IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

MorseNet: A Unified Neural Network for Morse Detection and Recognition in Spectrogram

WEIHAO LI[1,2], KEREN WANG[2], AND LING YOU[2]

[1]PLA Strategic Support Force Information Engineering University, Zhengzhou 450001, China
[2]National Key Laboratory of Science and Technology on Blind Signal Processing, Chengdu 610041, China

Corresponding author: Weihao Li (liweihao315@gmail.com).

ABSTRACT Short-wave radio is an indispensable long-distance means of communication, among which Morse signals, which rely on simplicity and efficiency, plays an import role in military and civilian applications. Automatic Morse detection and recognition have been researched for several years, but some thorny problems in actual communication always restrict the performance of methods. In this paper, by introducing deep learning technology, we propose a network named MorseNet that can simultaneously locate and decode Morse signals in the spectrogram. MorseNet uses shared convolutions to extract shared features for both the detection and recognition branches. The detection branch regresses bounding boxes based on signal centerlines, and the recognition branch decodes Morse fragments cropped from feature maps by a convolutional recurrent neural network (CRNN). The losses of two branches are combined to implement the end-to-end training. Experimental results on four "simulated Morse + real background" datasets demonstrate that the proposed method achieves state-of-the-art performance in both detection and recognition, and it effectively improves four problems that have long been troublesome in accomplishing the tasks. Furthermore, the joint training strategy and architecture give MorseNet advantages over its two-stage deployment in terms of accuracy, speed, and model size.

INDEX TERMS Deep learning, Morse signal detection, Morse signal recognition, end-to-end network, spectrogram.

I. INTRODUCTION

A Morse signal is a type of continuous wave (CW) with a steady frequency and intermittent time. It consists of 5 types of codes: dot, dash, intra-code interval, inter-code interval and code group interval, the permutation order of which can represent different characters. Due to the simple coding scheme, narrow frequency band, and strong anti-jamming capability, Morse signals are widely applied in aviation, maritime and military communications [1]. At present, the copying of Morse signals, especially those sent manually, are mainly implemented by humans, which imposes pressure on the operators and has an unstable accuracy. Therefore, automatic Morse detection and recognition have been researched for many years, but some tricky problems in actual communications make it quite difficult. In recent years, in view of the excellent performance of deep learning (DL) technology on images, speech, and natural language processing, it has also been introduced to different aspects of communications and networks [2], [3] and has shown great potential.

Due to the specialty of having steady frequency and intermittent time in Morse, time-frequency analysis methods led by short-time Fourier transform (STFT) [4] dominate the preprocessing. The spectrogram obtained by STFT can clearly visualize the time and frequency information of the signals. Morse detection is the premise of recognition, and the aim is to detect the presence and time-frequency location of Morse in received wireless data. In the spectrogram, traditional methods usually first extract the fragments that contain signals by energy detection, and then, they design classifiers, including machine learning or deep learning models, to classify signal type [5]–[7]. Energy detection plays well in a spectrogram with scattered signals, but it could make mistakes when signals are densely distributed. Recently, some researchers have exploited the single shot multibox detector (SSD) network to detect multi-type signals in spectrograms [8], [9]. SSD is a common DL-based object

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3020860, IEEE Access

detection method that is capable of locating signals by a bounding box (BBox) and identifying the type. However, it uses the center point of the object to predict the BBox size, the receptive field of which is limited, especially for horizontally long signals. Thus, it usually fails to predict the complete BBox for the signals, which is inacceptable for the subsequent recognition task. In addition, it raises too many candidate anchors to regress, costing much time, and the anchor size is difficult to determine because of the dramatic change in the length of signals. Inheriting from the SSD, we targeted the characteristics of the signals and proposed an improved detector in our earlier work [10]. The detector first finds the centerline of the signal in a heat map, whose receptive field could cover the whole signal, and then, it predicts BBox size directly at the centerline points, thus abandoning the anchors, which proposes a more intact BBox and simplifies the model to greatly speed up. In view of the excellent performance of DL in computer vision, it has the potential to be introduced to spectrogram-based signal detection.

For the recognition of detected Morse, a common method is to identify the code types (dot, dash, and three intervals) in time sequence, and then, to look up the code-to-character table to obtain the final text. The code types are classified by the code lengths. In the spectrogram, traditional methods [5], [6], [11] implement image processing, including contrast enhancement, binarization and morphological denoising, to highlight the Morse regions in a spectrogram, where the lengths of the bright strips and their intervals are recorded as a feature set. Then, a clustering method, such as k-means or c-means, is introduced to classify the code types. Those methods divide the recognition task into multiple stages, which increases the complexity, and they depend heavily on the image processing effect. In [12], we utilized a convolutional recurrent neural network (CRNN) to accomplish end-to-end image-to-character level recognition. The CRNN makes use of a convolutional neural network (CNN) to extract an image's deep features and a recurrent neural network (RNN) to capture the context information, which has greatly improved accuracy, simplified the processing, and has no table look-up.

It can be seen that DL-based methods have achieved state-of-the-art performance in both detection and recognition of Morse [10], [12]. However, the two neural networks are trained separately, which means that the character level information in the recognition that could help improve the detection effect is ignored by the detection model. In addition, recognition is conducted on Morse regions cropped from the original image, one by one, which costs a substantial amount of time, especially for spectrograms that contain many Morse signals. In addition, duplicate CNN-based feature extractors in the detection and recognition network introduce operational redundancy. Inspired by the FOTS method [13], which is a typical text spotting network that combines two text detection and recognition networks to obtain better and faster performance, we propose to combine the Morse detection and recognition networks into a unified network named MorseNet and implement end-to-end training. Based on the detection model in [10], we add the CRNN model [12] after the feature extraction CNNs as a recognition branch. Thus, the feature extraction CNNs become a shared convolution, which supplies shared features to both the detection and recognition branches. The detection branch is a multi-channel convolutional network that locates signals at its centerline and regresses to BBox. The recognition branch consists of CNNs, a bidirectional long short-term memory (BLSTM) encoder, and a connectionist temporal classification (CTC) decoder. Through joint supervision, the visual and context information can be shared between two tasks, which are thus expected to improve the performances of each other. In addition, shared CNNs could save the duplicated time cost. Experimental results show that our MorseNet outperforms traditional methods and its two-stage version method in both accuracy and speed.

To summarize, the contributions of this paper are as follows:

- We propose a unified neural network named MorseNet for the detection and recognition of Morse signals in spectrograms. To the best of our knowledge, this study proposes the first DL-based architecture for simultaneously detecting Morse signals and recognizing Morse codes.
- We introduce a shared convolution, to extract shared features for the detection and recognition branches, and combine two branch losses to implement end-to-end training, which improves the accuracy and saves time.
- To make experiments persuasive, we simulate Morse signals and add them into real-world background in the time domain. Experimental results show that MorseNet obtains state-of-the-art performance in both detection and recognition on four datasets.

For the remainder of this paper, Section II reviews the related work on Morse detection and recognition, while Section III introduces data collection and some common problems in the task. Section IV describes the details of our methodology, and Section V evaluates the performance of MorseNet in comparison with baselines. The conclusions are drawn in Section VI.

II. RELATED WORK

Automatic Morse detection and recognition are two problems that have a long history. After the creation of Morse in 1837, many researchers have studied it. In this section, we give a brief introduction to related work on those two tasks, which are summarized in Table 1.

A. MORSE DETECTION

Existing Morse detection methods can be categorized into traditional methods and DL-based methods.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3020860, IEEE Access

Traditional methods focus mainly on the time domain, the frequency domain or both. Envelope detection is the earliest method [14], which has a fast speed but weak noise resistance and has poor practicability in the currently complex electromagnetic environment. A phase-locked loop [15] can track the signal frequency, under the premise of accurate signal frequency estimation, and it is sensitive to interference. Filtering methods, including Kalman filtering [16] and adaptive filtering [17], are also introduced. By elaborately designing a filter, the signal can be effectively denoised, but it also requires a frequency estimate in advance and is powerless to address an unstable frequency. Some signal transformations, such as Fourier transform [18], complex variance spectrum [19] and wavelet transform [20], can find the Morse frequency in the spectrum, but they only obtain the frequency distribution of the signals, without the time information, and they cannot effectively distinguish Morse from other signals. The methods above mainly process in the time domain, under the assumption that processed data contains Morse of only one channel. Time-frequency analysis methods, which take advantage of the typical characteristics of Morse in both the time domain and the frequency domain, have become the mainstream. Yue *et al.* [21] utilized a discrete Gabor transform to obtain time-frequency information of Morse, but they lacked a related algorithm to distinguish Morse from interference. As the main idea in most of the literature, Wei *et al.* [5], Sun *et al.* [6], and Yuan *et al.* [7] employed energy detection on the spectrogram and introduced a classifier such as a machine learning or DL model to classify the signal type. Among them, the CNN-based model in [7] obtained the best classification result. Nevertheless, since energy detection is quite sensitive to the noise, especially in short-wave communication, those methods suffer from a low detection accuracy for the signal types of interest.

For the DL-based methods, in recent years, Zha *et al.* [8] and Singh. A [9] utilized the DL-based object detector SSD, and they converted the task of multi-type signal detection in a spectrogram to object detection in an image, which is a sparkly idea capable of locating signals of different types. However, SSD and other object detectors usually raise many candidate anchors in advance, the size of which is difficult to determine because of the dramatic change in the length of signals, and their regression is time-consuming. Moreover, their center point-based detection is not suitable for horizontally long signals, which leads to incomplete BBox proposals. To make up for the above shortcomings, in [10], we proposed a centerline-based neural network that models the signal based on its centerline and corresponding properties, other than the candidate anchors, and we achieved state-of-the-art performance for multi-type signal detection in spectrograms.

B. MORSE RECOGNITION

Traditional recognition steps are to first obtain the code lengths, then classify the code types, and finally, look up the code-to-character table.

To obtain the code lengths, researchers in [23]–[25] directly tracked signal waveforms in the time domain. To get rid of interference and highlight the electric levels, they usually conducted filtering and binarization on the original data as preprocessing. Those methods worked under the condition that there is only one channel of Morse in data, and they depended greatly on the preprocessing effect. Wei *et al.* [5], Sun *et al.* [6], and Wang *et al.* [11] adopted the spectrograms by STFT and combined them with image processing tools to highlight the Morse regions, where the lengths of the bright strips and their intervals were counted. However, their performance was also limited by the image processing effect.

In classifying the code types, it depends on the time lengths of the codes. Theoretically, the length ratio of dot, dash, intra-code interval, inter-code interval, and code group interval is 1:3:1:3:5. Xiao *et al.* [20] modified the Gunther algorithm, which was an earlier but relatively powerless decoding algorithm. Some researchers constructed traditional machine learning models such as support vector machine (SVM) [16], [25], k-means cluster [6], [11], [22], [24], c-means cluster [6], and so on, to classify code. Traditional machine learning models use only code lengths as features, without context information, which is not robust to sharp code length deviations. The above methods accomplish code level recognition, and additional post-processing is inevitable, including code-to-character table look-up and error correction. Researchers in [6], [26] designed algorithms to speed up the table look-up, and those in [11], [24] made error correction rules to further improve the recognition results. To summarize, the bottleneck of traditional methods lies mainly in obtaining the code lengths, which places a large amount of pressure on preprocessing in the recognition task. In addition, the above methods are all multi-stage, which could cause error accumulation, and the table look-up and error correction are time-consuming.

Recently, Wang *et al.* [27] utilized the hidden Markov model (HMM) + deep neural networks (DNN) that was a classical speech recognition algorithm to accomplish character level recognition, but its performance was not sufficiently high. Inheriting from it, in [12], we used a deep neural network CRNN to recognize Morse in a spectrogram at the character level and obtained state-of-the-art performance, which was also the first DL-based attempt on this task. In recent years, DL technology shows a strong ability in image perception and sequence modeling, and thus, it is very suitable for spectrogram-based Morse recognition.

As can be seen, the applications of DL in Morse detection and recognition are relatively rare, let alone a unified network that implements the end-to-end task. Compared to two-stage processing of detection + recognition, end-to-end processing could let two tasks share the learned features from

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3020860, IEEE Access

each other, and save time by merging redundant structures. Thus, in this paper, we decided to construct a neural network with elegant and complementary architecture to accomplish this task.

III. PRELIMINARY

Considering the real-time ability of our system, the input is a narrowband spectrogram that contains multi-channel Morse signals. In this section, we introduce our data collection method and some long-standing problems faced by Morse detection and recognition tasks.

A. DATA COLLECTION

Our dataset mainly consists of synthetic Morse signals and real-world wireless signals that are mainly taken as background noise. To simulate various degrees of skill, we designed formulations for tuning the code speed deviation, frequency drift, frequency jitter and other common distortions for generating Morse signals, the same as in [12]. Then, real-world wireless signals are added with various signal-to-noise ratios (SNR), to constitute the final synthetic datasets. The background signals are collected by a short-wave radio station WiNRADiO G39DDC [28], which received wideband data that contains various types of signals. We implement digital down-conversion (DDC) to obtain narrowband backgrounds and add multi-channel Morse signals in the time domain. Background data are collected at different times of the year, and conversed from different frequency bands.

After the combination of simulated Morse and real background, we transform data to a narrowband spectrogram by STFT, the calculation of which is as follows:

$$S_n(e^{j\omega}) = \sum_{m=-\infty}^{\infty} s(m)w(n-m)e^{-j\omega m}, \quad (1)$$

$$P_n(\omega) = |S_n(e^{j\omega})|^2, \quad (2)$$

where $s(m)$ denotes the sampled signal, $w(m)$ denotes the Hanning window function, and $P_n(\omega)$ is the time-frequency energy matrix. The resolution of the spectrogram is determined by the step time Δl of the Hanning window and the FFT point n_{fft}. Although decreasing Δl or increasing n_{fft} could make the spectrogram display more detailed information in the time or frequency domain, it enlarges the image size, which reduces the real-time performance. Based on our engineering experience, we set $\Delta l = 0.02s$ and $n_{fft} = 1024$ for the data with a 15 s duration and a 9000 Hz sampling frequency. Fig. 1 is an instance of our input spectrogram.

B. MAIN PROBLEMS IN DETECTION AND RECOGNITION

Automatic detection and recognition of Morse have been researched for many years, but with few large breakthroughs. This circumstance is mainly blamed on the fact that several thorny problems have not been adequately addressed.

Fading and frequency drift in the shortwave channel: A shortwave channel is a typical random-parametric model, transmitting signals by ionospheric reflection, which has a multipath effect. A change in the ionosphere or weather destabilizes the channel, which is accompanied by energy fluctuations and frequency drift of the received signal. When burst interference or fast fading occurs, SNR declines sharply, which requires strong robustness of the algorithm at a low SNR.

Adjacent channel interference: Adjacent channel interference refers to when a radio station receives more than one channel of Morse or other signals at its working frequency. In this case, a detection algorithm must distinguish the signals of not only different channels but also different types, which demands high frequency resolution and strong classification ability.

Code speed deviation and frequency jitter: A mechanical transmitter sends Morse code with a standard time ratio of dot, dash, and intervals. However, Morse code sent manually usually has a code speed deviation. In addition, many telegraph operators use the telegraph key to send Morse, which could cause frequency jitter at the code start or end. The code speed deviation and frequency jitter place a large amount of pressure on the recognition algorithm.

IV. METHODOLOGY

MorseNet is an end-to-end trainable neural network that detects and recognizes all Morse signals in a spectrogram. It consists of four main modules: shared convolution, detection branch, region extraction, and recognition branch.

A. OVERALL ARCHITECTURE

Fig. 2 illustrates the overall architecture of MorseNet. Shared convolution is used to extract shared features for subsequent detection and recognition branches. The backbone of the shared convolution is the same as in [10], which is a ResNet18 network [29] combined with three up-convolutions. Fig. 3 shows the general structure of the shared convolution. The input first pass through a series forward convolutions with a size decrease and a channel increase, and then, three up-convolutions are implemented to enlarge the feature map. The level of the extracted features increases with the number of convolutions, and we connect low-level and high-level feature maps of the same size. From this, the features of different levels can be effectively combined to take account of both the detailed and overall information. The resolution of the final feature map is 1/4 of the original spectrogram. The detection branch is a multi-channel convolutional network that utilizes shared features to locate the centerlines of the Morse and regress to the BBoxes. Then, the region extraction module crops the Morse regions from the feature map and converts them to a fixed height. Finally, the text recognition branch translates the codes to text with CNNs, a BLSTM encoder, and a CTC decoder.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3020860, IEEE Access

B. DETECTION BRANCH

Since Morse signal has a fixed frequency and very narrow bandwidth, the centerline-based method in [10] is very suitable for its detection. Inspired by this, we construct a fully convolution network as the detection branch whose schematic diagram is plotted in Fig. 4. Using the shared features, it predicts three attributes of the Morse region: centerline, local offset, and border offsets. The centerline refers to the horizontal centerline of the Morse region, whose heat map has one channel and represents the pixel-wise probability of belonging to the centerline. Local offset is predicted to offset the positional deviation of the centerline during down-sampling and up-sampling of the shared convolutions, whose map has one channel and valid values within the centerline. Border offsets represent offsets between the centerline and up/down border lines of the Morse region, whose map has two channels and valid values within the centerline.

The loss of the detection branch is composed of three parts, which correspond to the above three attributes. For the centerline loss, during the ground truth map production, we apply smooth probability to the adjacent points of centerlines by a Gaussian kernel. The training objective is a pixel-wise focal loss [30]:

$$L_{\text{cl}} = \frac{-1}{|N|}\sum_{N}\begin{cases}(1-P_p)^{\alpha}\log(P_p), & if\ \hat{P}_p = 1\\ (1-\hat{P}_p)^{\beta}(P_p)^{\alpha} & \\ \log(1-P_p), & otherwise\end{cases}, \tag{3}$$

where p is a point in the map, $\hat{P}_p$ is the ground truth label at p, P_p is the corresponding prediction, N is all points set in the heat map, α and β are the hyper-parameters of the focal loss. In (3), $(1-P_p)^{\alpha}$ and $(P_p)^{\alpha}$ reduce the weights of the easy-to-classify samples and increase those of the difficult-to-classify samples. $(1-\hat{P}_p)^{\beta}$ reduces the weights of $0<\hat{P}_p<1$ (here referring to the adjacent points of centerlines), especially that close to 1, to reduce their impact on training. We empirically set $\alpha=2$ and $\beta=4$ in our experiments.

For the local offset and border offsets losses, we directly calculate the average difference between the ground truth and the predicted value at the centerline points:

$$L_{\text{lo}} = \frac{1}{|\Omega|}\sum_{\Omega}|O_p - (\frac{Y_{\tilde{p}}}{R} - Y_p)|, \tag{4}$$

$$L_{\text{bo}} = \frac{1}{|\Omega|}\sum_{\Omega}(|U_p - \hat{U}_p| + |D_p - \hat{D}_p|), \tag{5}$$

where $\tilde{p}$ is the mapping point of p in the original spectrogram, Ω is centerline point set in the heat map, R is the shrunken scale of the feature map (here 4), and $|\cdot|$ denotes the number of elements. Y is the vertical coordinate of point, and O_p is the predicted local offset. U_p, $\hat{U}_p$ are the predicted/ground truth up border offset, and D_p, $\hat{D}_p$ are the predicted/ground truth down border offset.

The detection loss is the weighted sum of three attribute losses:

$$L_{\text{detect}} = \lambda_1 L_{\text{cl}} + \lambda_2 L_{\text{lo}} + \lambda_3 L_{\text{bo}}, \tag{6}$$

where λ_1, λ_2, λ_3 are empirically set to 1.0, 0.5, 0.5 in our experiments. We assume that $x_{\min}$, $x_{\max}$ are the starting and ending abscissa of a centerline, whose ordinate is y. Thus, the lower left and upper right coordinates of a predicted BBox can be calculated as:

$$R\times(x_{\min}, y+O-D, x_{\max}, y+O+U). \tag{7}$$

C. REGION EXTRACTION

Region extraction is aimed at extracting Morse regions in a shared feature map, by using the output BBox of the detection branch. To adapt the convolution processing of the recognition branch, we shrink the region fragments to a fixed height 8 with an unchanged aspect ratio. Thus, the length of the regions is kept variable, which avoids the misalignment between the features and the original image and preserves the semantic information as much as possible. In practice, we pad each of the region fragments to the longest length of a batch and ignore the padding parts during the recognition.

When training MorseNet, the detection branch may provide nonstandard region proposals, especially at the beginning of the training, which could cause wrong learning of the recognition branch. Thus, we feed the ground truth Morse regions to the recognition branch during training. When testing, the confidence threshold and non-maximum suppression (NMS) are introduced to filter the region proposals. Selected Morse regions are then fed into the recognition branch for character level translation.

D. RECOGNITION BRANCH

The recognition branch utilizes the shared features in each Morse region to predict the text labels. It consists of sequential CNNs, a BLSTM encoder [31], and a CTC decoder [32]. The specific structure of the recognition branch is shown in Table 2.

Sequential CNNs are first built to further extract the image semantic information. Since the size of the feature map has shrunk twice in shared convolution and region extraction, we pool it only along its height axis to avoid missing text content, especially those characters that have a short code length. Through CNNs, the heights of the input feature fragments are compressed to 1 while the channels are increased to 256. In [12], we have performed related experiments to confirm that the convolution processing before the RNN encoder could effectively improve the model's performance.

The CNNs' outputs are permuted to feature sequences in the time axis and are fed into the RNN layer for encoding. Here, we chose LSTM with 256 units to effectively capture the contextual information. Since the front and back frames

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3020860, IEEE Access

in the feature sequence both help in the modeling of the current frame, we use BLSTM, which consists of a forward LSTM and a backward LSTM. Hidden states computed in two directions are summed up and fed into a fully-connected (FC) network. The FC transforms hidden states to a frame-to-character probability matrix. To avoid overfitting, a dropout operation is added before the FC.

The CTC layer is used to transcribe the probability matrix to the final text. The length of text is usually much shorter than that of feature sequence, since one character is usually mapped by multiple frames; as a result, what CTC does is to flexibly merge repetitive predictions of frames. CTC introduces a prediction path $\pi=(\pi_1,\pi_2,...,\pi_L)$ as frame-wise predictions for the feature sequence x, and a 'blank' character to separate adjacent same labels. By merging the same characters between two 'blank' and deleting 'blank', the prediction path is transcribed to the final text. For example, "-aa-p-p-ll-e-" to "apple" ("-" refers to "blank"). The text probability is the sum of the prediction path probabilities that can be transcribed to the text:

$$p(\pi \mid x) = \prod_{l=1}^{L} q_l^{\pi_l} , \tag{8}$$

$$p(y \mid x) = \sum_{\pi \in \Phi(y)} p(\pi \mid x) , \tag{9}$$

where $p(\pi \mid x)$ is the probability of a prediction path, $q_l^{\pi_l}$ is the softmax probability of label π_l at frame l, $\Phi(y)$ refers to all of the CTC prediction paths that can be transcribed to text y, and $p(y \mid x)$ is the final text probability. The recognition loss can be calculated as follows:

$$L_{\text{recog}} = \frac{-1}{N} \sum_{n=1}^{N} \log p(y_n \mid x) , \tag{10}$$

where N is the number of Morse regions in a spectrogram.

The end-to-end loss function of MorseNet is a combination of detection and recognition losses:

$$L = L_{\text{detect}} + \lambda_{\text{recog}} L_{\text{recog}} , \tag{11}$$

where λ_{recog} is a hyper-parameter that trades off the detection branch and recognition branch, which is set to 1 in our experiments.

V. EXPERIMENTS AND DISCUSSION

In this section, we conduct experiments on four datasets. We first give introductions to the baseline methods and implementation details, and then, we show the detection and recognition performances of methods. In particular, the results of MorseNet in the harsh situations mentioned in III-B are visualized, and several sensitivity tests on the SNR, code speed, and hyper-parameters are conducted. Finally, we illustrate our advantages over the two-stage version method in terms of the speed and model size.

A. BASELINES

MorseNet is a fully DL-based neural network, and its detection branch and recognition branch have each obtained state-of-the-art performance [10], [12]. Before our method, Sun *et al.* [6] implemented multi-channel Morse detection and recognition in a spectrogram by energy detection + decision tree for detection and k-means clustering for recognition, the ideas of which were then extended further, in which CNNs were used to replace decision tree, and they obtained the best classification effect [7]; image processing was also introduced before k-means to highlight the codes [5], [11]. We combine the above methods to a relatively advanced traditional method to compare with MorseNet. The DL-based method SSD is also compared in detection performance. In addition, a two-stage system based on MorseNet is built to demonstrate the accuracy and speed advantages of our end-to-end system.

Energy detection + CNNs + image processing + k-means (ECIK) [5]–[7], [11]: ECIK is actually a four-stage system that combines various traditional methods. Energy detection extracts fragments from a spectrogram with strong energy, and then, CNNs are built to classify them to select those that contain Morse. Image processing, including contrast enhancement, binarization, and morphological denoising, is implemented to highlight codes and to denoise. Finally, the code lengths are counted and fed to a k-means model to classify the code types, and the text is translated by the code-to-character table look-up.

SSD [8]: SSD is a representative of DL-based object detectors. The rough idea of SSD is to first raise candidate anchors at each pixel of the extracted feature map and, then, predict the positive probability and size regression for each anchor by several CNNs. The architecture of the SSD used in our experiments is the same as that in [8], where it is exploited to detect multi-type signals in wideband spectrograms. Since SSD targets the detection task, we only compare it in detection performance.

Our Two-Stage: We propose a joint training strategy and architecture to let the network be supervised by both the detection and recognition tasks, with the expectation of improving the accuracy and speed. To verify this approach, a two-stage system is built in which the detection model and recognition model are divided from MorseNet. Two models are trained separately. The Morse fragments cropped from the original spectrogram by the detection model are input to the recognition model.

B. IMPLEMENTATION DETAILS

Experimental dataset: Simulated signals combined with real-world backgrounds are used to evaluate the performance. The backgrounds are narrowband data down-converted from wideband. The same as in [7], we divide the experimental data into four datasets based on the frequency bands that the backgrounds are converted from. The dataset and spectrogram information are described in Table 3.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3020860, IEEE Access

Training setting: We implement the proposed MorseNet model using Tensorflow [33]. An Adam optimizer [34] with a learning rate of 2×10^{-4} is used to optimize the network. We set 0.3 dropout, 0.95 momentum and 1×10^{-5} weight decay to inhibit overfitting, and we exploit data augmentation, including randomly cropping, scaling, and Gaussian noise, to improve the learning effect. All of the models are trained to converge with a batch size of 50, and the experiments are performed on a Tesla P40 GPU.

Metrics: We evaluate the detection and recognition performance during the end-to-end task, where the input of the recognition branch is the Morse regions proposed by the detection branch. The detection metrics are the precision, recall, and F1-score, where the intersect-over-union (IoU) threshold is set to 0.5. What must be emphasized is that for the ECIK model, its detection module can only propose the frequency location of Morse, without the start/end time, which extracts the fragments with the whole time duration of the spectrogram as Morse regions. To fairly evaluate the detection performance of MorseNet and ECIK, we adjust the denominator of the IoU function to the minimum size of the predicted (P) and ground truth (G) BBoxes by (12). Since the annotated boxes in MorseNet and the extracted fragments in ECIK both have a fixed frequency band of 400 Hz, it will not be possible to propose a very large region to obtain a high IoU score. Recognition metrics are the character error rate (CER) and the word error rate (WER). CERs are calculated from the edit distance between the predicted and ground truth text, as in (13). WER refers to the proportion of mistranslated text in all text, as in (14).

$$\mathrm{IoU}=\frac{\mathrm{area}(P)\cap\mathrm{area}(G)}{\min(\mathrm{area}(P),\mathrm{area}(G))}, \tag{12}$$

$$\mathrm{CER}=\frac{\sum_i \mathrm{editdistance}(\mathrm{text}_P^i,\mathrm{text}_G^i)}{\sum_i \mathrm{num\_char}(\mathrm{text}_G^i)}, \tag{13}$$

$$\mathrm{WER}=\frac{\sum_i \mathrm{text}_P^i \neq \mathrm{text}_G^i}{I}. \tag{14}$$

C. DETECTION AND RECOGNITION PERFORMANCE

We compare MorseNet with baseline methods in detection and recognition, and we give the quantitative results in Table 4. As can be seen, MorseNet significantly outperforms the traditional ECIK method, and it also surpasses SSD and Our Two-Stage methods, both in detection and recognition.

For the detection part of ECIK, the classification ability of the CNNs is strong enough, and thus, the performance depends greatly on the energy detection. However, the energy threshold is difficult to determine, since the energy distribution fluctuates rapidly, and it is sensitive to interference. When the energy threshold is too high, some Morse signals could be omitted, or only part of the signal is detected, which results in the relatively low detection scores of ECIK. The MorseNet detection branch utilizes CNNs to classify the objects in the whole spectrogram, other than the selected fragments of the energy detection. CNNs can learn multidimensional features, unlike energy detection, which exploits only energy amplitudes, thus effectively distinguishing objects and improving the detection performance. At the same time, MorseNet regresses a BBox that tightly surrounds the signal, removing the needless background for a better subsequent recognition effect. Compared to SSD, MorseNet is more suitable for the signal characteristics—utilizing the points in the centerline to make predictions, instead of only the center point, which ensures that the receptive fields of the CNNs can cover the whole signal, hence leading to better performance.

For the ECIK recognition part, similar to the detection part, its performance is heavily determined by image processing. Image processing tools could get rid of only interference with weak energy or a scattered distribution, the effect of which is limited in real-world communications. Moreover, the k-means algorithm uses only the code lengths in the entire Morse region to cluster, which could easily produce errors when there are codes with a large length deviation. For the MorseNet recognition branch, the feature sequence extracted by the CNNs could clearly reflect the Morse and interference signals' distribution. Additionally, BLSTM possesses excellent sequence modeling ability, which learns various code length deviation cases during training, hence showing better recognition performance. In addition, the outstanding detection performance of MorseNet lays a good foundation for recognition.

The comparative results of MorseNet and Our Two-Stage show that our unified architecture contributes to better convergence compared with the separate models. The joint training strategy lets the feature extraction module simultaneously be optimized by two tasks, where the character level features learned from the recognition branch help detection branch to distinguish the Morse signal from background, and an enhanced detection module in turn improves the recognition effect. To reflect the universality of the methods, we test them in cross-data mode, which refers to training on the 5 M dataset and testing on the 11 M dataset. The "Cross-data" results in Table 4 show the performance of the method has a slight drop, most likely because some background noise in the test dataset has not been learned during training. However, MorseNet can still perform at a high level.

To further visualize the MorseNet performance in an actual environment, we plot some results in the spectrograms. As shown in Fig. 5, MorseNet greatly improves the detection and recognition effect under four commonly harsh circumstances mentioned in III-B. For the **low SNR** case, Fig. 5(a) shows two spectrograms at SNR -10 dB, where the Morse signals are covered by strong noise, and it is even hard

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3020860, IEEE Access

for the human eye to recognize the codes, while MorseNet completely locates the signals and correctly decodes them. For the **adjacent channel interference** case, benefitting from the strong image recognition ability of the CNN, MorseNet is not affected by the single frequency noise in the top spectrogram and the speech signal in the bottom spectrogram. For the **code speed deviation** case, in the top spectrogram, the interval lengths vary substantially between the codes, and in the bottom spectrogram, the dot lengths of "d", "v", and "w" are close to the dash lengths, which could cause errors for the clustering algorithms. MorseNet achieves accurate recognition, which is mainly due to the context-based modeling of BLSTM. For the **frequency drift and jitter** case, the top spectrogram has a frequency drift instance, and the bottom spectrogram has frequency jitter instances. MorseNet still detects the complete signals, but a bad code length deviation in the top picture causes wrong recognition. Although the frequency is unstable, during shared convolution, the original image is shrunk to a smaller feature map, which means that signals with limited frequency drift or jitter still roughly appear as a horizontal line in the feature map.

D. SENSITIVITY ANALYSIS

In this subsection, we evaluate the influence of the Morse signal properties and the model hyper-parameters. The signal properties include the SNR and the code speed. Specifically, we plot the F1-score curve for detection evaluation and the CER curve for recognition evaluation versus different parameters on the 5M dataset. We train models at basic configurations in V-B and test them with varied object parameters while keeping the others fixed.

SNR: The SNR in Fig. 6 specifically refers to the power ratio between the simulated Morse and the real background. For the detection performance, some extent of decrease in the SNR has few impacts on that of MorseNet, Our Two-Stage, and SSD, while ECIK has obvious frustration at low SNR. Through our inspection, the general outlines of the Morse signals can still be observed at a relatively low SNR, and due to the strong image recognition ability of the CNN, MorseNet, Our Two-Stage and SSD find the location of the Morse. Although the ECIK detection part also has a CNN model, the effect of the preceding energy detection has been greatly weakened under low SNR, which leads to signal missing or distorted proposals. For recognition, all of the three methods' performances tend to decrease as the SNR goes down. The reason can easily be considered to be that Morse codes are drowned out by strong noise and are too vague to recognize.

Code speed: The performance tendency in Fig. 7 is similar to that in Fig. 6. For the detection performance, all of the four methods are not influenced much by the code speed. The reason could be that although the speed changes, the Morse signal can still be relatively easy to distinguish in the spectrogram. For recognition, a code speed rise leads to a performance decrease for the three methods, and we think that time resolution of the spectrogram may do matter to it. Especially at 40 words per minute (wpm), the time length of a dot is approximately 0.03 s, while the time resolution of the experimental spectrogram is 0.02 s, which means that a dot takes up only one to two pixels, or vanishes, and thus, the error-presented signals are naturally misrecognized.

Hyper-parameters: We implement parameter tunings on several model hyper-parameters to determine the specific configuration of MorseNet. The results are plotted in Fig. 8-10: (1) Fig. 8 shows the F1-score under different channels of the first CNN layer in the detection branch (the first CNN layer in Fig. 4); (2) Fig. 9 shows the CER under different channels of the three CNN layers in the recognition branch; (3) Fig. 10 shows the CER under different layers and cell numbers (Ncell) of BLSTM in the recognition branch. Following the principle of ensuring the accuracy and keeping the model as small as possible, we finally chose "Channel: 32" in (1), "Channel: [64,128,256]" in (2), and "Layer: 1, Ncell: 256" in (3).

Although the method performances fluctuate with the SNR or the code speed variation, the DL-based methods MorseNet, Our Two-Stage, and SSD always surpass the ECIK method, thus showing a stronger robustness. In addition, the better performance of MorseNet compared with Our Two-Stage also demonstrates the improvements obtained from our joint training strategy.

E. SPEED AND MODEL SIZE

In Table 5, we evaluate the speed of four methods with and without GPU, and the model size of MorseNet and Our Two-Stage. The speed metric is FPS, which refers to the number of processed images per second. The model size is measured by the sizes of the model parameters. The results illustrate that the non-DL method ECIK (the neural network is only a small part of ECIK) has an advantage in speed, especially in recognition. For MorseNet and Our Two-Stage methods, benefiting from the concise centerline-based detection structure, they detect Morse at a fast speed, compared to SSD. However, their recognition part consumes most of the time, because the BLSTM is a sequential processing model that cannot take advantage of the GPU parallel computing capability. Without the GPU, the detection speeds of all of the methods decrease obviously, since the CNNs in the models lose parallel computing. However, the processing speed of MorseNet can be acceptable, and it still has obvious advantages in terms of speed and model size compared to Our Two-Stage. Since MorseNet uses a shared convolution to extract shared features, and it inputs the fragments cropped from shrunken feature maps instead of original image to the recognition branch, it effectively saves on computation and storage. As a consequence, MorseNet achieves state-of-the-art performance while keeping a real-time capability.

We calculated the average processing speed on all of the testing dataset described in Table 3. The input spectrogram is 749×512 images spanning 15 s in time length and 4.5 kHz

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3020860, IEEE Access

in frequency width, where the time resolution ($15/749 = 0.02$s) and frequency resolution ($4500/512 = 8.79$Hz) are sufficient to present the signal while not making the image too large. The speed results of MorseNet in Table 5 show that it can process 109.5 s ($7.3 \times 15 = 109.5$s) signals per second with a GPU, and 83.55 s ($5.57 \times 15 = 83.55$s) signals per second without a GPU. The experimental results were tested on Tensorflow and the used GPU is Tesla P40.

VI. CONCLUSIONS

In this work, we present a unified neural network named MorseNet for simultaneous Morse detection and recognition in spectrograms. The applications scenario is the narrowband that contains multi-channel Morse signals. MorseNet combines two networks that perform well in signal detection and recognition, and it implements end-to-end training. For evaluation, simulated Morse signals with added real-world backgrounds are collected and divided into four datasets. The experimental results show that our method significantly outperforms previous methods, effectively improves four long-standing problems in the task, and is more robust in different SNRs and code speeds. In addition, compared to the two-stage version method, our unified architecture improves the performance in both detection and recognition while speeding up the computation and reducing the model size.

In our future work, as the proposed MorseNet is task-oriented, it can be easily adjusted to apply to other signals with similar tasks instead of only Morse. In addition, for separate signal detection or recognition tasks, the corresponding branch divided from MorseNet could also be a good choice. Moreover, since there are more and more research studies that use the multi-task approach [35], [36], our unified network could also provide a new scheme for the multi-task architecture, which can be generalized to other problems consisting of multiple subtasks that are complementary.

REFERENCES

[1] S. Dey, K. M. Chugg, and P. A. Beerel, "Morse code datasets for machine learning," in *Proc. 9th Int. Conf. Comput. Commun. & Netw. Technol. (ICCCNT)*, Bangalore, India, 2018, pp. 1-7.

[2] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: a survey," *IEEE Commun. Surv. Tutor.,* vol. 21, no. 3, pp. 2224-2287, 2019.

[3] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescape, "MIMETIC: mobile encrypted traffic classification using multimodal deep learning," *Comput. Netw.,* vol. 165, p. 106944, 2019.

[4] Z. Wei, K. Jia, and Z. Sun, "An automatic detection method for morse signal based on machine learning," in *Proc. Int. Conf. Intell. Inf. Hiding Multimedia Signal Process*., Cham, Switzerland, 2017, pp. 185–191.

[5] Z. Wei, K. Jia, and Z. Sun, "An automatic detection method for Morse signal based on machine learning," in *Proc. Int. Conf. Intell. Inf. Hiding & Multimed. Sig. Process.*, Matsue, Japan, 2017, pp. 185-191.

[6] Z. Sun, Y. Wang, Z. Gong, and K. Jia, "Fast detection and recognition of Morse signal in wideband environment," *Comput. Eng. Appl.,* vol. 53, no. S2, pp. 92-96, 2017.

[7] Y. Yuan, Z. Sun, Z. Wei, and K. Jia, "DeepMorse: A deep convolutional learning method for blind Morse signal detection in wideband wireless spectrum," *IEEE Access,* vol. 7, pp. 80577-80587, 2019.

[8] X. Zha, H. Peng, X. Qin, G. Li, and S. Yang, "A deep learning framework for signal detection and modulation classification," *Sensors,* vol. 19, no. 18, p. 4042, 2019.

[9] A. Singh.A, "Deep learning framework for signal detection and modulation recognition," *J. Inf. & Comput. Sci.*, vol. 10, no. 3, pp. 327-333, 2020.

[10] W. Li, K. Wang, and L. You, "A deep convolutional network for multi-type signal detection in spectrogram," 2020, *Preprints: 2020050215.* [Online]. Available: https://www.preprints.org/manuscript/202005.0215/v1

[11] Y. Wang, Z. Sun, and K. Jia, "An automatic decoding method for Morse signal based on clustering algorithm," in *Proc. Int. Conf. Intell. Inf. Hiding & Multimed. Signal Process.*, 2016, pp. 21-23.

[12] L. You, W. Li, W. Zhang, and K. Wang, "Automatic decoding algorithm of Morse code based on deep neural network," *J. Electron. & Inf. Technol.,* to be published.

[13] X. Liu, D. Liang, S. Yan, D. Chen, Y. Qiao, and J. Yan, "FOTS: fast oriented text spotting with a unified network," in *Proc. Comput. Vis. & Pattern Recognit. (CVPR)*, 2018, pp. 5676-5685.

[14] C. Fan, B. Wen, and B. Wang, "Morse encoder of TPU used in air traffic management of civil aviation," *J. Electron. Meas. Instrum.,* vol. 18, no. 3, pp. 47-50, 2004.

[15] W. Ma, J. Zhang, and H. Wang, "Automatic decoding system of Morse code," *Ordnance Ind. Autom.,* vol. 26, no. 6, pp. 51-52,55, 2007.

[16] X. Zhou, G. Li, Y. Jiang, and L. Zeng, "Automatic reception of high-frequency CW telegraph with support vector machine," in *Proc. 2nd Int. Conf. Future Comput. & Commun.*, Wuhan, China, 2010, pp. V3-335-V3-338.

[17] G. Li, X. Zeng, X. Zhou, L. Zeng, and Y. Jiang, "Adaptive filtering of weak high-frequency CW telegraph signal," *J. Electron. Sci. Technol. Univ. China,* vol. 039, no. 2, pp. 227-231,250, 2010.

[18] P. Zahradnik and B. Šimák, "Implementation of Morse decoder on the TMS320C6748 DSP development kit," in *Proc. 6th Conf. European Embed. Des. Educ. Res.*, 2014, pp. 128-131.

[19] J. Lin, G. Li, X. Zhou, and L. Zeng, "Automatic detection of high-frequency CW telegraph signal in strong noise environment," *J. Chongqing Posts Telecommun. Univ.,* vol. 50, no. 5, pp. 505-509, 2008.

[20] S. Xiao and Y. Gao, "Multiplexed Morse telegraph automatic decoding based on wavelet transform," *Comput. Digit. Eng.,* vol. 45, no. 4, pp. 632-636, 2017.

[21] X. Yue, C. Zheng, and D. Chen, "Signal detection in short-wave telegraph for discrete Gabor spectrum," *J. Data Acquis. Process.,* vol. 14, no. 1, p. 22, 1999.

[22] S. Qu, H. Liu, and X. Zhang, "Morse recognition algorithm based on k-means," in *Proc. Conf. Cross Strait Quad Reg. Radio Sci. & Wirel. Technol. (CSQRWC)*, Taiyuan, China, 2019, pp. 1-2.

[23] C. Luo and D. Fuh, "Online Morse code automatic recognition with neural network system," in *Proc. 23rd Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Istanbul, Turkey, 2001, pp. 684-686.

[24] X. Wang, M. Zhang, H. Zhou, X. Lin, and X. Ren, "A robust real-time automatic recognition prototype for maritime optical Morse-based communication employing modified clustering algorithm," *Appl. Sci.,* vol. 10, no. 4, p. 1227, 2020.

[25] G. Li, J. Lin, and X. Zhou, "Robust recognition of high-frequency CW signal corrupted by impulsive noise," in *Proc. 9th Int. Conf. Fuzzy Syst. & Knowl. Discov.*, Chengdu, China, 2012, pp. 1929-1933.

[26] C. Yang, L. Jin, and L. Chuang, "Fuzzy support vector machines for adaptive Morse code recognition," *Méd. Eng. & Phys.,* vol. 28, no. 9, pp. 925-931, 2006.

[27] X. Wang, Z. Qi, M. Cheng, and J. Xiong, "Automatic Morse code recognition under low SNR," in *Proc. Int. Conf. Mech. Electron. Control. Autom. Eng. (MECAE)*, 2018.

[28] "WR-G39DDCe 'EXCELSIOR'." WiNRADiO Communications. https://www.winradio.com/home/g39ddce.htm

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. Comput. Vis. & Pattern Recognit. (CVPR)*, 2016, pp. 770-778.
[30] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, 2017, pp. 2999-3007.
[31] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.,* vol. 45, no. 11, pp. 2673-2681, 1997.
[32] A. Graves, S. Fernández, and F. Gomez, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proc. 23rd Int. Conf. Mach. Learn.*, 2006, pp. 369-376.
[33] M. Abadi *et al.,* "Tensorflow: Large-scale machine learning on heterogeneous distributed system," 2016, *arXiv: 1603.04467*. [Online]. Available: https://arxiv.org/abs/1603.04467
[34] D. Kingma. and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv: 1412.6980*. [Online]. Available: https://arxiv.org/abs/1412.6980
[35] J. Zhang, Y. Zheng, J. Sun, and D. Qi, "Flow prediction in spatio-temporal networks based on multitask deep learning," *IEEE Trans. Knowl. Data Eng.,* vol. 32, no. 3, pp. 468-478, 2020.
[36] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescapé, "Toward effective mobile encrypted traffic classification through deep learning," *Neurocomputing,* vol. 409, pp. 306-315, 2020.

WEIHAO LI received the B.S. degree from Information Engineering University in 2018, where he is currently pursuing the Ph.D. degree in information and communications engineering. His research interests are in deep learning and signal processing.

KEREN WANG received the Ph.D. degree from National Key Laboratory of Science and Technology on Blind Signal Processing, in 2014. His research interests are in deep learning and signal processing.

LING YOU received the Ph.D. degree from Information Engineering University, in 2000. His research interests are in signal analysis and processing.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3020860, IEEE Access

TABLE 1. Summary of related work in Morse detection and Morse recognition.

Task	Technique	References	Characteristics
Morse detection	Envelope detection	Fan *et al.* [14]	• The earliest method. • Fast speed but weak noise resistance.
	Phase-locked loop	Ma *et al.* [15]	• Being able to track the signal frequency. • Needing signal frequency estimation in advance. • Sensitive to interference.
	Filter	• Kalman filtering [16] • Adaptive filtering [17]	• Denoising signal. • Needing signal frequency estimation in advance. • Powerless for unstable frequencies.
	Signal transformation	• Fourier transform [18] • Complex variance spectrum [19] • Wavelet transform [20]	• Obtain frequency distribution but no time information. • Cannot effectively distinguish Morse from other signals.
	Discrete Gabor transform	Yue *et al.* [21]	• Too old to work well
	Energy detection + Classifier	Wei *et al.* [5], Sun *et al.* [6], Yuan *et al.* [7]	• Detecting Morse in spectrogram. • Depending greatly on energy detection performance. • Obtain frequency distribution but no time information. • The two layer CNN classifier in [7] has obtained the best classification effect.
	DL-based object detector	• SSD [8], [9]	• Detecting multi-type signals in spectrograms by BBoxes and identifying types. • Predicting incomplete BBox for horizontally long signals. • Time-consuming.
	Centerline-based neural network	Li *et al.* [10] (our earlier work)	• Detecting multi-type signals in spectrogram by BBoxes and identifying types. • Predicting complete BBox for horizontally long signals. • Get better performance in accuracy and speed than traditional DL-based object detectors.
Morse recognition	Obtaining code lengths + Classifying code types + Code-to-character table look up	Obtaining code lengths: • Tracking signal waveform in time domain [23]–[25] • Spectrogram + image processing [5], [6], [11] Classifying code types: • Gunther algorithm [20] • SVM [16], [25] • K-means cluster [6], [11], [22], [24] • C-means cluster [6]	• Multi-stage processing. • "Obtaining code lengths" is the main bottleneck that is sensitive to interference. • Code-type classifiers use only code lengths as features without context information.
	HMM + DNN	Wang *et al.* [27]	• Direct character level recognition. • Limited performance.
	CRNN	You *et al.*[12] (our earlier work)	• Direct character level recognition. • Fully DL-based method. • State-of-the-art performance.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3020860, IEEE Access

TABLE 2. The specific structure of the recognition branch. All of the convolutions are followed by batch normalization and ReLU activation. Max pooling only reduces the height axis of the feature map.

Layer	Kernel ([size, stride])	Out channels
conv_bn_relu1	[(3,3), (1,1)]	64
max_pool1	[(2,1), (2,1)]	64
conv_bn_relu2	[(3,3), (1,1)]	128
max_pool2	[(2,1), (2,1)]	128
conv_bn_relu3	[(3,3), (1,1)]	256
max_pool3	[(2,1), (2,1)]	256
BLSTM		256
FC		num_characters

TABLE 3. Experimental dataset description.

Name	5M	7M	9M	11M
Amount (training/testing)	2000/1000	2000/1000	2000/1000	2000/1000
Converted from	4-6M	6-8M	8-10M	10-12M
Sample rate	9kHz	9kHz	9kHz	9kHz
Time duration	15s	15s	15s	15s
Frequency band	4.5kHz	4.5kHz	4.5kHz	4.5kHz
Image size	749×512	749×512	749×512	749×512
Containing Morse per #	0-4	0-4	0-4	0-4

TABLE 4. Quantitative comparison results on four datasets. The input of recognition is the output of detection. "Cross-data" refers to training on the 5M dataset and testing on the 11M dataset. SSD is only compared in detection.

Method	Dataset	Detection			Recognition	
		Precision	Recall	F1-score	CER	WER
ECIK	5M	0.8237	0.7539	0.7873	0.2633	0.5762
	7M	0.8833	0.7564	0.8150	0.2459	0.5208
	9M	0.8340	0.7289	0.7779	0.2329	0.5171
	11M	0.8461	0.7267	0.7819	0.2502	0.5703
	Cross-data	0.7258	0.6366	0.6783	0.3007	0.5808
SSD	5M	0.8425	0.8014	0.8214		
	7M	0.8614	0.8167	0.8385		
	9M	0.8863	0.8542	0.8700		
	11M	0.8671	0.8264	0.8463		
	Cross-data	0.7981	0.7641	0.7807		
Our Two-Stage	5M	0.9021	0.8703	0.8859	0.1356	0.3590
	7M	0.9033	0.8804	0.8917	0.1247	0.3124
	9M	0.9162	0.8819	0.8987	0.1184	0.3025
	11M	0.8994	0.8663	0.8825	0.1243	0.3322
	Cross-data	0.8342	0.8286	0.8314	0.1478	0.3880
MorseNet	5M	**0.9484**	**0.9242**	**0.9361**	**0.0625**	**0.1999**
	7M	**0.9499**	**0.9262**	**0.9379**	**0.0624**	**0.2018**
	9M	**0.9572**	**0.9380**	**0.9475**	**0.0612**	**0.1967**
	11M	**0.9400**	**0.9173**	**0.9285**	**0.0594**	**0.1967**
	Cross-data	**0.8931**	**0.8714**	**0.8821**	**0.0827**	**0.2362**

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3020860, IEEE Access

TABLE 5. Speed and model size compared for different methods. "D+R" refers to the detection + recognition task. The model size of ECIK is not provided since the neural network is a small part of its overall architecture. D + R speed and model size of SSD are not provided since it is used for only the detection task.

Method	Speed			Params
	Use GPU	Detection	D + R	
ECIK	GPU	64.10 fps	15.92 fps	
	No GPU	12.52 fps	8.70 fps	
SSD	GPU	30.80 fps		
	No GPU	3.14 fps		
Our Two-Stage	GPU	61.73 fps	5.99 fps	40.62 M
	No GPU	7.80 fps	4.00 fps	
MorseNet	GPU	61.73 fps	7.30 fps	25.73 M
	No GPU	7.80 fps	5.57 fps	

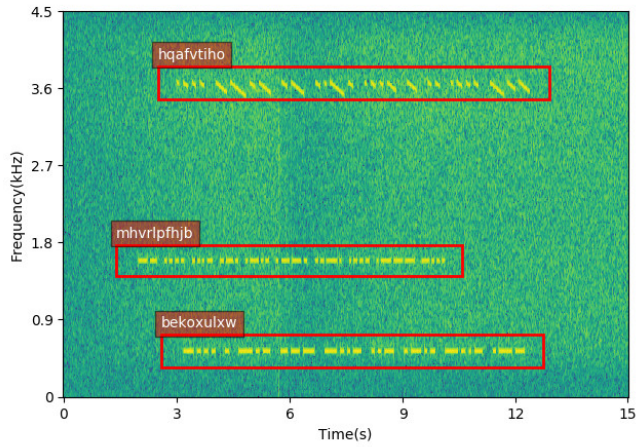

FIGURE 1. An instance of an input spectrogram. We label each Morse signal with a 400 Hz high, slightly longer box and the corresponding text.

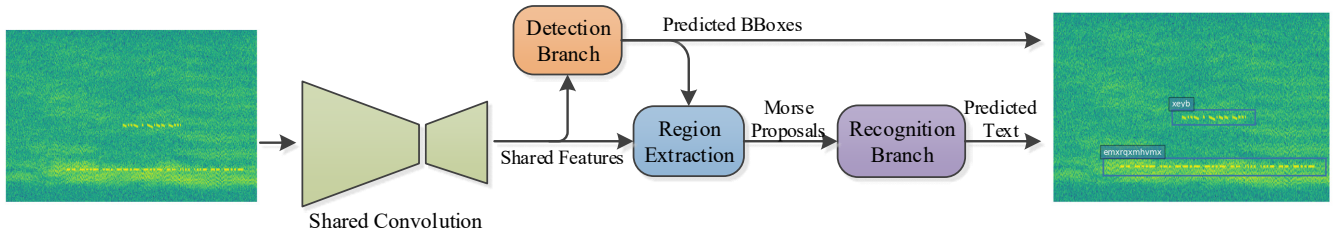

FIGURE 2. Overall architecture.

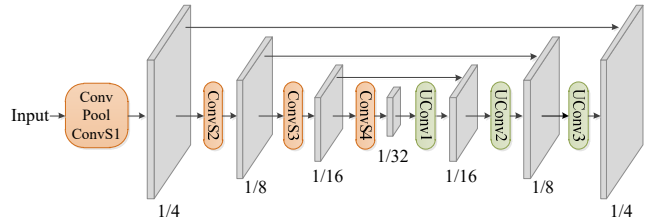

FIGURE 3. General structure of the shared convolution. "ConvS" refers to the convolution stage, which contains a 4-layer CNN and residual structures. "UConv" refers to up-convolution, and here, it is a transpose convolution. All of the convolutions are followed by batch normalization and ReLU activation.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3020860, IEEE Access

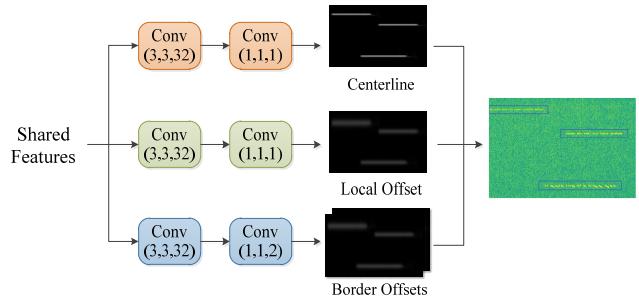

FIGURE 4. **Architecture of the detection branch.**

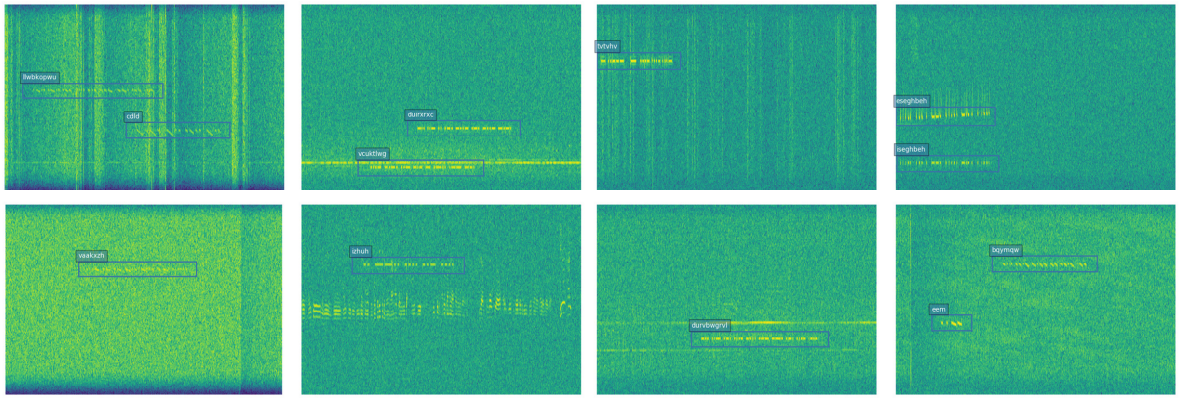

FIGURE 5. **Detection and recognition results of MorseNet under four harsh circumstances.**

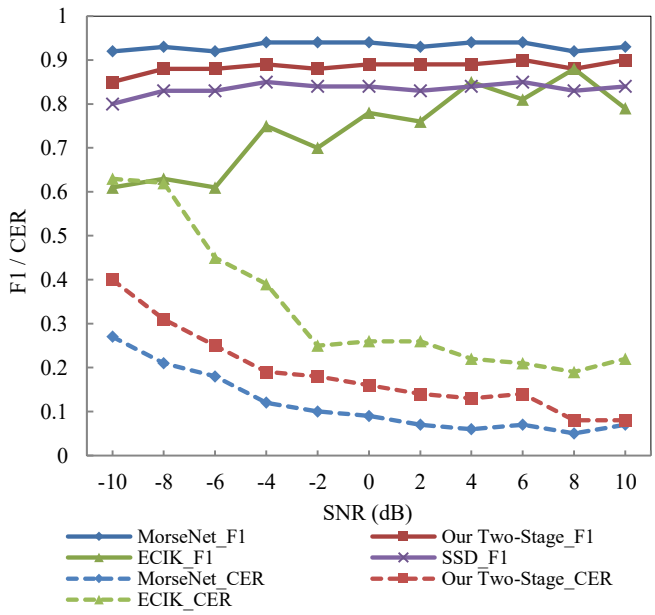

FIGURE 6. **F1-score and CER results versus the SNR. The SSD is compared only in the detection.**

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3020860, IEEE Access

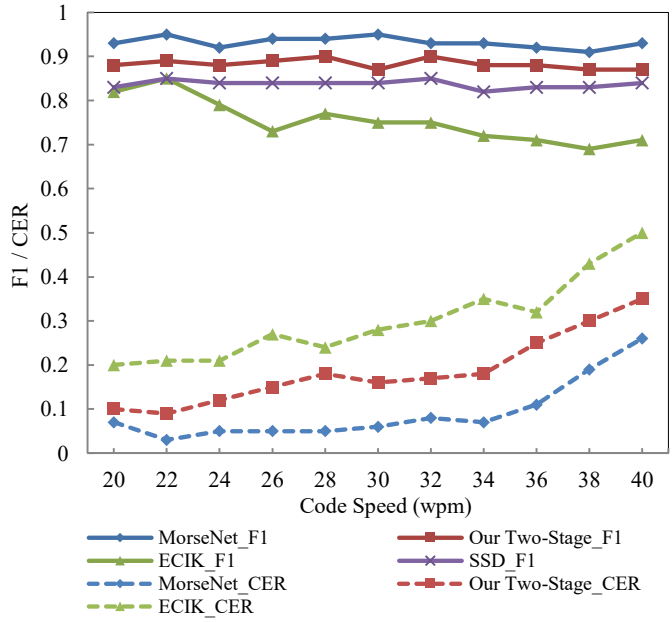

FIGURE 7. F1-score and CER results versus the code speed. The SSD is compared only in the detection F1.

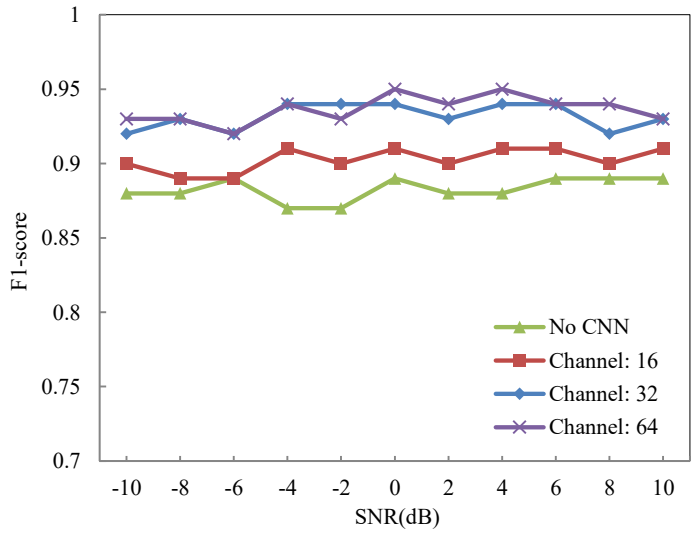

FIGURE 8. F1-score under different channels of the first CNN layer in the detection branch.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3020860, IEEE Access

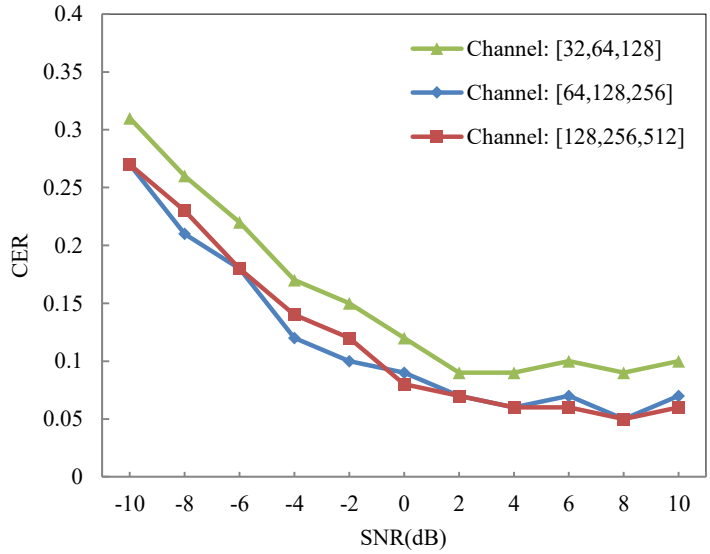

FIGURE 9. **CER under different channels of the three CNN layers in the recognition branch.**

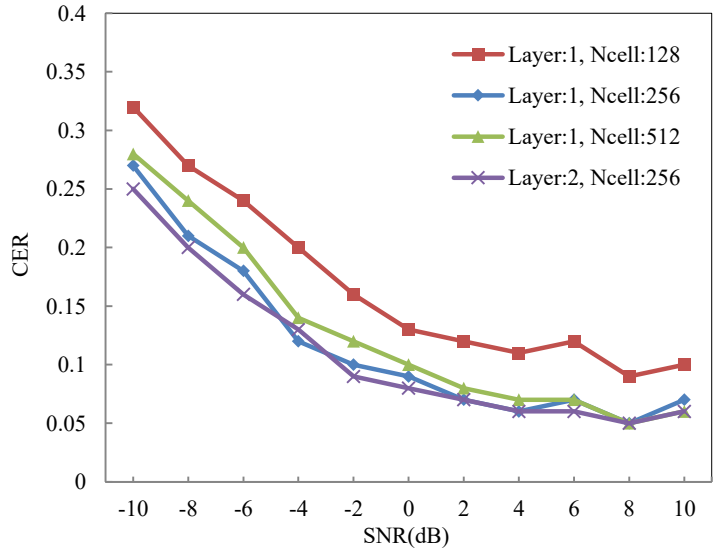

FIGURE 10. **CER under different layers and cell numbers of the BLSTM layer in the recognition branch.**