of continuous tracking of text targets in videos. This module realizes continuous tracking of text targets by analyzing the motion and deformation of text targets between consecutive frames.

The continuous multi-frame processing module uses Kalman filtering to predict the motion trajectory of text targets and predicts the position of the target in the next frame by modeling the motion state of the target in the previous frames. The appearance features of the text targets are extracted using a deep CNN, and the targets are matched and associated based on these features. The Hungarian algorithm is used for target association after combining the motion prediction of the Kalman filter and the feature matching of the appearance model. This algorithm achieves the optimal matching of targets by minimizing the matching cost between the detected and tracked targets. By comprehensively considering the motion state and appearance features of the target, the Hungarian algorithm can accurately associate the same text target between consecutive frames.

The continuous multi-frame processing module realizes continuous tracking of text targets by combining the motion prediction and appearance models, ensuring the association and continuity of text targets between video frames.

3.4 Loss Function

The loss function module plays an important role in the ATDTF model. It is used to measure the difference between the model's prediction results and the true labels and serves as an optimization target to guide model training. We design a multi-task learning loss function that comprehensively considers the losses of the text detection and tracking tasks.

The loss function module uses the cross-entropy loss function to measure the difference between the text detection results and the true labels. A loss function based on the target motion trajectory is designed to guide the model in learning accurate text target tracking by comparing the difference between the predicted and true trajectories [20]. Text detection loss and text tracking loss are weighted and fused to construct a comprehensive loss function for multi-task learning. Through multi-task loss fusion, the model can simultaneously optimize the performance of the two detection and tracking tasks during training.

The loss function module is of great significance in the ATDTF model. It not only measures the prediction performance of the model but also guides its optimization and learning. By designing a reasonable multi-task learning loss function, the ATDTF model can achieve end-to-end optimization training of text targets and improve the overall performance of text detection and tracking.

In summary, the ATDTF model combines the efficient text detection capability of PSENet and the robust target tracking capability of MOTR to build a high-performance end-to-end video text detection and tracking system. The various modules work closely together to achieve efficient detection and continuous tracking of text targets in complex video scenes through feature extraction, target association, motion prediction, and multi-task loss optimization. Experimental results show that the ATDTF model achieves significant performance improvements on multiple public datasets, verifying its effectiveness and robustness in practical applications.

4. Results Analysis

4.1 Datasets and Experimental Details

In this study, we selected several public datasets for experimental evaluation to fully verify the video text detection and tracking performance of the ATDTF model in different scenarios. The datasets used include ICDAR 2015 Video, ICDAR 2013 Video, and YouTube-VideoText (YVT), which cover various complex scenarios such as different lighting conditions, text sizes, font styles, motion blur, and occlusion.

The ICDAR 2015 Video dataset is a standard text detection and recognition dataset that contains numerous natural scene video clips comprising text information in various complex backgrounds. The ICDAR 2013 Video dataset is similar, containing a rich variety of scenes required for video text detection tasks. These video datasets provide real-world text instances that help evaluate and verify the performance of the model in practical applications. The YVT dataset contains text clips extracted from real videos, showing the dynamic changes and complex motion patterns of text in videos.

Experimental Details: To ensure the diversity and representativeness of the data, we performed standard preprocessing steps on each dataset. Preprocessing includes extracting frames from videos, adjusting image size, grayscale processing, and applying data enhancement

techniques (e.g., random cropping, rotation, and scaling) to improve the robustness and generalization ability of the model. The specific preprocessing process is as follows:

(1) Video frame extraction: Extract frames from videos to generate image sequences for training and testing.

(2) Image resizing: All images are uniformly adjusted to a fixed size to ensure the consistency of input data.

(3) Grayscale processing: Convert color images to grayscale images to reduce computational complexity.

(4) Data enhancement: Apply a variety of data enhancement techniques (e.g., random cropping, rotation, scaling) to increase the diversity of the dataset and improve the generalization ability of the model.

The training and testing of the model were performed on a high-performance computer equipped with an NVIDIA Tesla V100 GPU, and the deep learning framework used was PyTorch. We used the Adam optimizer for training, set the initial learning rate to 0.001, and decayed the learning rate during training (every 10 epochs). Each model was iterated multiple times during training to ensure the convergence and performance optimization of the model. After training, we evaluated the model on the test set and calculated the accuracy, recall, F1 value, and other indicators for text detection and tracking.

In terms of experimental details, we performed standard preprocessing steps on each dataset, including video frame extraction, image resizing, grayscale processing and data augmentation techniques (e.g., random cropping, rotation, scaling) to improve the robustness and generalization ability of the model. We used the OpenCV library for video frame processing and adopted common data augmentation techniques to increase the diversity of training data. These steps helped improve the training effect of the model and effectively simulated different practical application scenarios. Each model was trained for multiple iterations to ensure the convergence and performance optimization of the model. After training, we evaluated the model on the test set and calculated the accuracy, recall, F1 value, and other indicators for text detection and tracking.

To evaluate the performance of the model, we divided the dataset into a training set and a test set. We adopted multiple training validations in the ICDAR 2015 Video, ICDAR 2013 Video, and YVT datasets. This ensured the diversity and representativeness of data during training and testing, thereby improving the reliability of evaluation results.

Through the above experimental settings and detailed preprocessing steps, we could comprehensively evaluate the performance of the ATDTF model in different datasets and scenarios. The experimental results show that the ATDTF model can achieve excellent performance when dealing with complex video text detection and tracking tasks, demonstrating its potential and value in practical applications.

4.2 Experimental Environment

This experiment was conducted on a high-performance computer equipped with an NVIDIA Tesla V100 GPU. This GPU provides powerful computing power and large-scale parallel processing capabilities, enabling us to efficiently train deep learning models. The deep learning framework uses PyTorch, and model training and testing are completed in this environment. The experimental environment is shown in Table 1.

Table 1. Hardware and Software Configuration

Configuration Type	Components	More Information
Hardware Configuration	GPU	NVIDIA Tesla V100
	CPU	Intel Xeon E5-2698 v4
	Memory	256GB DDR4 RAM
	Storage	2TB SSD
Software Configuration	Operating System	Ubuntu 18.04 LTS
	Deep Learning Frameworks	PyTorch 1.7.0
	CUDA Version	10.2
	cuDNN Version	7.6

In such a hardware and software environment, we can efficiently train and test large-scale datasets to ensure the high performance and stability of the model. During the experiment, we also used a variety of tools and libraries to assist in data processing and result analysis, visualization, and so on.

4.3 Evaluation Indicators

In the experiment, various evaluation indicators were used to evaluate the performance of the model, including accuracy, recall, and F1 value. These indicators can comprehensively measure the performance of the model in text detection and tracking tasks and are specifically defined below.

Text detection indicators:

Precision: Precision is used to measure how much of the text area detected by the model is real text. The calculation formula is

$$Accuracy = \frac{TP}{TP + FP} \tag{1}$$

where *TP* (True Positive) indicates the number of correctly detected text instances, and *FP* (False Positive) indicates the number of incorrectly detected text instances.

Recall: Recall is used to measure how many of all the real text regions that the model finds are detected. The calculation formula is as follows:

$$Recall = \frac{TP}{FP + FN} \tag{2}$$

where *FN* (False Negative) represents the number of real text instances that failed to be detected.

F1-Score: *F1*-Score is a comprehensive measure of accuracy and recall that can provide a comprehensive evaluation of the overall performance of the model. The calculation formula is

$$F1 = 2 \times \frac{Accuracy}{Recall} \tag{3}$$

Researchers have developed a variety of indicators to comprehensively evaluate the performance of multi-target video tracking algorithms. Among them, multi-target tracking accuracy (*MOTA*) and multi-target tracking precision (*MOTP*) are the two most widely used evaluation indicators. They not only consider the tracking accuracy but also include an evaluation of the algorithm's performance in dealing with occlusion, false detection, and target loss, as well as the *F1* score (*IDF1*) value based on target identity.

MOTA comprehensively considers the effects of false detection, omission, and *ID* switching, thus providing an indicator to quantify the overall performance of multi-target tracking. The calculation formula of MOTA is

$$MOTA = 1 - \frac{\sum_t (FN_t + FP_t + IDSW_t)}{\sum_t GT_t} \tag{4}$$

where FN_t represents the number of missed targets at time, FP_t represents the number of falsely detected targets, $IDSW_t$ is the number of ID switches, and GT_t is the total number of true targets. The value of MOTA ranges from -∞ to 1, with higher values indicating better tracking performance.

MOTP measures the average error between the predicted position of a correctly tracked target and its true position, thus reflecting the accuracy of the tracker in locating the target. The calculation formula of MOTP is as follows:

$$MOTP = \frac{\sum_{i,t} d_{i,t}}{\sum_t M_t} \tag{5}$$

where $d_{i,t}$ is the distance between the predicted position and the true position of the target tracked at time, and M_t is the number of correctly matched targets at time. MOTP is measured in distance units (e.g., pixels), with lower values indicating higher position accuracy.

The IDF1 metric was proposed by Ristani et al. in 2016 to evaluate the effectiveness of tracking algorithms in maintaining the consistency of target identities. It is calculated by comparing the consistency between the predicted identity and the true identity. Specifically, IDF1 is the F1 score based on the target identity, combining the harmonic mean of precision and recall. The calculation formula is shown in Formula 6:

$$IDF1 = \frac{2 \times IDTP}{2 \times IDFP + IDFN \times IDFN} \tag{6}$$

where *IDTP* (True Positive) represents the number of correctly matched identities, *IDFP* (False Positive) represents the number of incorrectly matched identities, and IDFN (False Negative) represents the number of unmatched true identities.

During the experiment, we calculated these indicators on the ICDAR 2015 Video, ICDAR 2013 Video, and YVT datasets to comprehensively evaluate the performance of the ATDTF model. Through these evaluation indicators, we could quantify the performance of the model in different scenarios and conditions, thereby verifying its effectiveness and robustness in practical applications. The experimental results show that the ATDTF model performs well on all test datasets, achieving high precision, recall, and F1 values, fully demonstrating its superior performance in video text detection and tracking tasks.

4.4 Experimental analysis of video text detection

Video text detection is one of the key modules in the ATDTF model, and its main task is to accurately detect text regions from video frames. Multiple datasets are used to evaluate the text detection performance of the ATDTF model. To improve the accuracy and robustness of detection, we designed a multi-scale detection algorithm based on deep learning. The text detection module uses a pre-trained deep CNN (e.g., ResNet) to extract features from the input video frames. Through multi-level convolution operations, multi-scale image features are extracted that can capture information such as the edge, texture, and shape of the text region. The multi-scale feature pyramid network (FPN) is combined with the multi-directional prediction and CCA of PSENet to capture text information at different scales. FPN can effectively detect text regions of various sizes and shapes by fusing features at different scales.

Fig.1. Results of tracking visualization

Table 2. Video tracking accuracy

Dataset	Method	Video text tracking/%					FPS
		IDF1	MOTA	MOTP	M-Matched	M-Lost	
ICD15 video	USTB TexVideo	25.9	7.4	70.8	7.4	66.1	
	StradVision-1	25.9	7.9	70.2	6.5	70.8	
	USTB_TexVideo(II-2)	21.9	12.3	71.8	4.8	72.3	
	AJOU	36.1	16.4	72.7	14.1	62.0	8.8
	Free	57.9	43.2	76.7	36.6	44.4	8.8
	TransVTSpotter	57.3	44.1	75.8	34.3	33.7	9.0
	our ATDTF	58.5	46.2	74.6	33.7	32.8	30.1
ICD13 video	our ATDTF	49.5	42.3	71.6	34.1	38.2	33.1
YVT	our ATDTF	61.5	49.3	76.4	38.1	39.3	34.4

To further improve the detection accuracy, we added data augmentation techniques during the training process. These techniques include random cropping, rotation, flipping, and color perturbation, all of which help to enhance the generalization and robustness of the model. Through these methods, the model can still maintain high detection accuracy in the face of various complex scenes.

During the training process, the model is optimized by the cross-entropy loss function and the Intersection over Union (IoU) loss function using the annotated text region data. The cross-entropy loss is used to measure the classification error, while the IoU loss is used to measure the overlap between the detection box and the true box, thereby improving the detection accuracy. Specifically, the cross-entropy loss is used to calculate the classification error between the predicted text region and the actual label, and the IoU loss is used to evaluate the overlap between the predicted bounding box and the actual bounding box. By combining these two loss functions, the model can continuously adjust its parameters during the training process to improve the accuracy and robustness of text detection.

Comparative experiments on video text detection tasks on multiple public datasets demonstrate the performance advantages of the ATDTF model. The image pixels are fixed to 736 to show the trade-off between accuracy and speed. In these experiments, the ATDTF model improves the F value by 0.6% over the current method Free, the detection speed is at least 21 frames per second faster, and the running speed is increased twofold, as shown in Table 2. In addition, the visualization results are shown in Fig 1, which demonstrates the effectiveness of the model in the text detection task.

Through the above experimental analysis and results, the advantages of the ATDTF model in the video text detection task are verified, and its significant improvement in accuracy and robustness is demonstrated.

4.5 Comparative Experiment

Statistics of text size indicators: The text size often affects the accuracy of video text recognition. To better understand this impact, the text was classified and counted according to the area size of the text.

The classification of area sizes is as follows: small size, the text area does not exceed 1500 pixels; medium size, the text area is between 1500 and 2500 pixels; and large size, text area exceeds 2500 pixels. The model visualizes the area size classification (Table 3 and Fig 2). Accordingly, it can be judged that the medium-sized text has the best tracking effect.

Table 3. Area size classification

Dataset	Video text tracking/%				
ICD13Video	ID_{F1}	MOTA	MOTP	M-Matched	M-Lost
Small	44.2	20.5	71.1	28.6	53.3
Medium	43.5	5.1	77.7	17.7	49.6
Big	74.1	58.1	80.2	55.7	18.8

For example, the ATDTF model shows higher robustness and stability in dealing with motion blur, illumination changes, and partial occlusion. This is mainly due to the multi-scale feature extraction and multi-task learning strategy we introduced in the model design. Through multi-scale feature extraction, the ATDTF model can better capture text information of different scales, while the multi-task learning strategy improves the overall performance of the model by optimizing text detection and tracking tasks simultaneously.

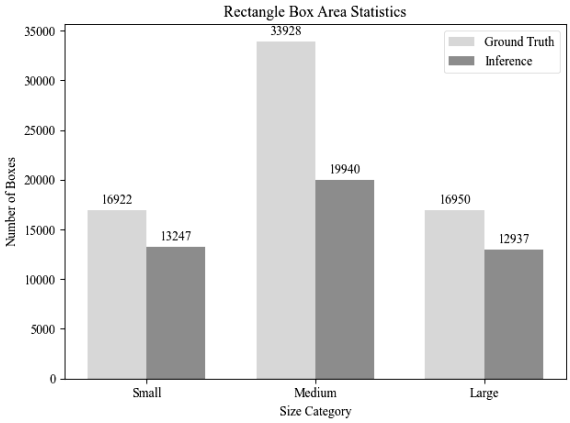

Fig. 2. Visualization results of area size classification

In this study, an ATDTF model based on pixel-level feature extraction is proposed for the first time. ATDTF solves two tasks, text detection and tracking, in one model and adopts the semantic information between learning contexts of consecutive frames. With the help of lightweight architecture, such as backbone network, effective detection head, and tracking head, ATDTF completes video text tracking with an IDF1 of 61.5% at 34.4 FPS on YVT (Table 2). This truly end-to-end and high inference speed method will be applied to more video and language tasks in the future.

The comparative experimental results fully demonstrate the superior performance of the ATDTF model in video text detection and tracking tasks and demonstrate its potential and value in practical applications. These results verify the effectiveness of the ATDTF model and provide important references and lessons for future research. Through further optimization and improvement, the ATDTF model is expected to play an important role in more practical applications and contribute to the development of the field of video text information processing.

5. Conclusion

This study proposes an ATDTF model based on pixel-level feature extraction, which comprehensively utilizes text detection and multi-target tracking algorithms to achieve accurate detection and continuous tracking of text in videos. The conclusions are as follows:

(1) The performance of the ATDTF model in text detection and tracking is significantly improved, and it can effectively handle a variety of video scenarios, which is beneficial to improving the accuracy of analysis and understanding of video content.

(2) Experimental results show that the ATDTF model exhibits high accuracy and stability in most test scenarios, verifying the practical application value of this method.

(3) The detection and tracking errors of the ATDTF model are small, indicating that the model can still maintain high recognition accuracy under various complex backgrounds and has strong robustness.

(4) The model proposed in this article reaches a high level in both recall and accuracy, further proving its practicality and effectiveness in video text tracking tasks.

Although the research in this article achieves certain results, it still has shortcomings. In particular, the model's adaptability in extremely complex and changeable scenarios still needs to be improved, and the demand for computing

resources is high. In the future, it is necessary to optimize the model structure, integrate more complex scene data, and improve real-time performance to improve the accuracy and practicality of video text tracking and provide a more reliable tool for in-depth understanding and intelligent analysis of video content.

References

[1] F. Naiemi, V. Ghods, and H. Khalesi, "Scene text detection and recognition: a survey," *Multimed Tools. Appl*., vol. 81, no. 14, pp. 20255-20290, Mar. 2022.

[2] L. Huang, S. Liao, and W. Yang, "Dc-Psenet: a novel scene text detection method integrating double resnet-based and changed channels recursive feature pyramid," *Vis. Comput*., vol. 40, no. 6, pp. 4473-4491, Jul. 2024.

[3] D. Hettiarachchi, Y. Tian, H. Yu, and S. Kamijo, "Text Spotting towards Perceptually Aliased Urban Place Recognition," *Multimodal Technol. Interact*., vol. 6, no.11, Art. no. 102, Nov. 2022.

[4] S. Dhivy, J. Sangeetha, and B. J. S. C. Sudhakar, "Copy-move forgery detection using surf feature extraction and SVM supervised learning technique," *Soft Comput.*, vol. 24, no. 19, pp. 14429-14440, Mar. 2020.

[5] W. Wu, et al., "End-to-end video text spotting with transformer," *Int. J. Comput. Vis*., vol. 132, pp. 1-17, Jul.2024.

[6] R. Bagi, T. Dutta, "Cost-effective and smart text sensing and spotting in blurry scene images using deep networks," *IEEE Sens. J.*, vol.21, no.22, pp.25307-25314, Nov. 2021

[7] T. Khan, R. Sarkar, and A. F. Mollah, "Deep learning approaches to scene text detection: a comprehensive review," *Artif. Intell. Rev.*, vol. 54, pp. 3239-3298, Jan. 2021.

[8] L. Eshwarappa and G. G. Rajput, "Optimal classification model for text detection and recognition in video frames," *Int. J. Image. Graph.*, to be published. Accessed: Aug. 4, 2023. doi: 10.1142/S0219467825500147.

[9] S. K. Pal, A. Pramanik, J. Maiti, and P. Mitra, "Deep learning in multi-object detection and tracking: state of the art," *Appl. Intell.*, vol. 51, pp.6400-6429, Oct. 2021.

[10] R. Bagi, T. Dutta, N. Nigam, D. Verma, and H. P. Gupta, "Met-MLTS: leveraging smartphones for end-to-end spotting of multilingual oriented scene texts and traffic signs in adverse meteorological conditions," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 12801-12810, Oct.2021.

[11] M. Labani, P. Moradi, and M. Jalili, "A multi-objective genetic algorithm for text feature selection using the relative discriminative criterion," *Expert Syst. Appl.*, vol. 149, Art. no. 113276, Jul. 2020.

[12] W. Qi, Z. Liu, Y. Xu, and J. Wang, "Moving object detection and trajectory prediction based on image processing," *J. Comput. Methods Sci. Eng.*, vol. 22, no. 6, pp.2149-2159, May. 2022.

[13] I. Hussain, R. Ahmad, S. Muhammad, K. Ullah, H. Shah, and A. Namoun, "PHTI: Pashto handwritten text imagebase for deep learning applications," *IEEE Access.*, vol. 10, pp. 113149-113157, Oct. 2022.

[14] C. Xiao, et al., "Motiontrack: learning motion predictor for multiple object tracking," *Neural Netw.*, vol. 179, Art. no. 106539, Mar. 2024.

[15] W. Gao, G. Liao, S. Ma, G. Li, Y. Liang, and W. Lin, "Unified information fusion network for multi-modal rgb-d and rgb-t salient object detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 4, pp. 2091-2106, May. 2021

[16] J. Kong, E. Mo, M. Jiang, and T. Liu, "MOTFR: Multiple object tracking based on feature recoding," *IEEE Trans. Circuits Syst. Video Technol.*, vol.32, no. 11, pp. 7746-7757, Nov. 2022.

[17] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, "Fairmot: on the fairness of detection and re-identification in multiple object tracking," *Int. J. Comput. Vis.*, vol. 129, pp.3069-3087, Sep. 2021.

[18] S. Soni, S. S. Chouhan, and S. S. Rathore, "Textconvonet: A convolutional neural network based architecture for text classification," *Appl. Intell.*, vol. 53, no. 11, pp.14249-14268, Jun. 2023.

[19] G. Liao, Z. Zhu, Y. Bai, T. Liu, and Z. Xie, "Psenet-Based Efficient Scene Text Detection," *Eurasip J. Adv. Signal Process.*, vol. 2021, Art. no. 97, Oct. 2021.

[20] G. Ciaparrone, F. L. Sánchez, S. Tabik, L. Troiano, R. Tagliaferri, and F. Herrera, "Deep learning in video multi-object tracking: A survey". *Neurocomputing*, vol. 381, pp. 61-88, Jun. 2020.