Received May 13, 2018, accepted June 18, 2018, date of publication July 2, 2018, date of current version August 7, 2018.
*Digital Object Identifier 10.1109/ACCESS.2018.2851942*

# A Novel Approach for Video Text Detection and Recognition Based on a Corner Response Feature Map and Transferred Deep Convolutional Neural Network

**WEI LU, HONGBO SUN, JINGHUI CHU, XIANGDONG HUANG, (Member, IEEE), AND JIEXIAO YU**
School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China
Corresponding author: Xiangdong Huang (xdhuang@tju.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61501322.

**ABSTRACT** The text presented in videos contains important information for content analysis, indexing, and retrieval of videos. The key technique for extracting this information is to find, verify, and recognize video text in various languages and fonts against complex backgrounds. In this paper, we propose a novel method that combines a corner response feature map and transferred deep convolutional neural networks for detecting and recognizing video text. First, we use a corner response feature map to detect candidate text regions with a high recall. Next, we partition the candidate text regions into candidate text lines by projection analysis using two alternative methods. We then construct classification networks transferred from VGG16, ResNet50, and InceptionV3 to eliminate false positives. Finally, we develop a novel fuzzy c-means clustering-based separation algorithm to obtain a clean text layer from complex backgrounds so that the text is correctly recognized by commercial optical character recognition software. The proposed method is robust and has good performance on video text detection and recognition, which was evaluated on three publicly available test data sets and on the high-resolution test data set we constructed.

**INDEX TERMS** Video text detection and recognition, corner response feature map, transferred convolutional neural network, fuzzy c-means clustering.

## I. INTRODUCTION

With the rapid development of the internet, communication technology, and smart phones, video has become the most popular medium. The number of online videos has increased dramatically because of the convenience of uploading and downloading videos. Accordingly, there is a high demand for efficient indexing, retrieval, and localization of desired content from massive videos. A lot of algorithms have been developed for this purpose [1]–[3]. For video indexing and retrieval, the video text can depict the video more directly and accurately compared with low-level perceptual content (such as edge, shape, and texture) and other semantic content (such as face, vehicle, and human action) [4]. Furthermore, the content analysis of video text can be used to monitor illegal videos. On the other hand, after text areas are localized, neighbor-pixel interpolation algorithms can be used to restore images that are blocked by text [5], [6]. Thus, video text detection and recognition are significant and challenging tasks because of variations in languages, fonts, and complex backgrounds [7].

Generally speaking, video text can be classified into scene text and artificial text [4]. Of these, the latter usually concisely depicts important video content. For instance, captions in news videos usually describe event information, and subtitles in speech videos usually provide core ideas. Thus, in this paper, we focus on the detection and recognition of artificial text in video frames. Video OCR technology [8] generally has similar processing steps, including text detection, localization, extraction, and recognition. The detection step aims to find text regions, the localization step concentrates on the accurate position of text lines. The extraction step focuses on separating clean text from the complex background.

Text recognition generally can be performed successfully with commercial OCR software, and, therefore, it is beyond the scope of this article.

Traditional methods proposed for the first three steps can roughly be classified into two categories. The first category utilizes inter-frame information to detect video text. For example, Yusufu *et al.* [9] proposed using the variation in the number of SURF feature points in the horizontal and vertical directions between the adjacent frames to track video text. Huang [10] proposed a text extraction method based on a Log-Gabor filter, which selects the 1st, 10th, 20th, and 30th frames from 30 consecutive video frames containing the same text characters for filtering, clustering, and synthesizing in order to obtain video text. The second category is based on the connectivity of text, edge, and texture features. Shivakumara *et al.* [11] used a designed edge feature detector to accurately identify the boundary of text lines. Zhao *et al.* [12] detected video text using the Harris corner points under some heuristic rules. Recently, neural networks have been widely used in uncertain continuous function approximation and discriminative features learning. For example, Niu *et al.* [13] approximated the desired virtual stabilizing functions and desired actual control input by applying radial basis function (RBF) neural networks in the controller design procedure. Wang *et al.* [14]–[16] proposed corresponding adaptive neural control approaches for different systems based on the universal approximation property of the RBF neural networks. Wang *et al.* [17] combined a convolutional neural network (CNN) with unsupervised feature learning to detect and recognize video text. Based on a specific CNN, Saidane and Garcia [18] proposed an automatic binarization method for color text regions in videos.

Previous methods have achieved promising performance, but there are still two inevitable problems. First, it is difficult to identify text regions accurately and completely because of various languages, fonts, resolutions, and particularly complex backgrounds. For example, edge-based approaches may produce many false positives when the complex background also has a high density of edges. Second, the heuristic constraints and machine learning methods proposed to eliminate false positives for video text are always optimized for specific situations, which reduces the generalizability of these methods.

In fact, no matter what the language and font the text has, the component characters are always formed by crosses of strokes in limited space. Therefore, many corners exist [12]. CNNs can learn discriminative features for precise classifications directly from a large amount of diverse raw data. Transfer learning can transfer the knowledge from one specific task to relevant tasks with good performance. For example, in the task of predicting image memorability, Jing *et al.* [19] constructed connections between visual feature sets and higher level image attributes by transferring attribute knowledge from external sources to enhance representation ability of visual features. Fuzzy models are usually used for controlling nonlinear systems and clustering. For example, Zhao *et al.* [20] designed a set of adaptive fuzzy hierarchical sliding-mode controllers for a class of MIMO nonlinear time-delay systems by using fuzzy systems to approximate uncertain functions. Feng and Zheng [21] proposed an improved stability criterion of continuous Takagi-Sugeno fuzzy systems with time-varying delay which provide a powerful control methodology for nonlinear systems. The fuzzy c-means clustering algorithm (FCM) [22] displays good performance with regard to extracting text layer from images. Inspired by these observations and previous studies, we propose a novel approach for video text detection and recognition that detects text regions with a corner response feature map, verifies text regions with transferred deep CNN classifiers, and separates clean text layer from the background with a novel separation method based on FCM clustering.

The major contributions of our research can be summarized as follows:

- For text detection, we propose using the corner response feature map, which reflects the areas where text is present. Accurate and complete text regions can then be obtained after gray-scale morphological processing and adaptive threshold binarization. The proposed method is capable of detecting text in various languages and fonts against complex backgrounds.
- For text verification, we construct transferred deep CNN classifiers from VGG16 [23], ResNet50 [24], and InceptionV3 [25] with a series of strategies, such as layer concatenation and fine-tuning. This has achieved remarkable performance.
- For text extraction, we propose a novel FCM-based separation method to extract a text layer from a complex background. We use a five-dimensional feature vector that includes position and color information to depict each pixel for clustering. Compared with existing methods (such as Otsu [26] and K-means [27]), the text extracted using the proposed method is cleaner and more complete.
- We have built a test dataset containing 2,000 typical high-resolution video frames collected from various sources, including movies, cartoons, and TV shows. Among the available test datasets, the Microsoft common test set [28] is obsolete, and the TV news and YouTube test sets [29] have only a small amount of high-resolution video frames. The effectiveness of our approach was validated using the three public test datasets as well as our own test dataset.

The rest of the paper is organized as follows. Section II reviews related work. Section III presents the detection algorithm based on a corner response feature map, FCM-based separation algorithm, and the construction of transferred deep CNN classifiers. In section IV, we present the experimental results and a discussion. Finally, we draw conclusions in section V.

## II. RELATED WORK

Traditional approaches for video text detection and recognition can be divided into two general categories. The first category utilizes inter-frame information. For example, Shi *et al.* [30] combined discrete cosine transformation (DCT) and block matching methods between adjacent frames to extract text from videos. Multi-frame synthesis methods are based on the fact that video text can remain relatively invariant in contrast with the background in a given period. De Jesus *et al.* [31] identified embedded text in videos based on image regularization and video text persistence. Wang *et al.* [32] applied a multiple frame integration (MFI) method to minimize the variation of the background, and a time-based maximum (or minimum) pixel value search and sobel edge map are combined to detect video text. The key for multi-frame synthesis is the choice of frame number. If there are not enough frames for synthesis, the enhanced effect of the text will not be obvious. However, using too many frames will cause different texts to become mixed [33].

The second category is based on the connectivity of text, edge, and texture features. Yan and Gao [34] applied color clustering to the image, which allows the candidate text regions to be obtained from each color layer by connected component analysis. The texts are distinguished by a cascade Adaboost classifier and recognized by an OCR package in each color layer. The final recognition results are verified by the relationships among different layers. The contrast between the characters and the background is always high, which causes rich edges. Based on the characteristics, a lot of edge-based approaches have been proposed. For example, Zhao *et al.* [35] utilized edge information and sparse representation to localize video text. Although this kind of method is simple and feasible, the performance is not ideal when the background contains a large amount of edge information [36]. Meanwhile, many methods use various filters to extract texture features. Li *et al.* [37] proposed using Key Text Points (KTPs) for video text extraction, which are acquired using the three high-frequency sub-bands obtained from the wavelet transformation. KTPs simultaneously have strong textual structures in multiple directions. Shivakumara *et al.* [38] presented a method composed of wavelet decomposition and color features. The wavelet decomposition is applied on three RGB channels separately to obtain three high-frequency sub-bands for each channel. The average of the nine sub-bands is calculated to increase the gap between text and non-text, on which the Laplacian method is employed for text detection. Epshtein *et al.* [39] proposed a stroke width transform (SWT) operator for video text detection. For each pixel, the operator computes the width of the most likely stroke containing the pixel. In [12], Zhao *et al.* utilized corner points for video text detection. This method is robust for multiple languages and fonts. However, there exists a deficiency resulting from the parameter sensitivity and discreteness of corner points.

Traditionally, hand-designed features and corresponding algorithms have been designed for specific classification and regression tasks. For example, Wang *et al.* [40] extracted and compared shape features from the neutral files to compute 3D model similarity based on surface bipartite graph matching. Wang *et al.* [41] proposed a weighted sparse neighborhood-preserving projections approach to reduce dimensionality for face recognition. This not only incorporates more local discriminant information, but also puts a constraint on the number of non-zero reconstruction coefficients. Ren *et al.* [42] combined a generalized low-rank approximation of matrices with supervised manifold regularization to learn new features for drusen segmentation from retinal images. Jing *et al.* [43] integrated low-rank multi-view embedding and regression analysis into a unified framework for micro-video popularity prediction. Owing to the promising performance of the neural network for solving time-series analysis and classification tasks, some approaches have been proposed in recent years to employ neural networks to learn the representative features from the original data. For example, Jia *et al.* [44] introduced a genetic algorithm into the Elman neural network to improve the recognition precision and operation efficiency on nonlinear, dynamic, complex data. Zhu *et al.* [45] recognized characters by jointly exploiting CNN and bimodal image enhancement techniques. Delakis and Garcia [46] proposed an approach to detect and localize horizontal text lines from raw color pixels based on CNNs. Hu *et al.* [47] utilized the CNN classifier for text line verification and localization.

In this paper, we propose the corner response feature map as an improvement upon discrete corner points for video text detection. We construct transferred CNN classifiers with a series of strategies for text verification against various backgrounds, such as layer concatenation and fine-tuning. We also apply a novel FCM-based separation algorithm for text layer extraction from complex backgrounds. The experimental results demonstrate that the proposed method achieves remarkable performance compared with other state-of-the-art approaches.

## III. PROPOSED APPROACH

The proposed approach is composed of four steps: video decoding, text detection, candidate text line localization, and false text line elimination using a deep learning method. First, we utilize the OpenCV library to decode video into frames. Next, we use a corner response feature map detector to obtain candidate text regions. Because there may be multiple text lines in the candidate text region, we then further partition the candidate text lines using two alternative methods. For the first method, candidate text lines are partitioned through projection analysis onto the contours of candidate text regions. If the first method fails, we use a more complicated method, which employs an FCM-based separation method to extract the candidate text layer, converts it to the gray-scale image, and conducts the projection analysis to partition the candidate text lines. In the last step, false text lines are removed by our constructed transferred deep CNN classifiers. The true text

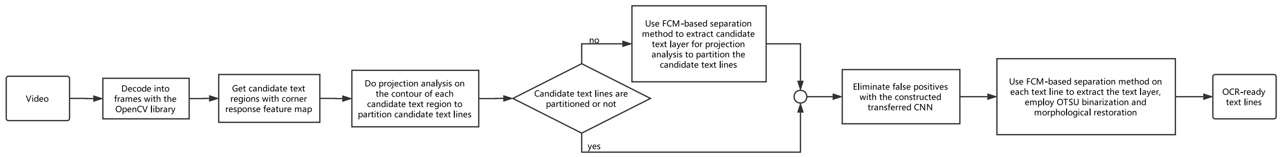



**FIGURE 1.** **Flowchart for the proposed method.**

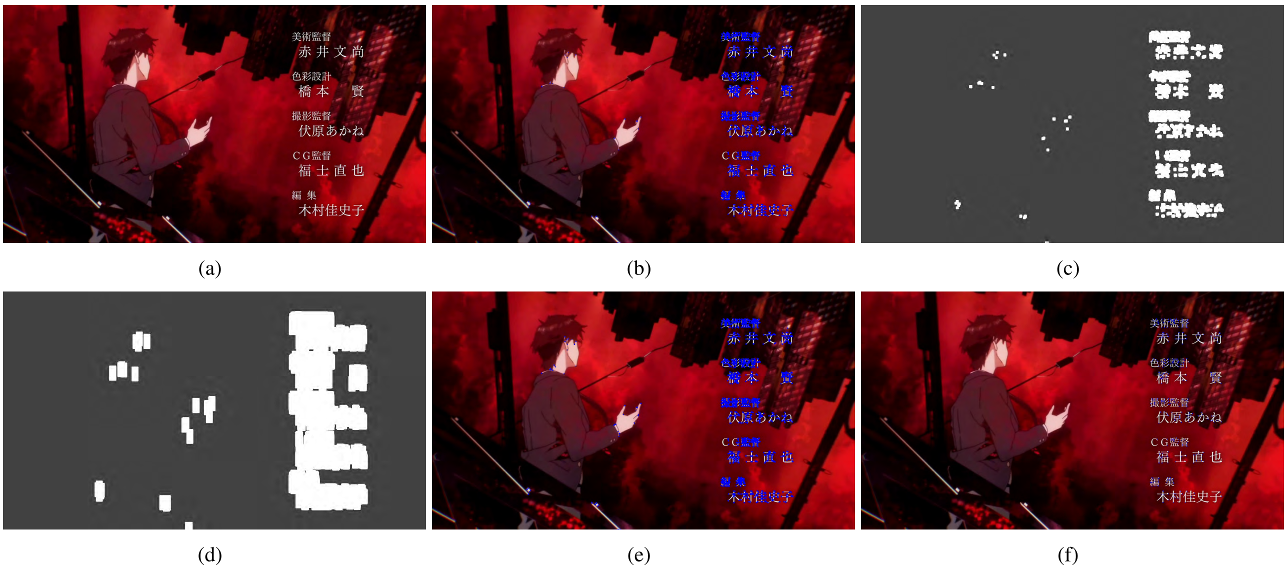



**FIGURE 2.** **Corner point distributions in the video frame. (a) Input frame. (b) Denser corner points in the text region than in the non-text region. (c) Formed candidate text region with insufficient dilation of corner points. (d) Formed candidate text region with excessive dilation of corner points. (e) Corner points distribution with a smaller *k*. (f) Corner points distribution with a larger *k*.**

lines then undergo FCM-based separation, Otsu binarization, and morphological restoration to obtain OCR-ready binary text. FIGURE 1 shows the flowchart for the proposed method.

### A. CORNER RESPONSE FEATURE MAP

Text in videos always provides supplemental information with good readability (especially the captions). The crosses of strokes in characters cause the generation of many corners. Video text always has a regular distribution of corner points, which the background generally does not have. Compared with other features, such as edge feature, corners are more stable and robust. The detailed mathematical derivation about corners was presented in [12]. Given a gray-scale image $I$, we take an image patch over the window $W(x, y)$, shift it by (u,v), and calculate the change produced by the shift as follows:

$$E(u, v) = \sum_{W} [I(x + u, y + v) - I(x, y)]^2. \tag{1}$$

The first-order Taylor expansion after omitting the Peano remainder term is used to approximate the shifted image as follows:

$$I(x + u, y + v) \approx I(x, y) + \begin{bmatrix} I_x(x, y) & I_y(x, y) \end{bmatrix} \begin{bmatrix} u & v \end{bmatrix}^T, \tag{2}$$

where $I_x$ and $I_y$ denote first-order partial derivatives in $x$ and $y$ directions, respectively. Substituting approximation (2) into (1) yields:

$$E(u, v) = \begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix}, \tag{3}$$

where $M$ is the following Hessian matrix:

$$M = \begin{bmatrix} \sum_{W} (I_x(x, y))^2 & \sum_{W} I_x(x, y) I_y(x, y) \\ \sum_{W} I_x(x, y) I_y(x, y) & \sum_{W} (I_y(x, y))^2 \end{bmatrix}. \tag{4}$$

If the two eigenvalues of $M$ are large and distinct positive values, a shift in any direction will cause a significant increase, and a corner can be determined. Instead of eigenvalues decomposition, Harris and Stephens [48] proposed the response function as follows:

$$f = \det(M) - k(trace(M))^2, \tag{5}$$

where $k$ is a tunable parameter. When $f$ is greater than the predefined threshold $R$, the corner is determined. As the most popular interest point detector, the Harris detector has many advantages, such as stability, reliability, and a simple calculation method.

Compared to non-text areas, there are denser corner points in text regions as shown in FIGURE 2(b). Based on these characteristics, many methods have been proposed for video text detection. In [12], the Harris corner detector [48] is used to extract corner points, and candidate text regions are formed by morphological dilation on the binary corner point image. The extracted corner points are discrete. To generate an appropriate candidate area, the choice of kernel and number of dilation are essential. If the dilation is not sufficient, the corner points cannot be connected to form a reasonable candidate text region as shown in FIGURE 2(c). If the dilation is excessive, it may cause misconnections between the text region and background as shown in FIGURE 2(d). In [49], the number of corner points plays a critical role in determining candidate text regions. However, the disadvantage of this kind of method is that the number of corner points is strongly influenced by parameter $k$ in (5), which is shown in FIGURE 2(e)(f). Although the advantages of the corner point include a simple calculation and regular distribution, the performance is limited by its discreteness and parameter sensitivity.

Inspired by [50], we adopt the continuous corner response feature map (CRM) for corner detection, which calculates the spatial derivative-based function of the source image as follows:

$$CRM = D_x^2 \cdot D_{yy} + D_y^2 \cdot D_{xx} - 2D_x \cdot D_y \cdot D_{xy}, \quad (6)$$

where $D_x$ and $D_y$ are the first-order derivatives of the source image $I$ in the $x$ and $y$ directions respectively, $D_{xx}$ and $D_{yy}$ are the second-order derivatives, and $D_{xy}$ is the mixed derivative. The corners exist in the local maximum area of CRM. As such, it is not necessary to determine the window and sensitive parameter $k$ in [48]. We use CRM to depict corners, and then utilize gray-scale morphological processing and an adaptive threshold to obtain corresponding areas (i.e., candidate text regions). The continuous CRM effectively overcomes the shortcomings of the discreteness and parameter sensitivity of corner points. Using this approach, we can obtain candidate text regions more completely and accurately.

## B. TEXT DETECTION

OpenCV is an open-source computer vision library that can be used to process videos and images. First, we use it to decode the video into frames through the cvQueryFrame function. Considering human visual characteristics, the video texts always last at least 2 seconds. Therefore, we grab one frame per second for video text detection so that no video text is missed.

We obtain the CRM of original frame according to (6). Regions of higher brightness always correspond to video texts. We then apply a series of gray-scale morphological operations to enhance the text regions and suppress the non-text regions. In the gray-scale morphological operation, erosion and dilation are defined as follows:

$$erode\,[f\,(x,y)\,,B] = \min_{(x',y')\in D_B} f\left(x+x', y+y'\right), \quad (7)$$


**FIGURE 3.** Sample video frames and corresponding CRMs after gray-scale morphological processing.

$$dilate\,[f\,(x,y)\,,B] = \max_{(x',y')\in D_B} f\left(x+x', y+y'\right), \quad (8)$$

where $f\,(x,y)$ denotes the original gray-scale image, $B$ is the structuring element and $D_B$ is the region in which $B$ lies. The close operation and tophat operation can be derived from erosion and dilation as follows:

$$close\,[f\,(x,y)\,,B] = erode\,[(dilate[f\,(x,y)\,,B\,]), B]\,, \quad (9)$$

$$tophat\,[f\,(x,y)\,,B] = f\,(x,y) - dilate\,[(erode[f\,(x,y)\,,B\,]), B]\,. \quad (10)$$

A close operation is utilized to remove the dark points that belong to the background in CRM, and then the tophat operation is used to enhance the bright text areas. The combination of the two operations makes the text regions more distinct and complete. The CRMs after gray-scale morphological processing are presented in FIGURE 3.

After the gray-scale morphological processing, Gaussian filtering is used to smooth the CRM, which contributes to the completeness of text regions to be detected. In order to form reasonable candidate text regions, we propose a binarization method with an adaptive threshold. At the left side

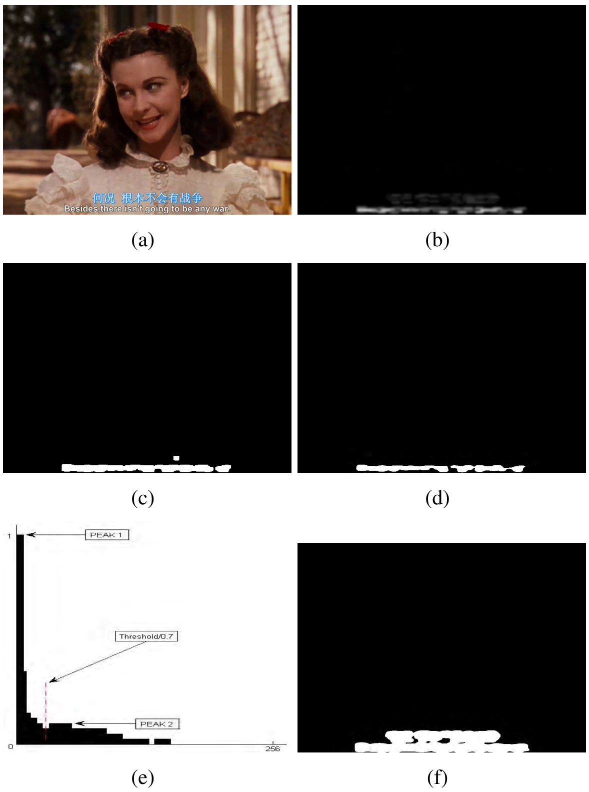



**FIGURE 4.** **Binarization results of CRM after preprocessing with two existing adaptive thresholds and the proposed adaptive threshold. (a) Original frame. (b) CRM after preprocessing. (c) Binarization result with OTSU. (d) Binarization result with an iterative threshold algorithm by Perez and Gonzalez [51]. (e) Normalized brightness distribution histogram of CRM after preprocessing. (f) Binarization result with the proposed adaptive threshold.**

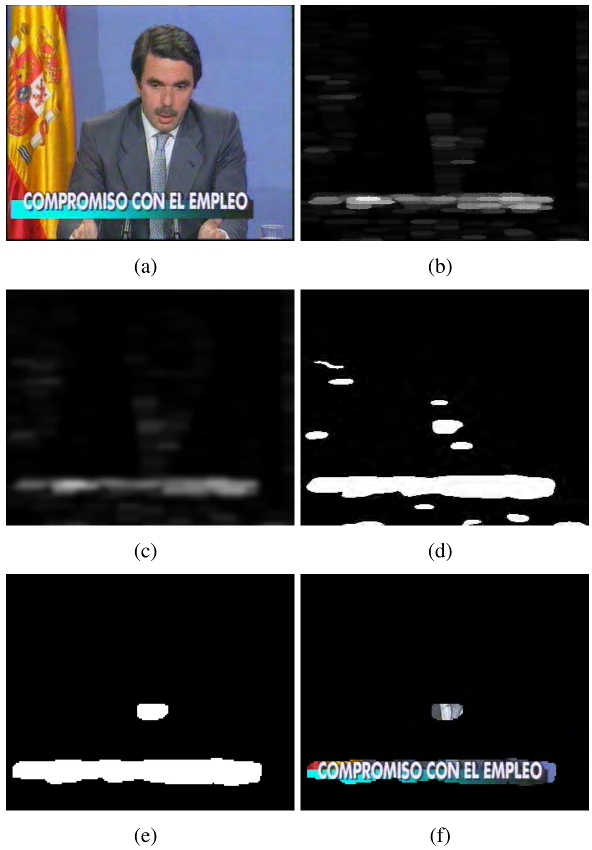



**FIGURE 5.** **Video text detection. (a) Original frame. (b) CRM after gray-scale morphological processing. (c) Gaussian smooth processing. (d) Binary image with the proposed adaptive threshold. (e) Contours of candidate text regions after sliding window filtering and area limitation. (f) Candidate text regions.**

of the normalized brightness histogram of the preprocessed CRM, there are always two local maxima. The second local maximum is far smaller than the first one, and the brightness in the neighborhood of the second local maximum varies more slowly as shown in FIGURE 4(e). The reason for this is that, generally speaking, a large number of pixels in non-text regions have lower brightness, and they form the first higher peak. A small number of pixels in text regions always have higher brightness, which form the second lower peak. Thus, there is a valley between the two peaks, which can be used to discriminate text regions from non-text regions. Based on our analysis and experiments, we propose the following heuristic adaptive threshold:

$$\begin{aligned} Th &= \min_i \; k \cdot i \\ s.t. \quad & i \in \{2, 3, \ldots, 255\} \\ & h[i] - h[i-1] > 0 \\ & h[i] - h[i-1] < T_W, \end{aligned} \tag{11}$$

where $Th$ is the adaptive threshold, $k$ is a positive parameter for adjustment, and $h[i]$ denotes the normalized number of pixels of the gray-scale value $i$, and $T_w$ is the presupposed variation threshold. $k$ and $T_w$ are selected via a coarse-to-fine grid search method. We performed a coarse grid search first. Once we identified an ideal region, a refined grid search was applied. Finally, we set $k$ to 0.7 and $T_w$ to 0.001. As shown in FIGURE 4, compared with the Otsu and an iterative threshold algorithm by Perez and Gonzalez [51] (FIGURE 4(c)(d)), the candidate text regions are more accurate and complete with our proposed adaptive threshold (FIGURE 4(f)).

In order to obtain a candidate text region with a more regular shape, we adopt a sliding window method. A $N \times N$ window is used to slide the binary image with an offset $\delta$ in the horizontal and vertical directions. In each window $W$, we calculate the number of non-zero pixels and conduct the following processing:

$$W(i,j) = \begin{cases} 255 & if\,(Weight\,(W) > T_n) \\ 0 & otherwise, \end{cases} \tag{12}$$

where $W(i,j)$ is the gray-scale value of pixel in the $(i,j)$ position of $W$, $Weight\,(W)$ denotes the number of non-zero pixels in the window, and $T_n$ is the threshold (which is proportional to the size of the window). The parameters $N$, $\delta$ and $T_n$ are determined via a grid search in a heuristic manner, ranging from 3 to 9 with an interval of 2, ranging from 1 to 5 with an interval of 1, ranging from $0.7N^2$ to $0.9N^2$ with an interval of


**FIGURE 6.** Candidate text line localization. (a) Original frame. (b) Localization result of the candidate text lines. (c) Candidate text region 1. (d) Corresponding contour 1. (e) Horizontal projection on contour 1. (f) Vertical projection on the first partitioned part. (g) Vertical projection on the second partitioned part. (h) Candidate text region 2. (i) Gray-scale image of candidate text layer extracted by the proposed FCM-based separation algorithm on candidate text region 2. (j) Horizontal projection on the gray-scale image. (k) Vertical projection on the first partitioned part. (l) Vertical projection on the second partitioned part.

$0.05N^2$ separately. Finally, we set $N$ to 5, $\delta$ to 2, and $T_n$ to 20. Using this approach, candidate text regions are more regular as shown in FIGURE 5(e).

We define the area of each candidate text region as the number of non-zero pixels in the region. Considering the size of the character that is readable for humans in different video formats, we set the threshold to 225 for SDTV (Standard Definition Television) and 625 for HDTV (High Definition Television). When the area is less than the threshold, the candidate text region is removed. After this simple screening, we obtain preferable candidate text regions as shown in FIGURE 5(e). A sample of the whole procedure for video text detection is shown in FIGURE 5.

### C. CANDIDATE TEXT LINE LOCALIZATION

In video text detection, adjacent text lines may adhere to each other due to preprocessing. The artificial texts (especially captions) in videos always have a horizontal orientation for readability. Based on this observation, we propose two alternative methods for accurately localizing the candidate text lines. Changes in the candidate text region contour acquired from the video text detection depict positional information for candidate text lines. For the first method, we use horizontal projection analysis on the contour as shown in FIGURE 6(e). The boundary always exists in the place where dramatic change occurs. Based on the sharp change of the horizontal projection, we can partition the candidate text region into candidate text lines horizontally. For each partitioned part, we use a vertical projection to relocate the vertical boundary as shown in FIGURE 6(f)(g). In this way, we accurately and efficiently localize the candidate text lines in markedly different lengths.

However, this approach does not perform satisfactorily when the candidate text lines have a similar length. To address this situation, we propose a novel FCM-based separation algorithm to extract the candidate text layer for projection analysis in order to partition the candidate text lines in the second method. FCM clustering is an improvement upon the hard c-means (HCM) clustering algorithm [27]. As described in [22], FCM determines $s$ fuzzy groups for $n$ data vectors and uses $u_{ij}$, the membership in [0, 1], to describe the extent to which the $j_{th}$ data vector belongs to the $i_{th}$ fuzzy group. The sum of $u_{ij}$ satisfies the following formula:

$$\sum_{i=1}^{s} u_{ij} = 1 \quad (j = 1, 2, ..., n). \tag{13}$$

The objective function is as follows:

$$J(U, c_1, ., c_i, ., c_s) = \sum_{i=1}^{s} \sum_{j=1}^{n} u_{ij}^m d_{ij}^2, \tag{14}$$

where $U$ is the set of $u_{ij}$, $c_i$ is the clustering center of the $i_{th}$ fuzzy group, $m$ is a weight exponent, and $d_{ij} = \left\| c_i - x_j \right\|_2$ is the Euclidean distance between $c_i$ and $x_j$ (the $j_{th}$ data vector). The necessary conditions for minimizing (14) are obtained using Lagrange multiplier method as follows:

$$c_i = \frac{\sum_{j=1}^{n} u_{ij}^m x_j}{\sum_{j=1}^{n} u_{ij}^m} \tag{15}$$


**FIGURE 7. Result of FCM clustering. (a) Candidate text region. (b) Background layer. (c) Border layer. (d) Candidate text layer.**

$$u_{ij} = \frac{1}{\sum_{k=1}^{s} (\frac{d_{ij}}{d_{kj}})^{\frac{2}{m-1}}}. \tag{16}$$

The FCM clustering procedure is designed as follows. First, we initialize the clustering centers. We then determine memberships according to (16). Next, we calculate the relevant termination conditions to determine whether the iterations are done. If they are not, the clustering centers are updated according to (15), and a new round of iterations is performed. We cluster the data vectors according to the final memberships. For FCM clustering, the number of clustering centers ($s$), the setting of the initial clustering centers, and the weight exponent $m$ are very important [52].

We use the position and color information of pixels to perform FCM clustering to extract the candidate text layer. First, we denote the candidate text area with $X = [w * X_p, X_c]$, which includes position information $X_p$, color information $X_c$ and a weight $w$. For each pixel, we typically use the horizontal coordinate $x$ and vertical coordinate $y$ to denote the position information, and we use RGB values to denote color information. Thus, we obtain $X_p = [X_x, X_y]$, $X_c = [X_r, X_g, X_b]$, and $X = [w * X_x, w * X_y, X_r, X_g, X_b]$. The pixels that make up the text, border, and background always have high, middle, and low brightness, respectively. Based on the observation and a large amount of experiments with a grid search, we found that the clustering achieved good performance when we adopted the following settings: clustering number of 3; initial centers at $\{0, 0, 50, 50, 50\}$, $\{0, 0, 120, 120, 120\}$, and $\{0, 0, 200, 200, 200\}$; $w$ of 0.2; $m$ of 2 (the weight exponent in the objective function (14)); and the FCM clustering iteration terminates when the total Euclidean distance between adjacent-iterative clustering centers is less than 0.02, or the number of iterations is more than 5,000. We call the three separated layers background layer, border layer, and candidate text layer, respectively, as shown in FIGURE 7(b)(c)(d).

For the candidate text area where the difference between the background and text is not sufficiently large, a one-off clustering is not enough. Characters in the same text region always have similar brightness. Based on the observation, we convert the candidate text layer into a gray-scale image denoted as $I$, where $I_k$ denotes gray-scale value of the $k_{th}$ pixel, $n$ denotes the number of pixels, the calculated mean is $I_{ave} = (\sum_{k=1}^{n} I_k)/n$ and the variance of the gray-scale image is $var = [\sum_{k=1}^{n} (I_k - I_{ave})^2]/n$. When the variance is less than 200 or the change in the variance is less than 100, the separation is terminated. The two thresholds are set via a grid search in a heuristic manner, ranging from 100 to 1000 with an interval of 100, ranging from 50 to 150 with an interval of 10 separately.

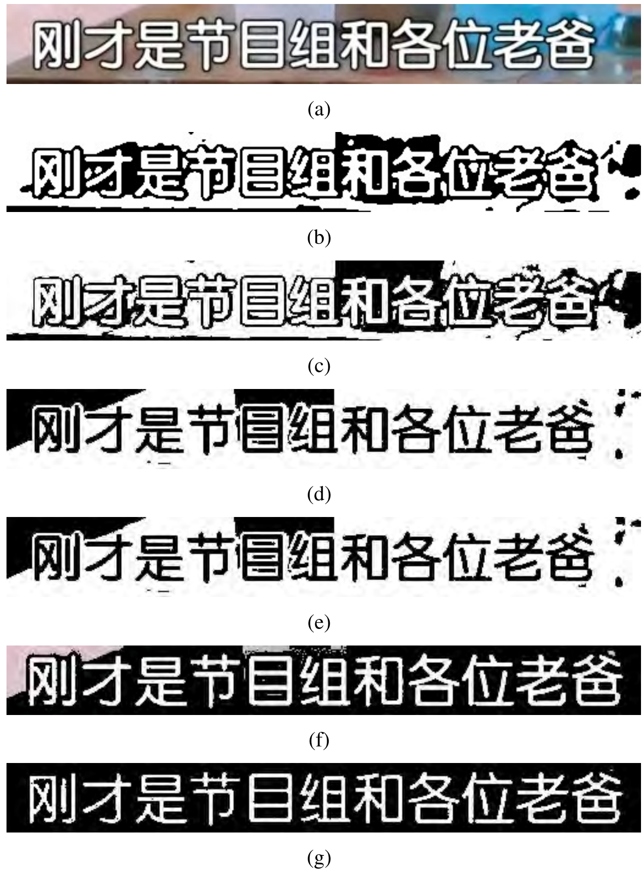



**FIGURE 8. Experimental results of text layer separation. (a) Text region. (b) Niblack [53]. (c) OTSU. (d) K-means clustering based on gray-scale information. (e) K-means clustering based on RGB color information. (f) The first FCM clustering. (g) The second FCM clustering.**

The pseudocode for the proposed FCM-based separation algorithm is summarized in Algorithm 1.

In order to evaluate the performance of the proposed FCM-based separation algorithm, we use several common text layer separation methods for comparison. The comparison algorithms include threshold methods, such as Otsu, and clustering methods, such as K-means clustering. The experimental results are shown in FIGURE 8. The proposed method achieves optimal performance. After the first FCM clustering, the variance requirement is not achieved, and the separation effect is poor. The second FCM clustering satisfies the variance conditions, and the text layer is well-separated.

After the FCM-based separation on the candidate text region (FIGURE 6(h)), we convert the candidate text layer image into a gray-scale image (FIGURE 6(i)) and perform a horizontal projection analysis as shown in FIGURE 6(j).

**Algorithm 1** FCM-Based Candidate Text Layer Separation

**Input:** Candidate text region image $I^0$, $s = 3$, $c_1^0 = \{0, 0, 50, 50, 50\}$, $c_2^0 = \{0, 0, 120, 120, 120\}$, $c_3^0 = \{0, 0, 200, 200, 200\}$, $w = 0.2$, $m = 2$, $var^0 = 0$, $p = 0$, $q = 0$
**Output:** $I^q$
**do**
  $p = 0$;
  **while** not clustering termination **do**
    Update $u_{ij}^p$ according to (16);
    Update $c_i^{p+1}$ according to (15);
    Check clustering termination conditions:
      $\sum_{i=1}^{s} \left\| c_i{}^{p+1} - c_i{}^{p} \right\|_2 < 0.02$ or $p > 4999$
    $p = p + 1$;
  **end while**
  Update $u_{ij}^p$ according to (16);
  Separate the pixels that belong to the third group to form $I^{q+1}$ according to $u_{ij}^p$;
  Convert $I^{q+1}$ into gray-scale image;
  Calculate $var^{q+1}$;
  Check the separation termination conditions:
    $var^{q+1} < 200$ or $abs(var^{q+1} - var^q) < 100$;
  $q = q + 1$;
**while** not separation termination

We partition the candidate text lines in a horizontal orientation according to the sharp change of the projection. For each partitioned part, we apply a vertical projection to obtain the vertical boundary as shown in FIGURE 6(k)(l). The result of localization is shown in FIGURE 6(b), in which the candidate text lines partitioned by the first method are bounded by red boxes, and those partitioned by the second method are bounded by white boxes. The two alternative methods localize the candidate text lines accurately.

### *D. FALSE TEXT LINE ELIMINATION BY A DEEP LEARNING METHOD*

Due to the complex background, the candidate text lines detected by previous procedures may still contain a lot of false positives. Therefore, we construct transferred CNN classifiers to eliminate false text lines. In current computer vision and image classification tasks, convolutional neural networks are taking on a more and more important role. The layer structure can provide scaling, tilting, and other forms of deformation invariance. This is because various convolutional kernels in layer extract different features, and the features are deepened gradually by the stacked layers until finally combined to form the discriminative feature. Low layers extract low-level features (such as edge feature), middle layers extract more complex features (such as texture features) and high layers extract the overall and discriminative feature for the final classification. The local receptive area and weight-sharing mechanism make the convolutional neural network similar to a biological neural network, which also reduces the number of weights and avoids the complexity of data computation.

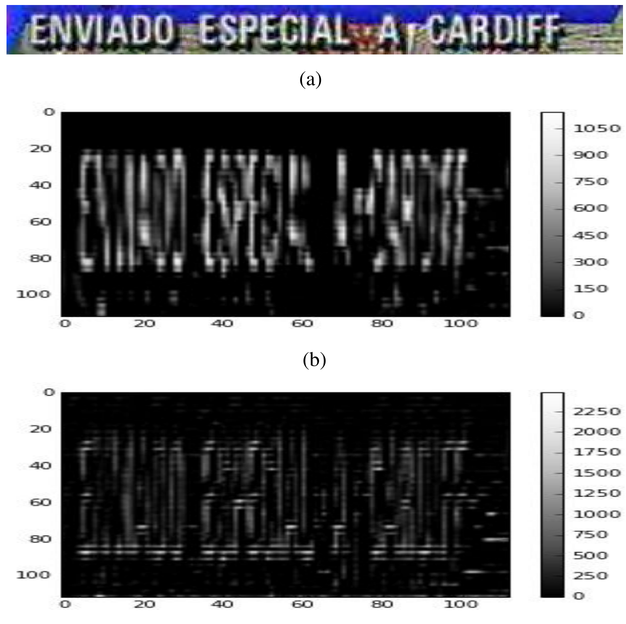



**FIGURE 9. Feature map samples from the Block2_conv1 of TVGG. (a) Text line. (b) Sample feature map 1. (c) Sample feature map 2.**

In the ImageNet large-scale visual recognition competitions in recent years [54], several typical CNN models achieved excellent performance, including VGG16, ResNet50, and InceptionV3. The VGG16 model structure is simple and effective, and it only uses a $3 \times 3$ convolution core to increase network depth. Unlike traditional sequential network architectures, such as AlexNet [55] and OverFeat [56], ResNet50 introduces identity mappings, which prevent the network from degrading with increasing depth. The InceptionV3 module acts as a "multiple-level feature extractor" with various convolution cores, and the outputs of different convolution cores are concatenated as input for the next layer. These models are highly generalizable for datasets other than ImageNet by means of transfer learning techniques and fine-tuning.

In this paper, we construct three CNN models transferred from VGG16, Resnet50, InceptionV3, and denote them TVGG, TRESNET, and TINCEPTION, respectively. The procedure for transfer learning of the three models is similar, and, therefore, we only explain in detail how to build and train the TVGG model. We remove the top three layers of VGG16 and use the rest as a deep feature extractor. Inspired by [57], the shallow layers capture local features, and the deep layer capture global features. In video text verification, some local features, such as edge feature, are also important. We extract and reshape the local features (FIGURE 9) from the Block2_conv1 layer as secondary features. We then add the global average pooling layer to pool both the shallow features and deep features, and we utilize the concatenation

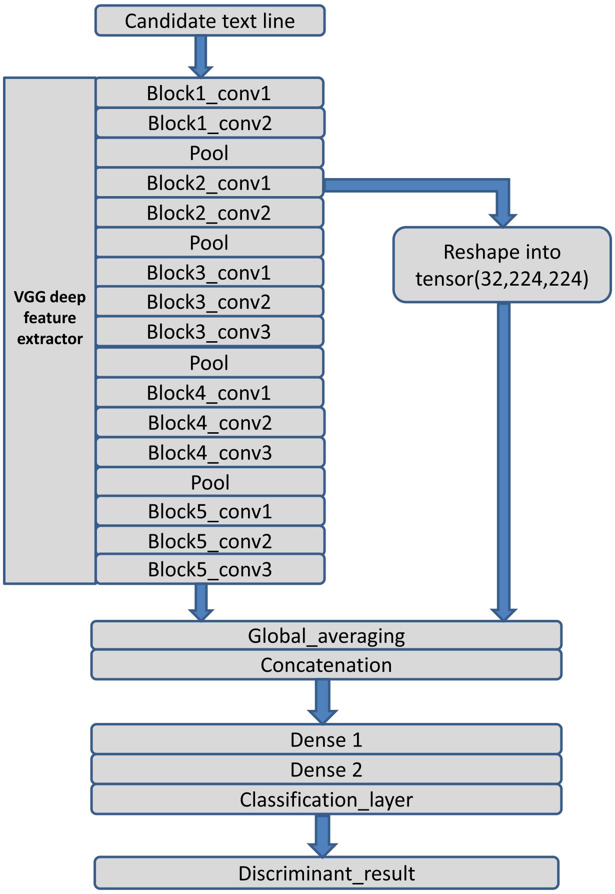



**FIGURE 10.** The proposed TVGG structure.

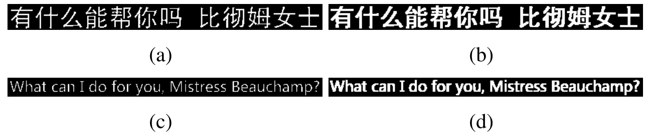



**FIGURE 11.** Acquisition of OCR-ready text lines from FIGURE 7(a). (a) The first text line after FCM-based separation and OTSU binarization. (b) The first text line after morphological restoration. (c) The second text line after FCM-based separation and OTSU binarization. (d) The second text line after morphological restoration.


**FIGURE 12.** Examples of the training-positive samples.

layer to concatenate them to form the discriminative feature. Next, two 4,096-dimensional dense layers with ReLU activation function and the final binary-class dense layer with softmax activation function are connected as the classifier. In this way, we build our classification model structure as shown in FIGURE 10. In Section IV, the experimental results show that the layer concatenation strategy enhances text verification ability for low-resolution video frames. We transferred the weights of the same layers from VGG16 to TVGG. The training is carried out by optimizing the categorical cross-entropy objective function with the Nadam optimizer [58] based on back propagation. The batch size is set to 64, and the initial learning rate is set to $10^{-5}$ via a coarse-to-fine grid search. To avoid overfitting, we use the dropout technique from [59] between the two new 4,096-dimensional densely-connected layers with an empirical dropout ratio of 0.5. The weights of the three new dense layers are initialized with the random initialization procedure proposed by Glorot and Bengio [60]. During fine-tuning, we first train the top three layers with a training set from our own dataset. After the classification accuracy in the validation set stop increasing, we train the whole network until convergence. The layer concatenation and fine-tuning allow the transferred CNNs to achieve superior performance, which has been proven by experiments on various test datasets.

Finally, we improve the quality of the verified text lines through several processing steps to make them more recognizable to OCR software. We use the proposed FCM-based separation method to extract the text layer and enhance the brightness contrast between text and background. Next, we use the Otsu method to binarize the text line, which achieves good performance as shown in FIGURE 11(a)(c). Because the text line may lose some edge pixels due to clustering, we utilize the binary morphological dilation to effectively bridge the gaps and remove burrs as shown in FIGURE 11(b)(d).

## IV. EXPERIMENTAL RESULTS

The performance of the proposed method is evaluated using three publicly available test datasets and our proposed test dataset. The three public datasets are the Microsoft common test set [28], TV news test set [29], and YouTube test set [29]. The first dataset contains 45 pictures of low resolution and poor quality, which is not up-to-date. The other two datasets contain high-resolution pictures. However, the size of the two datasets is too small to support further research. Our constructed dataset consists of more than 6,000 typical video frames of high resolution and high quality, about 25,000 text lines, and 42,000 negative samples. These frames are collected from various sources, including movies, cartoons, and TV shows. We sampled 2,000 video frames randomly and used them as the proposed test dataset. Some training positive samples are shown in FIGURE 12.

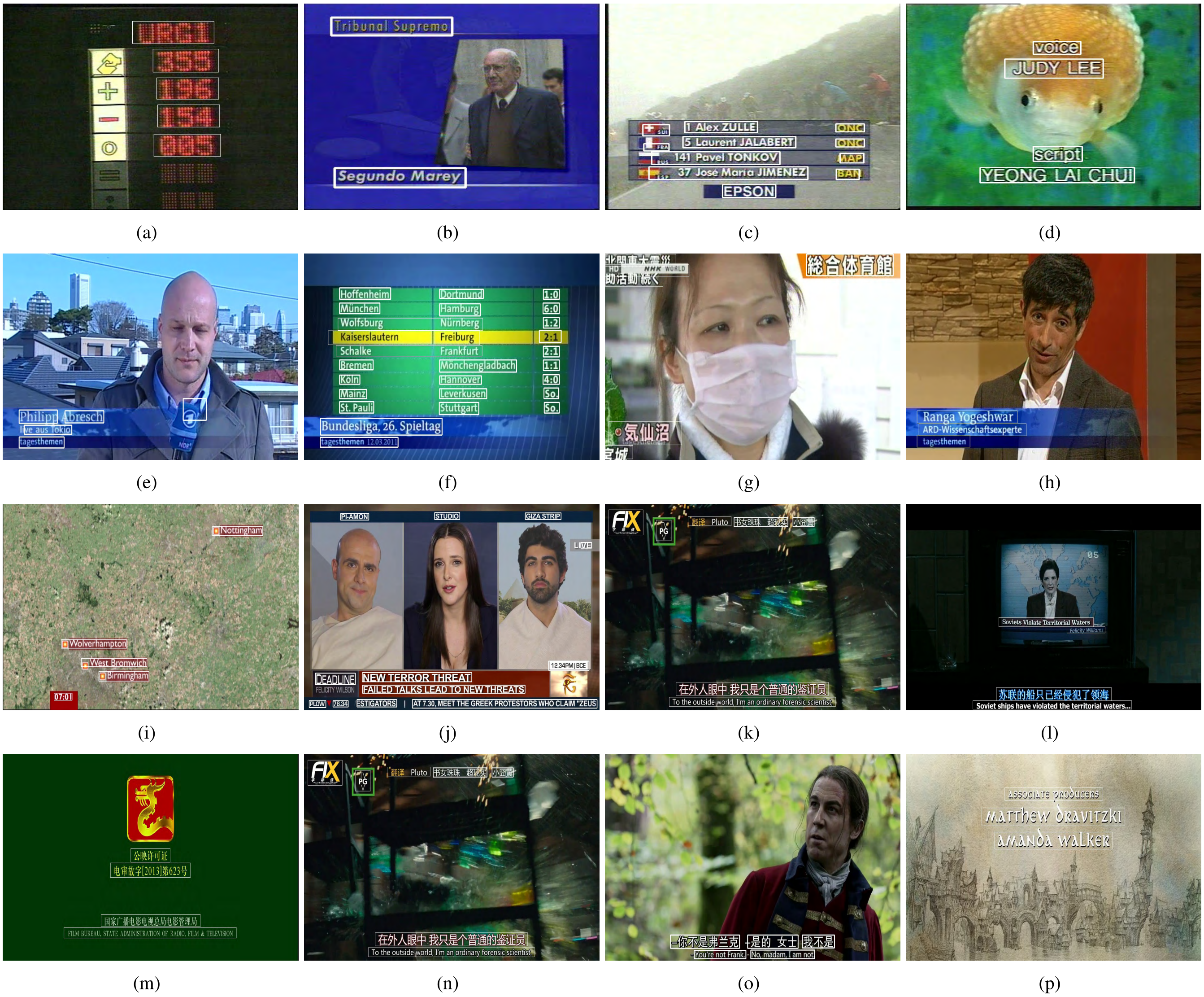



**FIGURE 13.** Examples of text detection results on various test sets. Detected text lines are bounded with white boxes.

We performed our experiments using Python with the Theano backend [61] and C++ with the OpenCV library. The hardware configuration includes an NVIDIA Geforce GTX 1080Ti with 11-GB GPU memory, an AMD Ryzen5 1400@3.20GHz×4 processor with 64-GB RAM. We resized the candidate text line images into the following input sizes: 224 × 224 for TVGG and TRESNET, 299 × 299 for TINCEPTION. In our constructed dataset, 2,000 images are randomly chosen as the test data. For the rest of the images, 80% are randomly selected for training, and the remaining 20% are selected for validation. We adopted the pixel-based evaluation method in [29], and the experimental results are shown in Table 1. The results show that our methods achieve good performance on a wide range of videos, and our TVGG based method performs best. Therefore, we chose the TVGG based method to compare with several state-of-the-art methods on three public test sets.

**TABLE 1.** Experimental results on the proposed test set.

| Method | Recall | Precision | F1-measure |
| --- | --- | --- | --- |
| Our TVGG based method | **0.88** | **0.83** | **0.85** |
| Our TRESNET based method | 0.88 | 0.82 | 0.85 |
| Our TINCEPTION based method | 0.87 | 0.82 | 0.84 |

For the three public test sets, we randomly assigned 80% of our constructed dataset to the training set and 20% to the validation set to train the TVGG model. We adopted the criteria described in [35] for comparison with other methods on the Microsoft common test set. The results are shown in Table 2. TVGG- denotes TVGG without the layer concatenation strategy. Our method achieves the second-highest precision and F1-measure score on the Microsoft common test set. The main reason for this is that the training set

**TABLE 2.** Performance comparison between our proposed method and several state-of-the-art methods on the Microsoft common test set.

| Method | Recall | Precision | F1-measure |
|---|---|---|---|
| Yang et al. [29] | 0.93 | 0.94 | 0.93 |
| Zhao et al. [35] | **0.94** | **0.98** | **0.96** |
| Gllavata et al. [62] | 0.9 | 0.87 | 0.88 |
| Shivakumara et al. [63] | 0.92 | 0.9 | 0.91 |
| Our TVGG based method | 0.91 | 0.96 | 0.93 |
| Our TVGG- based method | 0.9 | 0.93 | 0.91 |

**TABLE 3.** Performance comparison between our proposed method and several state-of-the-art methods on the TV news test set.

| Method | Recall | Precision | F1-measure |
|---|---|---|---|
| Yang et al. [29] | 0.86 | 0.81 | 0.83 |
| Hu et al. [47] | **0.92** | 0.90 | 0.91 |
| Our TVGG based method | 0.89 | **0.96** | **0.92** |

**TABLE 4.** Performance comparison on the YouTube test set.

| Method | Recall | Precision | F1-measure |
|---|---|---|---|
| Yang et al. [29] | 0.84 | 0.86 | 0.85 |
| Our TVGG based method | **0.86** | **0.88** | **0.87** |

of our constructed CNN is made up of high-resolution and high-quality images, which reduces the classification performance on the low-resolution dataset. Furthermore, the layer concatenation strategy is observed to enhance detection and recognition ability with an increase of 1% for recall and 3% for precision.

The pixel-based evaluation method in [29] is adopted for use on the TV news test set and YouTube test set. As shown in Table 3 and Table 4, we achieved the highest precision and F1-measure score on the TV news test set and the highest recall, precision, and F1-measure score on the YouTube test set. Compared with hand-designed features in [29], our constructed CNN can learn more discriminative features. The simple CNN proposed in [47] discriminated characters and determined the text line, whereas we verified the text line directly with the transferred deep CNN. By using the layer concatenation strategy and fine-tuning, our proposed transferred CNNs have stronger capabilities for feature representation and classification. FIGURE 13 shows some detection results for our method on the four test sets.

## V. CONCLUSION

In this paper, we propose a novel approach based on a corner response feature map and a transferred deep convolutional neural network for video text detection and recognition. The corner response feature map is used to detect candidate video text regions with a high recall. If the candidate text lines within the region have a distinct length, a projection analysis on the contour of the region is conducted to localize candidate text lines. Otherwise, an FCM-based separation algorithm is utilized to extract the candidate text layer, and then a projection analysis on the candidate text layer is conducted to localize candidate text lines. The constructed, transferred CNN model identifies video text lines accurately using the layer concatenation strategy and fine-tuning. The validated text lines undergo FCM-based separation, Otsu binarization, and morphological restoration to remove the background, and the final output is OCR-ready binary text. The experimental results show that our method performs well on recently produced videos containing various languages and fonts. In the future, we will improve the transferred CNN model to support more visual effects on text lines.

## REFERENCES

[1] Y. A. Aslandogan and C. T. Yu, "Techniques and systems for image and video retrieval," *IEEE Trans. Knowl. Data Eng.*, vol. 11, no. 1, pp. 56–63, Jan. 1999.
[2] H. Bhaskar and L. Mihaylova, "Combined feature-level video indexing using block-based motion estimation," in *Proc. Conf. Inf. Fusion*, Jul. 2010, pp. 1–8.
[3] W. Hu, N. Xie, L. Li, X. Zeng, and S. Maybank, "A survey on visual content-based video indexing and retrieval," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 41, no. 6, pp. 797–819, Nov. 2011.
[4] K. Jung, K. I. Kim, and A. K. Jain, "Text information extraction in images and video: A survey," *Pattern Recognit.*, vol. 37, no. 5, pp. 977–997, 2004.
[5] M. Khodadadi and A. Behrad, "Text localization, extraction and inpainting in color images," in *Proc. Iranian Conf. Elect. Eng.*, May 2012, pp. 1035–1040.
[6] A. Mosleh, N. Bouguila, and A. B. Hamza, "Automatic inpainting scheme for video text detection and removal," *IEEE Trans. Image Process.*, vol. 22, no. 11, pp. 4460–4472, Nov. 2013.
[7] K. I. Kim, K. Jung, and J. H. Kim, "Texture-based approach for text detection in images using support vector machines and continuously adaptive mean shift algorithm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 12, pp. 1631–1639, Dec. 2003.
[8] M. Cai, J. Song, and M. R. Lyu, "A new approach for video text detection," in *Proc. Int. Conf. Image Process.*, vol. 1, Sep. 2002, pp. I-117–I-120.
[9] T. Yusufu, Y. Wang, and X. Fang, "A video text detection and tracking system," in *Proc. IEEE Int. Symp. Multimedia*, Dec. 2013, pp. 522–529.
[10] X. Huang, "A novel video text extraction approach based on Log–Gabor filters," in *Proc. Int. Congr. Image Signal Process.*, vol. 1, Oct. 2011, pp. 474–478.
[11] P. Shivakumara, W. Huang, and C. L. Tan, "Efficient video text detection using edge features," in *Proc. Int. Conf. Pattern Recognit.*, Dec. 2008, pp. 1–4.
[12] X. Zhao, K.-H. Lin, Y. Fu, Y. Hu, Y. Liu, and T. S. Huang, "Text from corners: A novel approach to detect text and caption in videos," *IEEE Image Process.*, vol. 20, no. 3, pp. 790–799, Mar. 2011.
[13] B. Niu, H. Li, T. Qin, and H. R. Karimi, "Adaptive NN dynamic surface controller design for nonlinear pure-feedback switched systems with time-delays and quantized input," *IEEE Trans. Syst., Man, Cybern. Syst.*, to be published, doi: 10.1109/TSMC.2017.2696710.
[14] H. Wang, H. R. Karimi, P. X. Liu, and H. Yang, "Adaptive neural control of nonlinear systems with unknown control directions and input dead-zone," *IEEE Trans. Syst., Man, Cybern. Syst.*, to be published, doi: 10.1109/TSMC.2017.2709813.
[15] H. Wang, P. X. Liu, S. Li, and D. Wang, "Adaptive neural output-feedback control for a class of nonlower triangular nonlinear systems with unmodeled dynamics," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published, doi: 10.1109/TNNLS.2017.2716947.
[16] H. Wang, P. X. Liu, and S. Liu, "Adaptive neural synchronization control for bilateral teleoperation systems with time delay and backlash-like hysteresis," *IEEE Trans. Cybern.*, vol. 47, no. 10, pp. 3018–3026, Oct. 2017.
[17] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng, "End-to-end text recognition with convolutional neural networks," in *Proc. Int. Conf. Pattern Recognit.*, Nov. 2012, pp. 3304–3308.
[18] Z. Saidane and C. Garcia, "Robust binarization for video text recognition," in *Proc. Int. Conf. Document Anal. Recognit.*, vol. 2, Sep. 2007, pp. 874–879.
[19] P. Jing, Y. Su, L. Nie, and H. Gu, "Predicting image memorability through adaptive transfer learning from external sources," *IEEE Trans. Multimedia*, vol. 19, no. 5, pp. 1050–1062, May 2017.

[20] X. Zhao, H. Yang, W. Xia, and X. Wang, "Adaptive fuzzy hierarchical sliding-mode control for a class of MIMO nonlinear time-delay systems with input saturation," *IEEE Trans. Fuzzy Syst.*, vol. 25, no. 5, pp. 1062–1077, Oct. 2016.

[21] Z. Feng and W. X. Zheng, "Improved stability condition for Takagi–Sugeno fuzzy systems with time-varying delay," *IEEE Trans. Cybern.*, vol. 47, no. 3, pp. 661–670, Mar. 2017.

[22] J. C. Bezdek, W. Full, and R. Ehrlich, "FCM: The fuzzy C-means clustering algorithm," *Comput. Geosci.*, vol. 10, nos. 2–3, pp. 191–203, 1984.

[23] K. Simonyan and A. Zisserman. (2014). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: https://arxiv.org/abs/1409.1556

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[25] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 2818–2826.

[26] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-9, no. 1, pp. 62–66, Jan. 1979.

[27] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, Sep. 1999.

[28] X.-S. Hua, L. Wenyin, and H.-J. Zhang, "An automatic performance evaluation protocol for video text detection algorithms," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 4, pp. 498–507, Apr. 2004.

[29] H. Yang, B. Quehl, and H. Sack, "A framework for improved video text detection and recognition," *Multimedia Tools Appl.*, vol. 69, no. 1, pp. 217–245, 2014.

[30] J. Shi, X. Luo, and J. Zhang, "Dct feature based text capturing and tracking," in *Proc. Chin. Conf. Pattern Recognit.*, Nov. 2009, pp. 1–4.

[31] Â. M. de Jesus, S. J. F. Guimaraes, and Z. K. G. do Patrocinio, Jr., "Video text extraction based on image regularization and temporal analysis," in *Proc. IEEE Int. Symp. Multimedia*, 2011, pp. 305–310.

[32] R. Wang, W. Jin, and L. Wu, "A novel video caption detection approach using multi-frame integration," in *Proc. Int. Conf. Pattern Recognit.*, vol. 1, Aug. 2004, pp. 449–452.

[33] X. Huang, "Automatic video text detection and localization based on coarseness texture," in *Proc. Int. Conf. Intell. Comput. Technol. Automat.*, Jan. 2012, pp. 398–401.

[34] J. Yan and X. Gao, "Detection and recognition of text superimposed in images base on layered method," *Neurocomputing*, vol. 134, pp. 3–14, Jun. 2014.

[35] M. Zhao, S. Li, and J. Kwok, "Text detection in images using sparse representation with discriminative dictionaries," *Image Vis. Comput.*, vol. 28, no. 12, pp. 1590–1599, 2010.

[36] Q. Meng, Y. Song, Y. Zhang, and Y. Liu, "Text detection in natural scene with edge analysis," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2013, pp. 4151–4155.

[37] Z. Li, G. Liu, X. Qian, D. Guo, and H. Jiang, "Effective and efficient video text extraction using key text points," *IET Image Process.*, vol. 5, no. 8, pp. 671–683, 2011.

[38] P. Shivakumara, T. Q. Phan, and C. L. Tan, "New wavelet and color features for text detection in video," in *Proc. Int. Conf. Pattern Recognit.*, Aug. 2010, pp. 3996–3999.

[39] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2963–2970.

[40] J. Wang and H. Wang, "A study of 3D model similarity based on surface bipartite graph matching," *Eng. Comput.*, vol. 34, no. 1, pp. 174–188, 2017.

[41] Y. Wang, H. Zhang, and F. Yang, "A weighted sparse neighbourhood-preserving projections for face recognition," *IETE J. Res.*, vol. 63, no. 3, pp. 358–367, 2017.

[42] X. Ren *et al.*, "Drusen segmentation from retinal images via supervised feature learning," *IEEE Access*, vol. 6, pp. 2952–2961, 2018.

[43] P. Jing, Y. Su, L. Nie, X. Bai, J. Liu, and M. Wang, "Low-rank multi-view embedding learning for micro-video popularity prediction," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 8, pp. 1519–1532, 2018, doi: 10.1109/TKDE.2017.2785784.

[44] W. Jia, D. Zhao, Y. Zheng, and S. Hou, "A novel optimized GA–Elman neural network algorithm," *Neural Comput. Appl.*, pp. 1–11, Jul. 2017, doi: 10.1007/s00521-017-3076-7.

[45] Y. Zhu, J. Sun, and S. Naoi, "Recognizing natural scene characters by convolutional neural network and bimodal image enhancement," in *Proc. Int. Workshop Camera-Based Document Anal. Recognit.*, 2011, pp. 69–82.

[46] M. Delakis and C. Garcia, "Text detection with convolutional neural networks," in *Proc. Int. Conf. Comput. Vis. Theory Appl.*, 2008, pp. 290–294.

[47] P. Hu, W. Wang, and K. Lu, "Video text detection with text edges and convolutional neural network," in *Proc. Asian Conf. Pattern Recognit.*, Nov. 2015, pp. 675–679.

[48] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. Alvey Vis. Conf.*, vol. 15, no. 50, 1988, pp. 147–151.

[49] X.-S. Hua, X.-R. Chen, L. Wenyin, and H.-J. Zhang, "Automatic location of text in video frames," in *Proc. ACM Workshops Multimedia, Multimedia Inf. Retr.*, 2001, pp. 24–27.

[50] L. Kitchen and A. Rosenfeld, "Gray-level corner detection," *Pattern Recognit. Lett.*, vol. 1, no. 2, pp. 95–102, 1982.

[51] A. Perez and R. C. Gonzalez, "An iterative thresholding algorithm for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-9, no. 6, pp. 742–751, Nov. 1987.

[52] J. Nayak, B. Naik, and H. Behera, "Fuzzy C-means (FCM) clustering algorithm: A decade review from 2000 to 2014," in *Computational Intelligence in Data Mining*, vol. 2. New Delhi, India: Springer, 2015, pp. 133–149.

[53] W. Niblack, *An Introduction to Digital Image Processing*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1986, pp. 115–116.

[54] *Large Scale Visual Recognition Challenge (ILSVRC)*. Accessed: Jul. 13, 2018. [Online]. Available: http://www.image-net.org/challenges/LSVRC/

[55] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[56] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. (2013). "OverFeat: Integrated recognition, localization and detection using convolutional networks." [Online]. Available: https://arxiv.org/abs/1312.6229

[57] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 818–833.

[58] T. Dozat, "Incorporating nesterov momentum into adam," in *Proc. ICLR Workshop*, 2016, pp. 2013–2016.

[59] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[60] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. Int. Conf. Artif. Intell. Stat.*, 2010, pp. 249–256.

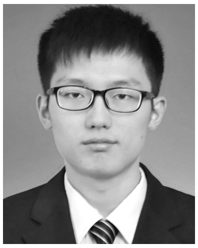
[61] R. Al-Rfou *et al.* (2016). "Theano: A Python framework for fast computation of mathematical expressions." [Online]. Available: https://arxiv.org/abs/1605.02688

[62] J. Gllavata, R. Ewerth, and B. Freisleben, "Text detection in images based on unsupervised classification of high-frequency wavelet coefficients," in *Proc. Int. Conf. Pattern Recognit.*, vol. 1, Aug. 2004, pp. 425–428.

[63] P. Shivakumara, T. Q. Phan, and C. L. Tan, "Video text detection based on filters and edge features," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jun. 2009, pp. 514–517.


**WEI LU** received the B.Eng. degree in electronic engineering and the Ph.D. degree in signal and information processing from Tianjin University, Tianjin, China, in 1998 and 2003, respectively.

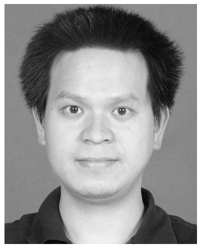
He is currently an Associate Professor with the School of Electrical and Information Engineering, Tianjin University. His teaching and research interests include digital filter design, digital multimedia technology, embedded system design, web application design, and pattern recognition. He is currently a Senior Member of the Chinese Institute of Electronics.



**HONGBO SUN** received the B.S. degree in electronic information engineering from Tianjin University, Tianjin, China, in 2012, where he is currently pursuing the M.S. degree with the School of Electrical and Information Engineering. His research interests involve multimedia content analysis.

**JINGHUI CHU** received the B.Eng. degree in radio technology, and M.Eng. and Ph.D. degrees in signal and information processing from Tianjin University, Tianjin, China, in 1991, 1997, and 2006, respectively.

She is currently an Associate Professor with the School of Electrical and Information Engineering, Tianjin University. Her teaching and research interests include digital video technology and pattern recognition.



**XIANGDONG HUANG** was born in 1979. He received the M.S. and Ph.D. degrees from Tianjin University, Tianjin, China, in 2004 and 2007, respectively. In 2009, he was with The University of Hong Kong, as the Visiting Scholar. In 2011, he was with the University of Macau, as a Research Assistant. In 2013, he was with the University of Delaware, as the Visiting Scholar. He is currently an Associate Professor with the School of Electrical and Information Engineering, Tianjin University. His research interests include filter design, and spectral analysis.

**JIEXIAO YU** was born in 1980. She received the M.S. degree from Nankai University in 2002, and the M.S. and Ph.D. degrees from Tianjin University, Tianjin, China, in 2005 and 2010, respectively. She is currently an Associate Professor with the School of Electrical and Information Engineering, Tianjin University. Her research interests include wireless positioning, filter design, and spectral analysis.