IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 23 November 2025, accepted 8 December 2025, date of publication 12 December 2025,
date of current version 18 December 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3643512

APPLIED RESEARCH

Real-Time Multilingual Video Subtitle Spotting on Resource-Constrained Devices

ILLYA DEGTYARENKO[1,2], (Member, IEEE), OLGA RADYVONENKO[1], (Member, IEEE), NAZARII TKACH[1,3], (Member, IEEE), VALERII SIELIKHOV[1], KOSTIANTYN SELIUK[1], AND OLEKSANDR IVANOV[1]

[1]Samsung Research and Development Institute Ukraine, 01032 Kyiv, Ukraine
[2]G.E. Pukhov Institute for Modeling in Energy Engineering, 03164 Kyiv, Ukraine
[3]National University of Kyiv-Mohyla Academy, 04655 Kyiv, Ukraine

Corresponding author: Illya Degtyarenko (i.degtyarenk@samsung.com)

ABSTRACT Accurate real-time on-device video subtitle spotting is essential for many applications, such as subtitle translation, text-to-speech conversion, video content comprehension. However, most video content providers encrypt or embed subtitles in ways that prevent direct text extraction, necessitating the use of Optical Character Recognition (OCR) for detection and recognition. Current state-of-the-art video text spotting methods are not optimized for real-time operation on edge devices. To address this challenge, this study introduces the specialized neural network architectures designed for on-device video content classification, subtitle tracking, detection and recognition. To enhance efficiency, the proposed neural network architectures employ advanced optimization techniques, including pruning and Quantization-Aware Training (QAT), significantly reducing memory and computational demands while maintaining high real-time performance on TV devices. Through rigorous testing and on-device end-to-end (E2E) evaluation, we achieved an impressive novel state-of-the-art E2E word recognition accuracy of over 97% across seven languages, with a low latency of under 150 ms per screen. The findings hold great potential for extending this technology to other platforms, including IoT devices and digital appliances.

INDEX TERMS Artificial intelligence, computer vision, text spotting, optical character recognition, subtitles, on-device.

I. INTRODUCTION

The rapid advancement of consumer devices, communication technologies, and video services has led to a significant transformation in how information is consumed by users [1]. Video traffic has become increasingly dominant, having had the strongest growth at 50 percent over the period from 2020 to 2024 [2], as on average, consumers spend about 3.7 hours daily watching online video content, though activities vary by generation [3]. For example, weekly consumption of any video news increased to 75% in 2025 with social video rising to 65%, whereas TikTok demonstrated significant short-form video growth [4]. Projections indicate video's dominance will continue, taking into account recent extremely fast changes shaping media landscapes toward user-generated videos. Another global trend is the implementation of video subtitling for content localization, making it accessible to global audiences [5], and captioning for better accessibility to the media content [6]. According to a recent poll, over half of Americans keep subtitles or closed captions turned on some (21%) or all (34%) of the time, especially younger people [7]. Access to these text data provides background for several scenarios, for example:

- accessibility services for reading subtitles aloud for visually impaired users who are unable to read text from the screen [8];
- real-time translation of closed captions or other on-screen text for users watching foreign-language content [5];

The associate editor coordinating the review of this manuscript and approving it for publication was Jiachen Yang.

- content understanding and indexing for search engines and automatic video summarization by extracting key textual information [9], [10];
- facilitating learning comprehension and cognitive load in educational video [11];
- interacting with text in Augmented Reality (AR) and interactive media, enhancing gaming and immersive experiences [12].

Subtitles are defined as a transcription of the screenplay in languages different from the language of the original audio track, whereas closed captions are defined as a transcription of the content of the original audio track [13]. Despite the fact that this data can be provided by the special interface in compact text format, most of the video content providers restrict access to all multimedia elements, including text elements like subtitles or closed captions. These text elements are typically embedded within the streamed video (burned-in) and can be extracted only by AI-based text spotting solutions.

Over the last few decades, the task of text-on-video spotting has gained considerable attention in the fields of computer vision and multimedia processing [14], [15], [16], [17] with applications ranging from content indexing to accessibility services. Text spotting from videos involves detection, tracking, and recognizing text that appears naturally within video frames (scene text) as well as overlaid text like subtitles. While substantial progress has been made in video text spotting [16], the challenge of achieving real-time, high-accuracy on resource-constrained devices remains an open issue. Current solutions generally rely on server-based cloud computing systems, where video data is uploaded to a remote server for processing. These solutions, while effective, are hindered by several drawbacks, including latency, privacy concerns, and dependency on internet connectivity. To overcome these limitations, there is a growing push towards on-device text spotting, where extraction occurs directly on edge computing devices or embedded systems [17]. This approach promises real-time performance, enhanced privacy, reduced latency, and better energy efficiency. The need for on-device text spotting solutions has become particularly critical with the increasing usage of smartphones and other edge devices. These devices are increasingly tasked with real-time video processing, but their limited computational power, storage capacity, and energy constraints present significant challenges for implementing deep learning models capable of robust text detection. While cloud-based systems can leverage large computational resources to process video content with higher accuracy, on-device systems need to strike a balance between efficiency and accuracy. Optimizing these systems for edge devices requires a deep understanding of model compression techniques, hardware-aware algorithms, and efficient resource management.

Despite the essential benefits of text extraction from video content, it is an increasingly challenging task due to the dynamic nature of video streams, varying lighting conditions, complex backgrounds, and text styles. In addition, the implementation of real-time text spotting on a device requires solving several problems caused by limited resources and architectural constraints.

Subtitle text represents a specialized case within text video spotting, characterized by several unique properties: temporal persistence with dynamic changes, high contrast visibility, and often consistent positioning within the video frame. In addition to standard on-device deployment challenges, subtitle text video spotting presents specific difficulties, including: distinguishing subtitles from other textual elements within the video stream, dynamic content handling (managing subtitle text that changes over time while maintaining recognition accuracy), dealing with complex backgrounds, handling artistic or stylized subtitle fonts that deviate from standard text formats, supporting multilingual content. The specialized nature of subtitle video text spotting requires a tailored approach that accounts for these unique characteristics while maintaining robust performance across diverse video content and taking into account the on-device deployment scenario.

This paper presents an in-depth exploration of on-device text spotting solutions, focusing on the methodologies and algorithms that enable real-time subtitle extraction from video content on resource-constrained devices. The main contributions of this work are summarized as follows:

1) a novel pipeline for video-text spotting is introduced. It includes the new elements: a content classifier and subtitle tracking. Implementations are based on a lightweight and efficient convolutional neural (CNN) architecture, harnessing the power of specific on-device chipset optimizations. These modules allow for the differentiation of content and detect subtitles' appearance in broadcasting videos.
2) enhanced approach for adjustment of the results of text line detection and classification is proposed. These lightweight approaches are based on the subtitles' style features extraction and processing.
3) the adaptation of the OCR convolutional recurrent neural network (CRNN) architecture for on-device implementation by the introduction of a text chunking procedure for the extraction of visual features and merging them for effective and accurate sequence prediction.
4) a novel on-device solution for text-on-video spotting is presented. It supports seven languages and provides background for real-time services of text translation and/or reading aloud.

The rest of this paper is structured as follows: Section II goes through the evolution of approaches for text spotting in video. The proposed methodology is described in detail in Section III. In Section IV we introduce the dataset strategy. Section V presents the experimental results and the performance evaluation, followed by a comprehensive discussion in Section VI. Finally, Section VII summarizes this research and provides a future outlook.

II. RELATED WORK

The task of detecting [18] and extracting text from video content has gained significant attention in recent years, owing to the growing demand for real-time text spotting in consumers' edge devices and multimedia services. Text-on-video spotting is an important subfield of computer vision, where the focus is on identifying [19], localizing, and recognizing [20], [21] text embedded in video frames. Early systems for text detection may be categorized into [22]: edge-based [23], [24] and texture-based [25]. Both approaches utilize a handcrafted set of features [26] for localization of video caption text. Most of modern systems are primarily designed for offline, cloud-based processing pipelines where video frames are processed on devices with high computational capability [27], [28] proposed a deep learning-based method for detecting subtitles in videos, which used Convolutional Neural Networks (CNNs) [29] for robust text localization under varying conditions such as text distortion, different fonts, and complex backgrounds [30]. Such cloud-based solutions come with significant latency [31], [32], as large video files need to be transmitted to remote servers for processing, which is unsuitable for real-time or live applications [33]. However, the increasing need for real-time processing, privacy preservation [21], and offline functionality on edge devices has driven a substantial shift toward on-device solutions [34]. There are two basic approaches for text-on-video spotting system realization:

- Serial architecture, when the system is decomposed into multiple modules (ML models) — text detection [35], [36], text tracking, text recognition [37], etc. Every separate module is tuned for each task and formed based on the specific architecture. With the advent of deep learning, CNNs and Recurrent Neural Networks (RNNs) rapidly supplanted these methods. A review of several related works is presented below.
- End-to-End (E2E) architecture, when the system that combines detection, tracking, and recognition [38] into a single ML model [16], [39], [40]. These modern approaches utilized Transformer architecture that inherently manage long-range dependencies and sequential context [39], offering improved performance on complex video scenes. Although Transformer-based solutions have set new accuracy benchmarks [41]. This approach has high potential for further development, but its extensive computational and memory requirements may pose significant challenges for immediate deployment on edge devices in most of practical use cases [42]. This constraint does not allow for the implementation of real-time text spotting scenarios for challenging setups with limited computational capability. The research and proposed method in this study are primarily focused on such a scenario.

As it was mentioned in previous Section, the popularity of video content has exploded with the rise of streaming platforms and social media, which have manifested in the intensification of consumption of TV and streaming services and an increasing role of video in education and distance learning. On the other side, subtitles make videos available all over the world in multiple languages.

Modern DL techniques are widely used for text recognition and processing [17], [43], [44], [45]. These technologies have had rapid development in the last two decades and have considerably improved performance both in terms of accuracy and latency [46], [47], [48]. Progress in hardware development allows running text detection and OCR methods on-device [34].

One of the possible approaches to text recognition in the video is based on existing image-based techniques [32], [49]. Although these approaches can achieve plausible accuracy, it would be erroneous to believe that they can be applied directly for computationally efficient on-device subtitle recognition in real-time video. Processing each video frame as an independent image leads to enormous computation costs; in addition, the processing of static images cannot provide the temporal information in the video, which is crucial for subtitle recognition. Taking into account that our goal is subtitle recognition with further voicing, the task of the utmost importance is to correctly classify subtitles and distinguish them from other texts simultaneously displayed on the screen: for instance, scene text in the video, channel names, etc.

In our previous work [17], we proposed and evaluated a computationally efficient and accurate end-to-end solution for real-time subtitle recognition based on an original feature-design approach and NN-based text binarization procedure followed by the CRNN for text recognition. Compared with that work, this manuscript introduces the following updates and novelties:

- A novel content classifier module was added. In pair with a postprocessing filtration for the text detector, it effectively eliminates the spotting of non-target (non-subtitle) text elements.
- A new text block adjustment approach was implemented, which allows to increase recognition accuracy owing to reducing the likelihood of text lines cutting by text detector ML model.
- The recognition model was adapted to handle both color and grayscale images, eliminating the UNet-based text binarization step. This change considerably improves subtitle recognition quality on complex backgrounds.

Comparative tests results for individual modules and the whole solutions are presented in Section V.

The presented text spotting techniques were implemented in commercialized TV services for subtitles detection recognition in real time [50]. The subtitle recognition results can be used for alternative audio track generation and content translation. This approach is patented [51]. Particularly, the patent describes future development aimed at the service usability enhancement due to audio content personalization.

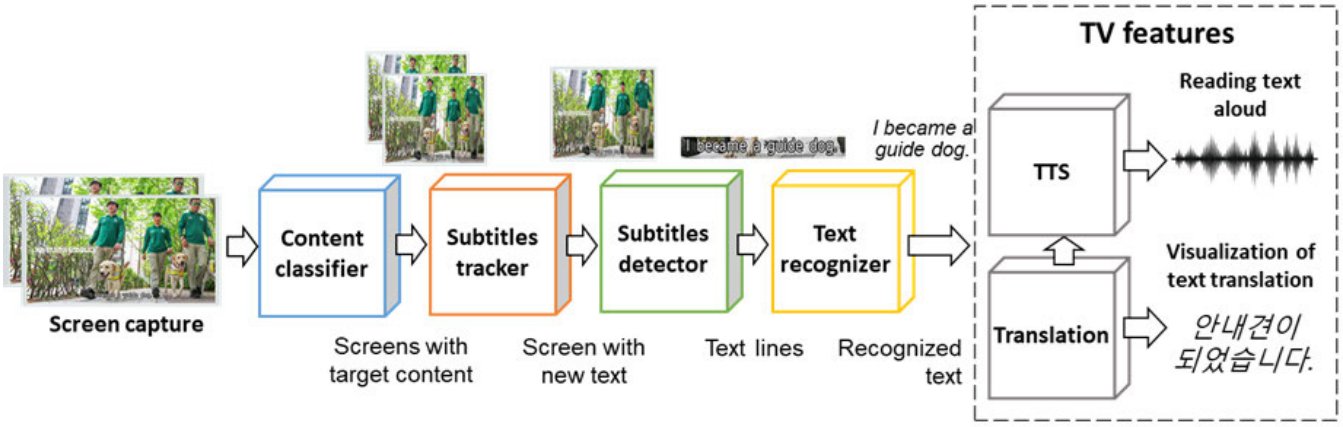

FIGURE 1. The overview of the proposed video subtitles spotting pipeline.

III. PROPOSED APPROACHES

We present here a lightweight on-device real-time solution for subtitle recognition and reading aloud on the fly. It consists of five main components: content classifier, subtitle tracker, text line detector, text recognition engine, and TTS (Figure 1). All components' architectures are based on the experiments and chosen from the point of the trade-off between accuracy and latency. In addition, we considered their applicability to on-device implementation. The details and structure of all modules are described in the next subsections.

A. CONTENT CLASSIFIER

The content classifier module is aimed at the filtration of input video frames. It selects only frames with target content: movies and other entertainment videos with subtitles. The screens with non-target content are skipped. Under non-target content, we consider cases when the screen contains non-subtitle text elements. It corresponds to news, sports, economics, advertisements, and other TV programs. Implementation of this module allows for prevention detection, recognition and sounding of non-subtitle text elements, that very important for correct user experience. This module is based on a lightweight CNN model, which takes as input resized to 300 × 300 pixels video frame. The model contains five convolutional blocks as a backbone and two fully connected layers with the softmax producing binary output (Figure 2). This ML model was trained using a specially collected dataset, which contains various examples of screenshots with target (movie, cartoon, entertainment) and non-target (news, sports, advertisements, etc.) content.

As the TV content has dynamic inertia (content is stable in a frame of some screen sequence) the special smoothing algorithm based on a simple moving average (SMA) filter was added to stabilize content classifier output. The last softmax layer of ML model normalizes the NN output to a probability distribution over predicted two classes p and this value is fed to the SMA filter. The smoothed value f is thresholded with λ to obtain the class *Output* value.

$$f_k = \frac{1}{N} \sum_{i=k-N+1}^{k} p_i, \tag{1}$$

$$Output_k = \begin{cases} 1, & \text{if } f_k > \lambda \\ 0, & \text{otherwise,} \end{cases} \tag{2}$$

where N is the SMA filter order.

The filter allows to minimize influence of the potential classification errors on the end-to-end solution accuracy. The following values of the parameters were selected $N = 15$, $\lambda = N/2$.

B. SUBTITLE TRACKER

The tracking module provides information about the appearance of subtitles of interest in the video. The core of the module is a CNN-based classification model that processes a tensor formed from two consecutive frames:

$$I^{B\times H\times W\times 2C}, \tag{3}$$

where B is the batch size, H and W are the spatial dimensions and C represents channels of the captured frame images (Figure 2). Given the positive label of subtitle appearance classification, the image is sent downstream to the overall pipeline of the solution, otherwise, it is discarded until the target class is detected.

This module is very sensitive to the specific frame rate of the video. To determine an optimal screen sampling rate, we need to find the trade-off between screen capturing latency and consumption of computational resources while taking into account the dynamics of subtitle changes in the video. Statistics on subtitles appearance and duration were collected from a large, diverse subtitled movie dataset (more than 100 movies). According to the data, the duration of more than 99% of subtitles falls is in the range of one to six seconds. Thus, in order to detect a change in subtitle correctly, at least two video frames must be captured within a specific and limited time window. A low sampling frequency

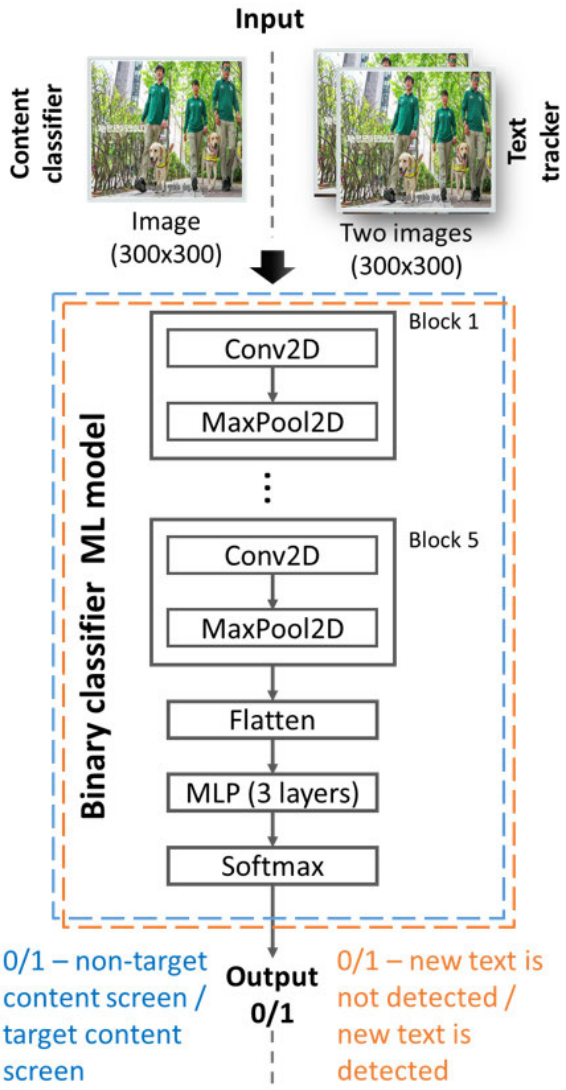

FIGURE 2. Architecture of Content classifier and Subtitle tracker.

results in the inability to capture the necessary subtitle appearance, whereas a high frequency imposes a heavy load on the computational resources of the device. Therefore, we heuristically selected sampling rate of 0.5 FPS as a reasonable trade-off.

For the subtitle tracking module, it is supposed that the concatenation along the channel axis of the closely sampled time frames is sufficient for the efficient detection of new subtitles. After that, the two consecutive frames are concatenated and fed to the CNN-based classifier. It has a similar architecture as Content Classifier. Tracker was trained on the synthetically generated dataset, which was formed by the following principles:

- class 1 (new subtitle detected) corresponds to the case when the first frame does not contain a new subtitle, and a new subtitle is present in the second frame;
- class 0 (no new subtitle): all other cases.

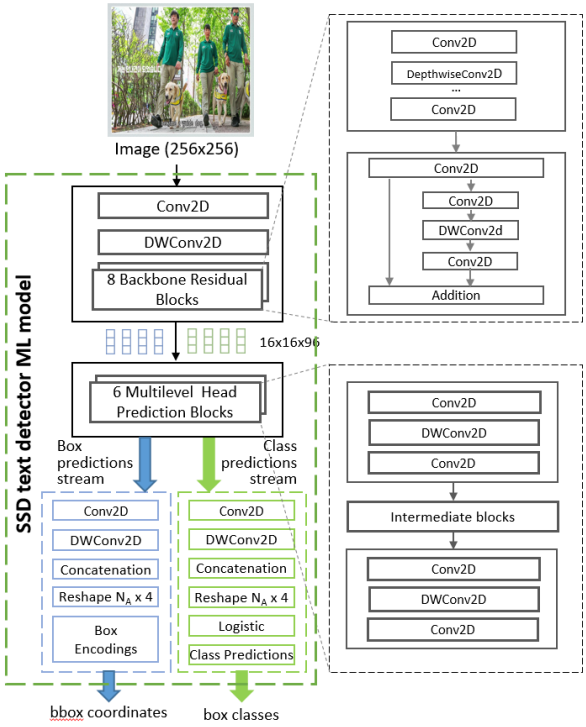

FIGURE 3. Architecture of text detector.

C. SUBTITLE DETECTOR

The text detection module is implemented based on the Single Shot MultiBox Detector (SSD) architecture [52]. The text detector network takes the captured frame image as an input and produces the list of text candidate boxes with the corresponding class confidences (Figure 3).

In general, the methods for text detection can be classified into segmentation-based and regression-based methods. While segmentation-based methods such as [53] can provide dense output per input pixel, it negatively affects the complexity of further post-processing. Due to a high computational complexity and latency, they cannot be used for on-device implementation for real-time video processing. Moreover, text subtitles come in rectangular regular form, which makes using the feature of high information retrieval capacity about the form of text non-relevant. Furthermore, the computational capability of target devices imposes limitations on using two-stage detectors [54].

For subtitle text line localization, we employ the SSD (head and predictor) with MobileNetV2 backbone architecture [55] similar to the TextBoxes++ model [56]. This architecture provides one of the most flexible compromises in terms of accuracy and latency. The implementation is based upon works [57] and was trained with quantization-aware training (QAT), which preserves high-quality location results despite rounding errors. The model was specifically ported for the Neural Processing Unit (NPU) in order to drastically reduce latency and Central Processing Unit (CPU) utilization. An input is resized to a 256 × 256 pixels screen frame.

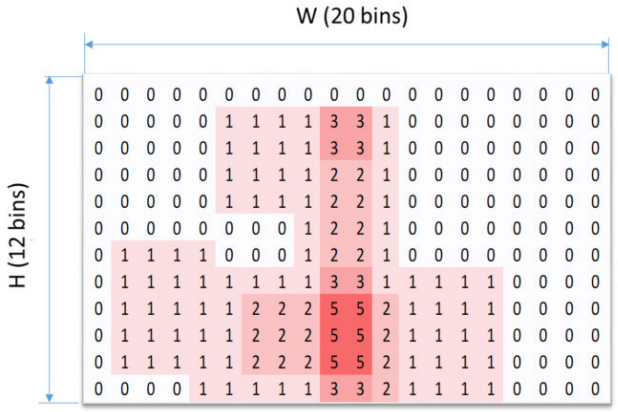

FIGURE 4. Heat map of subtitles text line centers location.

It should be noted that such size provides a great trade-off between the effective receptive field and the number of output anchors to process. Then image is normalized by subtracting the mean and dividing by the standard deviation. Due to the nature of subtitle text words' strict horizontal alignment, the data for text detection is labeled on the line level. While reducing the variety of supported text to the horizontal text lines, this inductive prior information makes word-level postprocessing of detected text obsolete. Aspect ratios of the SSD anchors are carefully fine-tuned for the specific type of text present in datasets and are selected by the k-nearest neighbor (KNN) algorithm. The statistics of location and width-to-height ratios collected from the sampled dataset provide valuable insights for the generation of synthetic training examples with high similarity with real-world data.

For better generalization during training, several augmentations were performed in run-time, such as jittering, color brightness change, contrast change, sharpening, etc. As the results of the localization, we obtain the list of coordinates of detected text lines' bounding boxes with the corresponding class confidences. Then both tensors are dequantized according to [57] and the denormalization procedure is applied to box coordinates. Afterward, intersection-over-union (IoU) and non-maximum suppression (NMS) thresholds are applied in order to select the most relevant candidates and suppress the detection of close false positive candidates with lower confidence. Based on the peculiarities of the subtitle text and the results from the detection neural network the IoU threshold of 0.15 and NMS confidence threshold of 0.5 are set.

To minimize the detection and recognition of non-target text elements (i.e., non-subtitles) we implemented several special post-processing rules. We examined typical subtitle locations using several thousand text lines extracted from real videos with various text styles (from different sources). Based on this analysis, a heat map 20×12 of subtitle text line center location was formed (Figure 4). If the detected text line has its central point in the zero zone of the heat map, the text line is skipped.

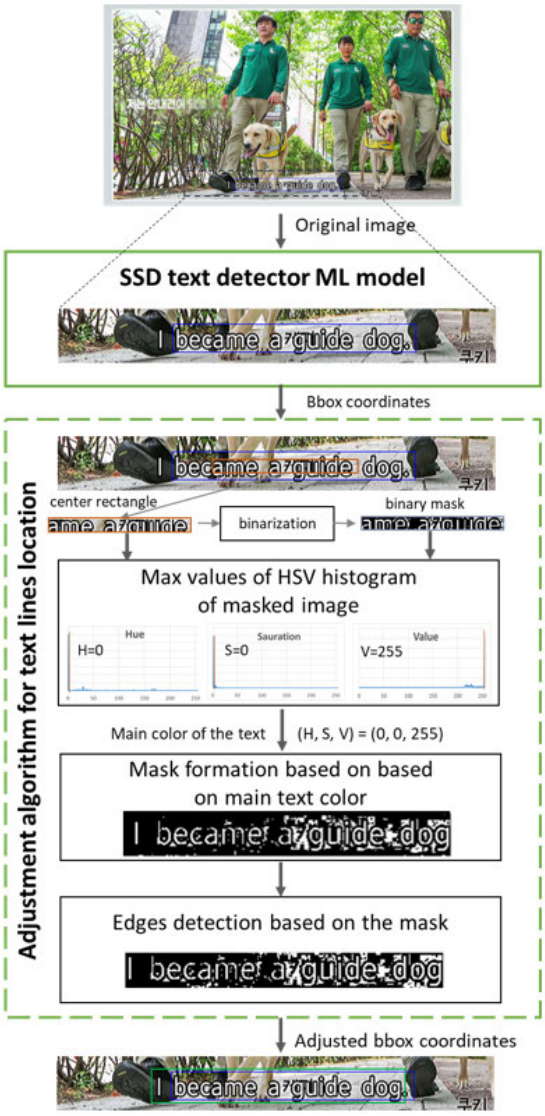

FIGURE 5. Subtitle coordinates adjustment approach.

The recognition engine is highly dependent on the subtitle text detection quality. Due to the sensitivity of the deep neural recognition network to noise artifacts present alongside the targeted text, it is necessary to have a bounding box with high precision. Therefore, the dedicated adjustment algorithm for text line detection localization results is proposed (Figure 5). The main idea of the proposed approach is to detect the most prevalent color of the subtitles and use features of that color for image filtering and bounding box adjustment. Efficiency estimation of the adjustment approach is presented in Section V. Adjusted text line boxes are normalized by a height of 48 pixels, and then are handled by the recognition model.

D. SUBTITLE RECOGNITION

There are a great number of DL approaches for image text recognition, which can be generally classified into

two groups: segmentation-based and segmentation-free methods. Segmentation-based methods extract rich feature maps and try to recognize distinct characters and subsequently merge results. Segmentation-free methods can be consequently subcategorized into full encoder-decoder methods and methods based on Connectionist Temporal Classification (CTC) [58], [59]. While encoder-decoder methods generally provide high accuracy in various OCR tasks, the sequence-to-sequence models have a far greater receptive field and are less prone to be affected by information loss among extracted timeframes. Encoder models that use attention, such as TransformerOCR [58] are heavily dependent on a resource-exhaustive global attention mechanism, which in the current state of hardware technology, makes them inapplicable for wide use on Edge and mobile devices. CTC-based recurrent models [59] such as Multi-Dimensional Long Short Term Memory (MDLSTM) [34] and CRNN [60] are far more flexible and are especially lightweight, which presents them as primary candidates for effective on-device usage. It is important to note that due to the regular rectangular form of the subtitle text, any advantage of the MDLSTM model compared to BiLSTM is minimized and the accuracy results are almost identical. From the latency standpoint, the two-dimensional (2D) approach used in MDLSTM is also less preferable to BiLSTM due to higher latency. In our proposed solution, we use CRNN-CTC architecture (Figure 6) with inference using a probabilistic Language Model and a token-passing algorithm [61].

According to evaluation results, the precision of CRNN-CTC using max-decoder gives about 1.5% WRR degradation, compared to the token-passing algorithm. The token passing algorithm has a multiplicative dependency on the length of the sequence [61] and has up to 4x higher latency. On top of the extracted visual features, the bidirectional Gated Recurrent Unit (BiGRU) architecture is placed, which has higher computational efficiency compared to LSTM.

For grammar correction, the class-based trigram language model is applied. The dictionary of the language model is about 100k words. Both results of the recognition neural network and the language model are linearly weighted in order to produce balanced output results. For the CNN we use MobileNetV3-small architecture without five last layers and modified strides reducing the length of the sequence by eight times along the X-axis.

In essence, all CRNN-CTC approaches perform CNN-based feature extraction, where an output resulting in vertical columns of feature maps, which are then fed as timeframes to the RNN model with the maximum likelihood of character from the predefined alphabet output per each time frame with the CTC loss, measuring the training error.

The input image for our CNN-BiGRU architecture [17] is firstly divided into equal non-overlapping chunks of 48×192 pixels. The feature extraction is performed per chunk and concatenated before it is fed to the BiGRU. It is important to note that in comparison with other sliding window convolutional approaches, our method proposes

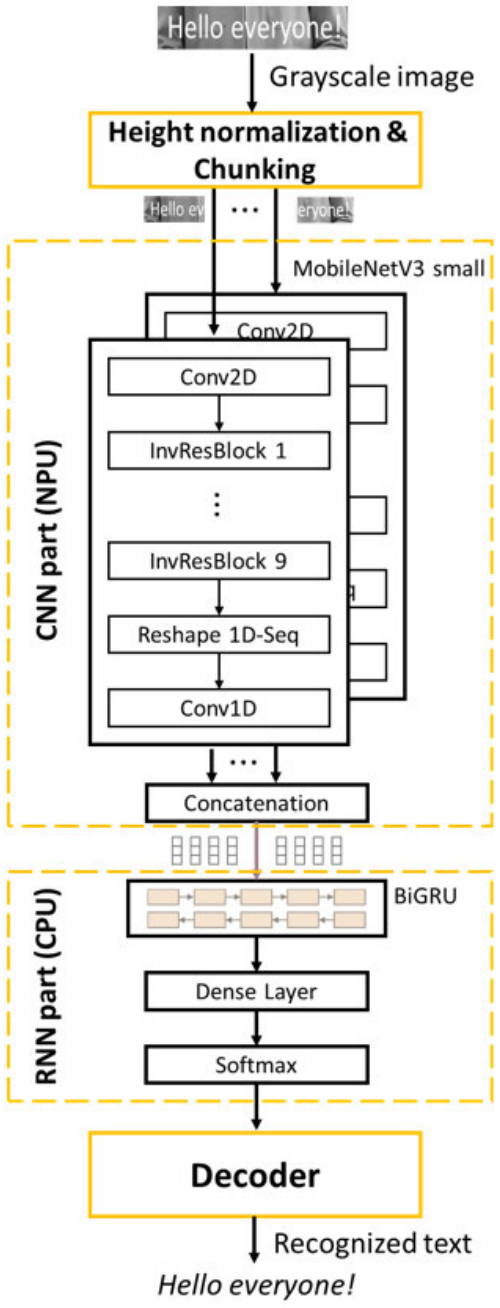

FIGURE 6. CRNN recognition architecture.

to use non-overlapping extracted patches, thus forcing the model to use non-repetitive context information from different chunks with zero shared information. It also benefits the execution time by skipping the already-seen part of the input image.

E. SUBTITLE VOICING

The results of subtitle recognition are voiced by the TTS engine: when the TTS is on, the original voice is automatically muted. The developed Audio Subtitle TV service

uses the TTS engine which is embedded into Tizen 9.0. In addition, the proposed pipeline solution is suitable for any general proprietary and non-commercial TTS engine. One of the main potential problems of dubbing real-time video using a synthetic voice is desynchronizing video and audio. This is caused by the accumulation of voicing delay implied by the difference between the pace of the real speech in the video and the speed of synthetic speech, as the frequency of subtitles change and their sizes can significantly variate. To overcome this problem, we propose an adaptive algorithm for the control of TTS speech speed. The algorithm uses information about current TTS queue to adjust TTS speech speed:

$$s_i = s_{i-1} + q_i, \tag{4}$$

where s_i is the TTS speed for the current i subtitle; q_i is the current size of the TTS queue.

IV. DATASETS

To acquire the training data for the DL models in our solution, we utilized synthetic datasets generated using our custom-built data generator. By incorporating open-source video footage and reference information about subtitles (such as text, style, and layout parameters), the generator produces videos and/or images with embedded subtitles along with corresponding annotations. The annotations contain information about the generated subtitles: time, placement, text, and style. This approach was used for dataset formation for training ML models for subtitle tracking, detection and recognition. We use 90/10 division of datasets for training and validation data. For independent testing purposes, we utilized specifically designed videos, which will be discussed in the subsequent section. The datasets are summarized in Table 1 and further detailed below:

1) A dataset for content classifier contains images of two classes: screens with target content and screens with non-target content. As we focus more on target content, the size of the target class is about 2 M frames taken from movies and other entertainment open-source videos, and the size of non-target content is about 0.6 M frames taken from news, sport, economics and other TV programs.
2) A dataset for subtitle tracker contains samples of two classes: class 0 (new subtitles appear) and class 1 (there is no new subtitle). Each sample is a pair of consecutive images taken from the annotated video. A sample corresponds to class 1 when the first image does not contain a new subtitle, and a new subtitle is present in the second image. All other cases correspond to the class 0.
3) For a text line localization dataset we take a screen frame (image) from the annotated videos and create a corresponding JSON file that contains useful metadata: subtitle layout parameters, and reference text.
4) For recognition models training there are six datasets for each target language were generated. The datasets contain synthetic images of subtitle text lines and text annotations (reference data). The part of the English dataset was added to the training data for other supported languages.

TABLE 1. Training datasets.

Module	Language	Dataset mln. img.	NPU/CPU model size, MB
Content Classifier	universal	2.6	0.27 / –
Tracker	universal	1.1	0.23 / –
Detector	universal	2.1	3.2 / –
Recognition	English	1.8	0.68 / 2.6
Recognition	Korean	2.3	0.68 / 3.6
Recognition	Spanish	1.4	0.66 / 2.8
Recognition	French	1.7	0.69 / 3.1
Recognition	German	2.0	0.69 / 3.1
Recognition	Italian	1.6	0.69 / 2.9
Recognition	Portuguese	1.7	0.69 / 2.9

V. EXPERIMENTS AND RESULTS

The evaluation of the proposed system was performed on TV sets using specifically designed independent video test sets for six supported languages. Every test set contains 64 videos (five minutes each) with burned-in subtitles in eight different styles. We have selected the most common styles which differ in text color (white/yellow), font (Arial-bold/Courier-bold) and background (black/transparent). Examples of the subtitles with these styles are shown in Figure 7.

All performance metrics were estimated by using commodity visual display device with Samsung NQ8 AI Gen3 chipset [62], which contains CPU and NPU modules. To minimize processing latency, memory and CPU consumption, all ML models, except RNN and decoding part of recognition, were quantized and ported to the NPU. Asymmetric affine quantization [57] is primarily used for classification networks due to its robustness and simplicity. Dynamic fixed point quantization [63] is more suitable for preserving the precision of convolutional backbone networks for recognition. This is important for providing coarse and fine-grained features to the RNN networks in the text recognition pipeline. Information about all ML model sizes is presented in Table 1. The total size of the database (DB) with all solution components for one language is less than 7.5 MB.

There are two main functional characteristics of the solution: subtitle recognition accuracy and processing latency. Both characteristics dramatically influence the user's perception of quality. The results of the research on the effect of these attributes are presented below.

For the evaluation of subtitle recognition quality, common character and word recognition rate metrics (CRR, WRR) [64] were applied. The CRR is based on the concept of Levenshtein distance, where we count the minimum number of character-level operations required to transform the ground

TABLE 2. On-device subtitle recognition accuracy metrics, %.

Language	CRR	WRR	CRR*	WRR*
English	99.24	98.34	95.76	94.89
Korean	98.9	97.17	94.58	92.83
Spanish	99.69	98.81	98.4	97.42
French	99.58	98.68	98.46	97.49
German	99.54	98.27	97.01	95.69
Italian	99.58	97.89	98.27	96.91
Portuguese	99.84	99.46	98.56	98.02
Average	**99.48**	**98.37**	**97.29**	**96.17**

truth text into the OCR output:

$$CRR = \frac{N_c - S_c - D_c - I_c}{Nc}, \tag{5}$$

where N_c is the number of characters in the reference text; S_c – the number of substitutions; D_c – the number of deletions; I_c – the number of insertion.

$$WRR = \frac{N_w - E_w - U_w - R_w}{Nw}, \tag{6}$$

where N_w is the number of words in the reference text; E_w – the number of words with recognition errors; U_w – the number of undetected words; R_w – the number of redundant (wrongly detected) words in recognition.

The punctuation and case information were discarded due to their minimal effect on end-user experience. Also, corresponding E2E metrics (CRR*, WRR*) were estimated. The E2E metrics take into account the performance of all solution modules and fully correspond with the user's perception and reflect their expectations for the overall performance.

Test results for seven supported languages and target chipset are presented in Table 2. The average WRR* metric for all target languages is 96.17%. This value is consistent with state-of-the-art OCR solutions [32], [49]. In addition, we have investigated the influence of different subtitle styles on recognition accuracy (Figure 7).

For independent evaluation of the solution we have formed a dedicated test dataset using videos from BOVText benchmark [65]. Twenty-six videos with subtitles were selected according to the tags "category" = caption, "language" = English. The list of selected test videos is presented in Table 3.

Direct benchmarking of the complete system was not possible due to the absence of publicly available subtitle text video spotting solutions. Particularly, the context classifier and subtitle tracking modules are unique and cannot be directly compared with the references [32], [49]. Therefore, we evaluated the main components of our pipeline separately, using the subset of open benchmark (Table 3).

For performance estimation of subtitle tracker, the Multiple Object Tracking Accuracy (MOTA) metric [66] was used. Since, in subtitle video spotting, only the moment of the first appearance of a new subtitle matters, the metric was

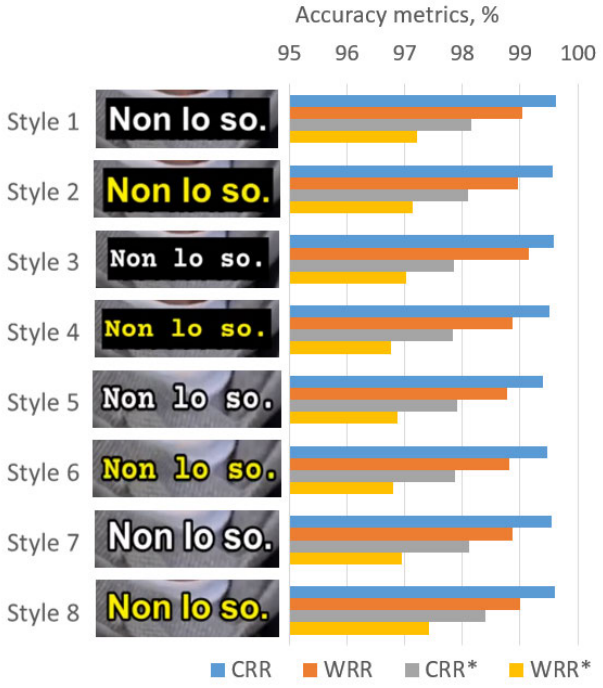

FIGURE 7. Evaluating subtitle recognition accuracy across diverse styles of subtitle.

TABLE 3. Test videos selected from BOVText dataset [65].

Test video name	Duration, seconds	Subtitles, number
Cls2_Cartoon_video34	30	3
Cls2_Cartoon_video54	30	5
Cls11_Movie_video79	30	11
Cls11_Movie_video97	30	11
Cls11_Movie_video102	30	11
Cls12_Interview_video10	30	8
Cls12_Interview_video12	30	5
Cls12_Interview_video22	30	5
Cls12_Interview_video32	30	14
Cls13_Introduction_video30	30	12
Cls14_Talent_video23	30	16
Cls14_Talent_video55	30	3
Cls14_Talent_video60	13	2
Cls15_Photograph_video45	26	4
Cls16_Government_video43	6	2
Cls17_Speech_video57	30	11
Cls17_Speech_video68	30	13
Cls19_Fashion_video47	30	10
Cls19_Fashion_video60	30	11
Cls20_Campus_video5	30	10
Cls21_Vlog_video87	30	4
Cls31_Eating_video28	30	12
Cls31_Eating_video34	30	8
Cls31_Eating_video42	30	7
Cls32_Unknow_video1	30	6
Cls32_Unknow_video3	30	11

calculated according to the formula:

$$MOTA = 1 - \frac{\sum_t (FN_t + FP_t + IDSW_t)}{\sum_t GT_t}, \tag{7}$$

TABLE 4. Subtitle tracker test results.

Solution	FP,%	FN,%	IDSW,%	MOTA
Tracker [17]	3.5	32.9	20.8	0.46
Our	1.0	20.8	3.0	0.75

TABLE 5. Subtitle detector test results (mAP).

Solution	IoU=0.5	IoU=0.75	IoU=[0.5:0.95]
Text detector [17]	0.92	0.31	0.44
Our	0.91	0.62	0.58

where t is the frame index, GT_t is the total number of ground truth objects (unique subtitles); FN_c – the number of false negatives (misses), when a ground-truth unique subtitle is not detected by the tracker; FP_c – the number of false positives, where the tracker identifies an unique subtitle that does not exist in the ground truth; $IDSW_c$ – the number of identity switches, where a tracker incorrectly swaps a unique subtitle with an already existing (non-unique) subtitle.

As a baseline we used the model described in [17]. Both subtitle tracking models have a similar architecture. However, in this study we propose to feed the tracker with two consecutive screenshots instead of three, as in [17]. As the screen capturing frequency is 0.5 FPS, this update reduced the subtitle detection delay by 500 ms. The results are presented in Table 4. It should be noted that, functionally only *FN* cases are critical (missing subtitle). Other types of tracking errors (*FP* and *IDSW*) mainly increase processing load but do not necessarily cause quality degradation. The *IDSW* errors are filtered out by checking for duplicates in recognition results, and *FP* detections correspond to processing frames that do not contain subtitles.

The principal difference between the subtitle detector from [17] and the proposed solution is in subtitle coordinate adjustment algorithm presented in Figure 5. To evaluate the detectors we built a test set of 215 screenshots, each containing a unique subtitle derived from the videos in Table 3. We employed mean Average Precision (mAP) [67] as the evaluation metric. The results are summarized in Table 5. Implementation of the proposed subtitles coordinate adjustment algorithm increases mAP@IoU = 0.75 twofold (from 0.31 to 0.62) and raises mAP@IoU = [0.5:0.95] by 0.14 (from 0.44 to 0.58). More accurate text line detection reduces the influence of background artifacts on recognition results and consequently improves overall recognition accuracy.

For a comparison of text recognition results, several benchmark on-device solutions were selected: the mobile version of PaddleOCR [68] and EasyOCR [69]. Both chosen OCR solutions are adapted for the edge computing. Using the same test set (text lines with subtitles from the videos listed in Table 3), we report the recognition results in Table 6. Our text recognition model demonstrates the highest recognition accuracy while the model size is five times smaller than that of the competing OCR systems. In addition to the benefit of the proposed approach, this advantage stems from training it specifically on subtitle data, while the other OCR solutions are designed for generic text recognition.

TABLE 6. Benchmarking of on-device text recognition.

Solution	CRR,%	WRR, %	Model size, MiB
PaddleOCR	99.4	95.03	16.0
EasyOCR	95.12	84.15	14.4
Our	99.39	**98.58**	**3.2**

E2E latency alongside the corresponding metrics from each module of the our solution for the English language are presented in Figure 8. E2E latency was estimated as a delay between the screen capturing start moment and the moment of obtaining the results of subtitle recognition. The achieved average latency is less than 150 ms. It meets the requirements for mouth-to-ear delay for real-time services, which should be less than 400 ms [70].

Since to our best knowledge, this is the first system where the video is processed on the fly on a device, while in previous works [32], [49] the input of the system is a video file and latency is not considered as a crucial constraint, we are not able to compare them directly.

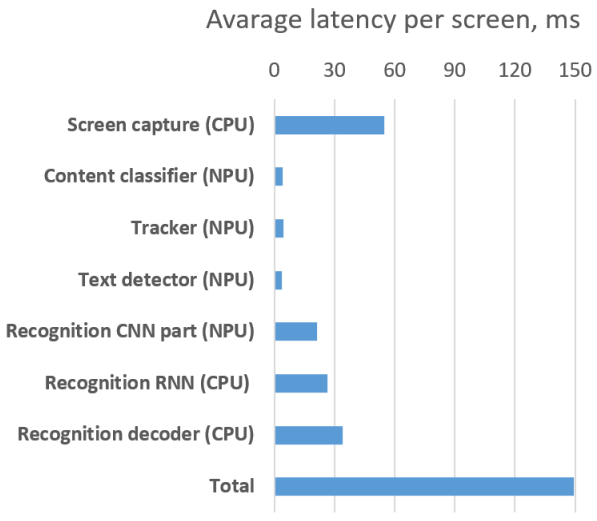

FIGURE 8. Latency of solution components per screen.

For on-device implementation, the computational constraints pose a significant importance regarding CPU usage and memory consumption. The aggregate CPU consumption for all main components (except screen capture) of the solution is less than 0.7%. The average memory consumption does not exceed 24 MB and the peak value is less than 36.5 MB.

Compared with our previous work [17] the described innovations, combined with the new training datasets generation

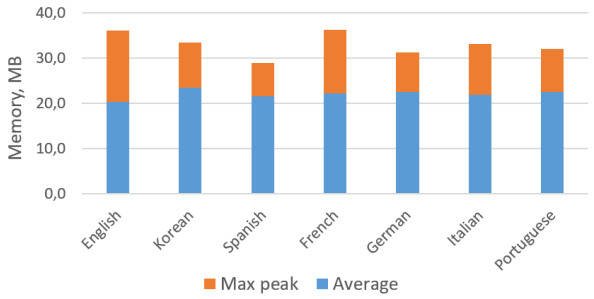

FIGURE 9. Memory consumption of the solution for different languages.

strategy, yield substantial gains: E2E WRR was improved from 92.1% to 94.9% for English language, E2E latency was decreased from 350 ms to 150 ms. In addition, this study introduses multi-language subtitle recognition for seven languages, enabling real-time translation and audio narration, while the earlier system was limited to English subtitles. Overall, the paper presents novel research contributions that not only outperform existing solutions in accuracy and latency but also extend practical applicability to commercial TV devices and other edge platforms.

VI. DISCUSSION

The presented evaluation results allow further analysis and insights regarding solution limitations and optimization opportunities.

The subtitle recognition accuracy is dependent on subtitle style and the peculiarities of language. The research on subtitle style influence on recognition accuracy indicates that the transparent background of subtitles causes slight WRR* metric degradation due to a higher percentage of undetected text lines on transparent backgrounds. Detection and recognition issues primarily arise from interference between white subtitles and complex screen backgrounds.

Additionally, the complexity of the target language impacts recognition results and memory consumption. Under the complexity, the whole set of important factors for accurate text recognition is meant, including the number of supported symbols (code-table size), cardinality of the supported word dictionary and sensitivity of the prediction result upon the text detection inaccuracies. From this point of view, the Korean language presents the challenge due to its larger code table size, agglutinative word formation, and support for multiple languages and scripts within subtitles. As the real Korean subtitles may contain English words (names, brands, emails, etc.), bilingual Korean-English recognition models were trained. These models use the code table of 137 symbols, which is about 20–25% larger than for any other target recognition language. The text detection accuracy in Korean is far more influential on the overall end-to-end results. It should be noted that the current solution achieves a significant leap forward in the domain of text recognition, taking into account the constraints and provides a basis for the continual improvements in that area of research.

The presented approach has several limitations connected with subtitle styles and locations. In particular, subtitles with the moving (rolling) text elements are not supported. Rolling subtitles appear gradually (from top to bottom or from left to right) and during the screen capture, part of text element can be corrupted or not visible (Figure 10a), so this part cannot be recognized properly. Correct processing of such cases requires a higher sampling frequency and a special tracking procedure. All this leads to significantly larger CPU consumption and processing latency, which can not be acceptable for the real-time on-device solutions.

The proposed text lines detection and adjustment approaches suppose that the color of the text line will be constant. Therefore, the text lines with multi-color elements can be processed incorrectly (Figure 10b). The issue with the multi-color text lines in frame of one screen was minimized by fine-tuning of subtitles style controlling algorithm parameters and by including multi-color text lines to the training data for the text detector. Some subtitles can be undetected due to low contrast of text to complex background or due to the small length of a text line (Figure 10d). The issue is caused by the limitations of the text detector model.

To minimize detection and recognition of non-target text elements (non-subtitles) we implement several special post-processing checking rules. One of the rules provides a comparison of the locations of the detected text lines. In case the center of the detected text line gets into the zero zone of the heat map (Figure 4), this text line will be skipped. Therefore, the subtitles with non-typical locations are not spotted (Figure 10c).

There are cases where non-subtitle text elements (movie credits, titles, comments, etc.) have a similar style and location with subtitles. These elements can be overlapped

FIGURE 10. Problematic cases of subtitles spotting. (a) Issue with capture of rolling subtitles. (b) Multicolor subtitles. (c) Untypical location of the subtitles. (d) Small subtitles. (e) Subtitles overlapping with other text elements. (f) Non-subtitles texts.

with subtitles (Figure 10e) or they can be detected and recognized instead of subtitles (Figure 10f). All this leads to errors in spotting results. In addition, in the frame of non-target content videos (news, advertisements, weather, sports, etc.), many text elements can be presented on the screen. Most of the screens with such content are skipped by the Content Classifier ML model. However, some of these screens have similar to normal target content (movie, entertainment) layouts. Therefore, the text elements on such screens can be spotted by mistake.

VII. CONCLUSION AND FUTURE OUTLOOK

In this work, we have presented a computationally efficient approach to on-device subtitles spotting, thus enabling the real-time reading and/or translation of burned-in subtitles. The proposed solution is specifically aimed at alternative audio track synthesis in multiple languages and demonstrates high performance. It is particularly valuable for visually impaired individuals or others who cannot read subtitles displayed on the screen.

The proposed architecture includes several innovative solutions: content classification, subtitle tracking, text line detection and classification, and an optimized recognition engine for on-device deployment. The solution currently supports seven languages: English, Korean, Spanish, Italian, French, German and Portuguese.

E2E on-device performance evaluation demonstrated an average latency of 150 ms, which satisfies the requirements for mouth-to-ear delay for real-time services. At the same time, the achieved average E2E word recognition accuracy of 96.17% in average across all six supported languages establishes a novel state-of-the-art for on-device implementation.

The future directions of video subtitle spotting include the support of new languages, implementing automatic subtitle language classification to provide different scenarios for multilingual content processing, and building the unified framework for both scene text and subtitle processing.

The visual cues derived from the proposed lightweight text detection and content classification modules, combined with consistent tracking history, can serve as a foundation for integration of additional features (facial expressions, scene and speaker content, etc.). It may enable richer semantic understanding and benefit visual and linguistic cohesion during real-time processing for multiple downstream tasks, e.g. content summarization, question answering. etc.

One more point for future development is a multi-modal (OCR + audio) approach for speaker-adaptive subtitles reading. It supposes changing the TTS voice parameters based on information about language, current active speaker, speech tempo and emotions. This will allow to make reading subtitles aloud more natural and informative.

The proposed approach presents promising opportunities for extension to other edge platforms, such as IoT devices and digital appliances. The results delivers new possibilities for video content localization and distribute accessibility for visually impaired users.

REFERENCES

[1] J. K. Chalaby, "The streaming industry and the platform economy: An analysis," *Media, Culture Soc.*, vol. 46, no. 3, pp. 552–571, Apr. 2024.

[2] Ericsson. (2025). *Ericsson Mobility Report Jun. 2025*. [Online]. Available: https://www.ericsson.com/en/reports-and-papers/mobility-report/reports/june-2025

[3] Deloitte. (2025). *2025 Digital Media Trends: The Rise of Hyperscale Social Video*. [Online]. Available: https://www.deloitte.com/us/en/insights/industry/technology/digital-media-trends-consumption-habits-survey/2025.html

[4] N. Newman, R. Fletcher, C. Robertson, K. Eddy, and A. Schulz. (2025). *Digital News Report 2025*. [Online]. Available: https://reutersinstitute.politics.ox.ac.uk/sites/default/files/2025-06/DigitalNews-Report2025.pdf

[5] A. Desai, R. Alharbi, S. Hsueh, R. E. Ladner, and J. Mankoff, "Toward language justice: Exploring multilingual captioning for accessibility," in *Proc. CHI Conf. Human Factors Comput. Syst.*, 2025, pp. 1–18.

[6] E. J. McDonnell, T. Eagle, P. Sinlapanuntakul, S. H. Moon, K. E. Ringland, J. E. Froehlich, and L. Findlater, "Caption it in an accessible way that is also enjoyable," in *Proc. CHI Conf. Human Factors Comput. Syst.*, 2024, pp. 1–16.

[7] J. Ballard. (2023). *Most American Adults Under 30 Prefer Watching TV With Subtitles–Even When They Know the Language*. [Online]. Available: https://today.yougov.com/entertainment/articles/45987-american-adults-under-30-watching-tv-subtitles

[8] S. Derbring, P. Ljunglöf, and M. Olsson, "SubTTS: Light-weight automatic reading of subtitles," in *Proc. Nordic Conf. Comput. Linguistics*, 2009, pp. 272–274.

[9] S. Garg, "Automatic text summarization of video lectures using subtitles," in *Recent Developments in Intelligent Computing, Communication and Devices*. Singapore: Springer, 2017, pp. 45–52.

[10] T. V. Daele, A. Iyer, Y. Zhang, J. C. Derry, M. Huh, and A. Pavel, "Making short-form videos accessible with hierarchical video summaries," in *Proc. CHI Conf. Human Factors Comput. Syst.*, 2024, pp. 1–17.

[11] S. Malakul and I. Park, "The effects of using an auto-subtitle system in educational videos to facilitate learning for secondary school students: Learning comprehension, cognitive load, and satisfaction," *Smart Learn. Environments*, vol. 10, no. 1, pp. 1–17, Jan. 2023.

[12] L. Jiang, M. Phutane, and S. Azenkot, "Beyond audio description: Exploring 360° video accessibility with blind and low vision users through collaborative creation," in *Proc. 25th Int. ACM SIGACCESS Conf. Comput. Accessibility*, Oct. 2023, pp. 1–17.

[13] E. Peters, E. Heynen, and E. Puimège, "Learning vocabulary through audiovisual input: The differential effect of L1 subtitles and captions," *System*, vol. 63, pp. 134–148, Dec. 2016.

[14] C. Y. Lokkondra, D. Ramegowda, G. M. Thimmaiah, A. P. B. Vijaya, and M. H. Shivananjappa, "ETDR: An exploratory view of text detection and recognition in images and videos," *Revue d'Intell. Artificielle*, vol. 35, no. 5, pp. 383–393, Oct. 2021.

[15] L. Yang, D. Ergu, Y. Cai, F. Liu, and B. Ma, "A review of natural scene text detection methods," *Proc. Comput. Sci.*, vol. 199, pp. 1458–1465, Dec. 2022.

[16] W. Wu, Y. Cai, C. Shen, D. Zhang, Y. Fu, H. Zhou, and P. Luo, "End-to-end video text spotting with transformer," *Int. J. Comput. Vis.*, vol. 132, no. 9, pp. 4019–4035, Sep. 2024.

[17] I. Degtyarenko, N. Tkach, O. Radyvonenko, I. Deriuga, K. Seliuk, O. Ivanov, V. Sielikhov, S. Y. Lee, Y.-H. Choi, and C.-H. Hahm, "SDRV: Real-time on-device subtitles detection, recognition and voicing," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. Workshops (ICASSPW)*, Jun. 2023, pp. 1–5.

[18] R. Lienhart and A. Wernicke, "Localizing and segmenting text in images and videos," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 4, pp. 256–268, Apr. 2002.

[19] J. Peng and Q. Xiaolin, "Keyframe-based video summarization using visual attention clue," *IEEE MultimediaMag.*, pp. 64–73, Jan. 2009.

[20] S. Karaoglu, J. V. Gemert, and T. Gevers, "Object reading: Text recognition for object recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 456–465.

[21] B. Yu, W. Ma, K. Nahrstedt, and H. Zhang, "Video summarization based on user log enhanced link analysis," in *Proc. 11th ACM Int. Conf. Multimedia*, 2003, pp. 382–391.

[22] T. Lu, S. Palaiahnakote, C. L. Tan, and W. Liu, *Video Text Detection*, vol. 9. Cham, Switzerland: Springer, 2014.

[23] P. Shivakumara, W. Huang, and C. L. Tan, "An efficient edge based technique for text detection in video frames," in *Proc. 8th IAPR Int. Workshop Document Anal. Syst.*, 2008, pp. 307–314.

[24] V. Abolghasemi and A. Ahmadyfard, "An edge-based color-aided method for license plate detection," *Image Vis. Comput.*, vol. 27, no. 8, pp. 1134–1142, Jul. 2009.

[25] L. Wu, P. Shivakumara, T. Lu, and C. L. Tan, "Text detection using Delaunay triangulation in video sequence," in *Proc. 11th IAPR Int. Workshop Document Anal. Syst.*, Apr. 2014, pp. 41–45.

[26] P. Shivakumara, T. Q. Phan, and C. L. Tan, "A Laplacian approach to multi-oriented text detection in video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 2, pp. 412–419, Feb. 2011.

[27] X.-C. Yin, Z.-Y. Zuo, S. Tian, and C.-L. Liu, "Text detection, tracking and recognition in video: A comprehensive survey," *IEEE Trans. Image Process.*, vol. 25, no. 6, pp. 2752–2773, Jun. 2016.

[28] S. Tian, X.-C. Yin, Y. Su, and H.-W. Hao, "A unified framework for tracking based text detection and recognition from web videos," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 542–554, Mar. 2018.

[29] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[30] A. Shahab, F. Shafait, and A. Dengel, "ICDAR 2011 robust reading competition challenge 2: Reading text in scene images," in *Proc. Int. Conf. Document Anal. Recognit.*, Sep. 2011, pp. 1491–1496.

[31] J. Zhang and R. Kasturi, "Extraction of text objects in video documents: Recent progress," in *Proc. 8th IAPR Int. Workshop Document Anal. Syst.*, 2008, pp. 5–17.

[32] H. Yan and X. Xu, "End-to-end video subtitle recognition via a deep residual neural network," *Pattern Recognit. Lett.*, vol. 131, pp. 368–375, Mar. 2020.

[33] D. Gilly and K. Raimond, "A survey on license plate recognition systems," *Int. J. Comput. Appl.*, vol. 61, no. 6, pp. 34–40, Jan. 2013.

[34] O. Viatchaninov, V. Dziubliuk, O. Radyvonenko, Y. Yakishyn, and M. Zlotnyk, "CalliScan: On-device privacy-preserving image-based handwritten text recognition with visual hints," in *Proc. UIST*, 2019, pp. 72–74.

[35] T. Q. Phan, P. Shivakumara, B. Su, and C. L. Tan, "A gradient vector flow-based method for video character segmentation," in *Proc. Int. Conf. Document Anal. Recognit.*, Sep. 2011, pp. 1024–1028.

[36] S. Uchida, Y. Shigeyoshi, Y. Kunishige, and F. Yaokai, "A keypoint-based approach toward scenery character detection," in *Proc. Int. Conf. Document Anal. Recognit.*, Sep. 2011, pp. 819–823.

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

[38] L. Neumann and J. Matas, "A real-time scene text to speech system," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 619–622.

[39] X. Zhang, Y. Su, S. Tripathi, and Z. Tu, "Text spotting transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 9509–9518.

[40] J. Lyu, J. Wei, G. Zeng, Z. Li, E. Xie, W. Wang, C. Ma, and Y. Zhou, "TextBlockV2: Towards precise-detection-free scene text spotting with pre-trained language model," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 21, no. 6, pp. 1–21, Jun. 2025.

[41] D. Bautista and R. Atienza, "Scene text recognition with permuted autoregressive sequence models," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 178–196.

[42] H. Wang, H. Zhou, Y. Zhang, J. Ma, and H. Ling, "Word length-aware text spotting: Enhancing dense text detection and recognition for camera-captured document image," *IEEE Trans. Instrum. Meas.*, vol. 74, pp. 1–15, 2025.

[43] A. Grygoriev, I. Degtyarenko, I. Deriuga, S. Polotskyi, V. Melnyk, D. Zakharchuk, and O. Radyvonenko, "HCRNN: A novel architecture for fast online handwritten stroke classification," in *Proc. ICDAR*, 2021, pp. 193–208.

[44] S. Long, X. He, and C. Yao, "Scene text detection and recognition: The deep learning era," *Int. J. Comput. Vis.*, vol. 129, no. 1, pp. 161–184, Jan. 2021.

[45] C. Li, W. Liu, R. Guo, X. Yin, K. Jiang, Y. Du, Y. Du, L. Zhu, B. Lai, X. Hu, D. Yu, and Y. Ma, "PP-OCRv3: More attempts for the improvement of ultra lightweight OCR system," 2022, *arXiv:2206.03001*.

[46] R. Mittal and A. Garg, "Text extraction using OCR: A systematic review," in *Proc. ICIRCA*, 2020, pp. 357–362.

[47] Q. Ye and D. Doermann, "Text detection and recognition in imagery: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 7, pp. 1480–1500, Jul. 2015.

[48] Y. Zhu, C. Yao, and X. Bai, "Scene text detection and recognition: Recent advances and future trends," *Frontiers Comput. Sci.*, vol. 10, no. 1, pp. 19–36, Feb. 2016.

[49] Y. Xu, S. Shan, Z. Qiu, Z. Jia, Z. Shen, Y. Wang, M. Shi, and E. I.-C. Chang, "End-to-end subtitle detection and recognition for videos in East Asian languages via CNN ensemble," *Signal Process., Image Commun.*, vol. 60, pp. 131–143, Feb. 2018.

[50] (2025). *Samsung Releases Live Translate and AI Audio Features At CES 2025*. [Online]. Available: https://autogpt.net/samsung-releases-live-translate-and-ai-audio-features-at-ces-2025/

[51] I. Degtyarenko, N. Tkach, K. Seliuk, O. Ivanov, and V. Sielikhov, "Electronic device and audio track obtaining method therefor," UA Patent U.S. 20 250 166 608 A1, May 22, 2025.

[52] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot MultiBox detector," in *Proc. ECCV*, 2016, pp. 21–37.

[53] W. Wang, E. Xie, X. Li, W. Hou, T. Lu, G. Yu, and S. Shao, "Shape robust text detection with progressive scale expansion network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9328–9337.

[54] M. Liao, Z. Zou, Z. Wan, C. Yao, and X. Bai, "Real-time scene text detection with differentiable binarization and adaptive scale fusion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 1, pp. 919–931, Jan. 2023.

[55] A. Howard, A. Zhmoginov, L.-C. Chen, M. Sandler, and M. Zhu, "Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation," in *Proc. CVPR*, 2018, pp. 4510–4520.

[56] M. Liao, B. Shi, and X. Bai, "TextBoxes++: A single-shot oriented scene text detector," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 3676–3690, Aug. 2018.

[57] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2704–2713.

[58] M. Li, T. Lv, J. Chen, L. Cui, Y. Lu, D. Florencio, C. Zhang, Z. Li, and F. Wei, "TrOCR: Transformer-based optical character recognition with pre-trained models," in *Proc. AAAI Conf. Artif. Intell.*, Jun. 2023, pp. 13094–13102.

[59] A. Graves, S. Fernandez, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proc. ICML*, 2006, pp. 369–376.

[60] B. Suvarnam and V. S. Ch, "Combination of CNN-GRU model to recognize characters of a license plate number without segmentation," in *Proc. 5th Int. Conf. Adv. Comput. Commun. Syst. (ICACCS)*, Mar. 2019, pp. 317–322.

[61] Z. Chen, M. Jain, Y. Wang, M. L. Seltzer, and C. Fuegen, "End-to-end contextual speech recognition using class language models and a token passing decoder," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 6186–6190.

[62] (2025). *Enjoy Your Show With Samsung Vision AI*. [Online]. Available: https://www.samsung.com/levant/tvs/8k-tv/highlights/

[63] Y.-C. Wu and C. T. Huang, "Efficient dynamic fixed-point quantization of CNN inference accelerators for edge devices," in *Proc. Int. Symp. VLSI Design, Autom. Test (VLSI-DAT)*, Apr. 2019, pp. 1–4.

[64] R. Karpinski, D. Lohani, and A. Belaïd, "Metrics for complete evaluation of OCR performance," in *Proc. CVPR*, 2018, pp. 104–110.

[65] W. Wu, Y. Cai, D. Zhang, S. Wang, Z. Li, J. Li, Y. Tang, and H. Zhou, "A bilingual, OpenWorld video text dataset and end-to-end video text spotter with transformer," 2021, *arXiv:2112.04888*.

[66] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The CLEAR MOT metrics," *EURASIP J. Image Video Process.*, vol. 2008, pp. 1–10, Jan. 2008.

[67] P. Henderson and V. Ferrari, "End-to-end training of object class detectors for mean average precision," in *Proc. Asian Conf. Comput. Vis.*, 2016, pp. 198–213.

[68] C. Cui, T. Sun, M. Lin, T. Gao, Y. Zhang, J. Liu, X. Wang, Z. Zhang, C. Zhou, H. Liu, Y. Zhang, W. Lv, K. Huang, Y. Zhang, J. Zhang, J. Zhang, Y. Liu, D. Yu, and Y. Ma, "PaddleOCR 3.0 technical report," 2025, *arXiv:2507.05595*.

[69] M. A. M. Salehudin, S. N. Basah, H. Yazid, K. S. Basaruddin, M. J. A. Safar, M. H. M. Som, and K. A. Sidek, "Analysis of optical character recognition using EasyOCR under image degradation," *J. Phys., Conf. Ser.*, vol. 2641, no. 1, Nov. 2023, Art. no. 012001.
[70] *ITU T-REC-G.114 Series G: Transmission Systems and Media, Digital Systems and Networks*, Standard ITU T-REC-G.114, May 2003.

ILLYA DEGTYARENKO (Member, IEEE) received the Ph.D. degree in automation and control systems from Donetsk National Technical University, Ukraine, in 2001. From 2001 to 2014, he was an Associate Professor with the Automation and Telecommunication Department, Donetsk National Technical University. Since 2014, he has been with Samsung Research and Development Institute Ukraine, as a Signal Processing and AI Expert. His main fields of interests include non-stationary process analysis, hierarchical deep neural networks, and edge computing.

OLGA RADYVONENKO (Member, IEEE) received the Ph.D. degree in artificial intelligence from Kharkiv National University of Radio Electronics, Kharkiv, Ukraine, in 2008. She is currently the Head of Research Laboratory, Samsung Research and Development Institute Ukraine. Her research interests include deep learning, computer vision, document intelligence, and intelligent user interfaces.

NAZARII TKACH (Member, IEEE) received the bachelor's degree in software engineering and the M.Sc. degree in computer science from National University of Kyiv-Mohyla Academy, Kyiv, Ukraine, in 2018 and 2020, respectively, where he is currently pursuing the Ph.D. degree in computer science. In 2018, he joined Samsung Research and Development Institute Ukraine. His research interests include scene text detection and recognition using neural networks and their applicability for implementation on edge devices.

VALERII SIELIKHOV received the M.Sc. degree in computer science from the National Technical University of Ukraine "Kyiv Polytechnic Institute", Institute for Applied System Analysis, in 2024. Since 2021, he has been with Samsung Research and Development Institute Ukraine, focusing on detection and recognition, using deep neural networks and their applicability for implementation on edge devices.

KOSTIANTYN SELIUK received the M.Sc. degree in applied mathematics from the Faculty of Physics and Technology, National Technical University of Ukraine "Kyiv Polytechnic Institute", in 2013. Since 2013, he has been with Samsung Research and Development Institute Ukraine, initially focusing on mobile devices. Starting in 2015, his work expanded to include deep neural networks, machine learning, projects related to smart TVs, and deployment of large language models on resource-constrained devices.

OLEKSANDR IVANOV received the M.Sc. degree in computer engineering from the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", in 2019. In 2017, he was with Samsung Research and Development Institute, starting with robotics, embedded programming, and wearable devices. He has since expanded his expertise to include machine learning, deep neural networks, and edge computing.