ICIC Express Letters
Volume **16**, Number **6**, June **2022**

ICIC International ©2022 ISSN 1881-803X
pp. **647–655**

MULTI-LANGUAGE VIDEO SUBTITLE RECOGNITION WITH CONVOLUTIONAL NEURAL NETWORK AND LONG SHORT-TERM MEMORY NETWORKS

Thanadol Singkhornart and Olarik Surinta*

Multi-agent Intelligent Simulation Laboratory (MISL)
Department of Information Technology
Faculty of Informatics
Mahasarakham University
Khamriang Sub-District, Kantarawichai District, Mahasarakham 44150, Thailand
63011283003@msu.ac.th; *Corresponding author: olarik.s@msu.ac.th

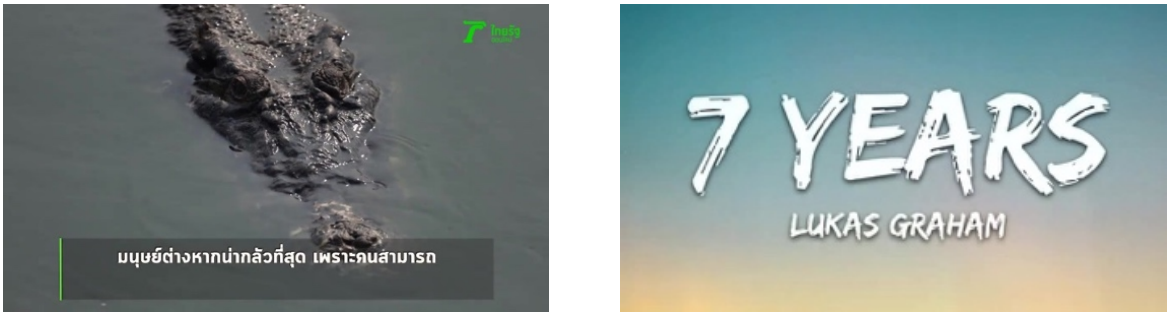
Received October 2021; accepted December 2021

Abstract. *Nowadays, many videos are published on Internet channels such as Youtube and Facebook. Many audiences, however, cannot understand the contents of the video, maybe due to the different languages and even hearing impairment. As a result, subtitles have been added to videos. In this paper, we proposed deep learning techniques, which are the combination between convolutional neural network (CNN) and long short-term memory (LSTM) networks, called CNN-LSTM, to recognize video subtitles. We created the simplified CNN architecture with 16 weight layers. The last layer of the CNN was downsampling using max-pooling before sending it to the LSTM network. We first trained our CNN-LSTM architecture on printed text data which contained various font styles, diverse font sizes, and complicated backgrounds. The connectionist temporal classification was then used as a loss function to calculate the loss value and decode the output of the network. For the video subtitle dataset, we collected 24 videos from Youtube and Facebook, containing Thai, English, Arabic, and Thai numbers. The dataset also contained 157 characters. In this dataset, we extracted 4,224 subtitle images from the videos. The proposed CNN-LSTM architecture achieved an average character error rate of 9.36%.*
Keywords: Video subtitle text recognition, Convolutional neural networks, Long short-term memory network, Connectionist temporal classification

1. **Introduction.** Video media has been published on various channels such as YouTube, Facebook, and Instagram. It gives the audience friendly options to choose and watch freely. Nowadays, video subtitles have been added to the videos to make them accessible to a broad audience, including foreigners and the hearing impaired. Significantly, adding subtitles increases the audience watching the video content and the video creator also received increased revenue from more video views. Some examples of the video subtitles are shown in Figure 1.

In recent years, deep learning methods, such as convolutional neural network (CNN) and long short-term memory (LSTM), have been proposed to address text and character recognition. Chamchong et al. [1] proposed a hybrid deep neural network combined with CNN and recurrent neural network (RNN). They designed hybrid deep neural networks with a tiny CNN weight layer. It included two CNN weight layers and each layer consists of 16 feature maps. The last CNN layer was combined with two layers of a bidirectional gated recurrent unit (Bi-GRU). It was called the 3CNN+BiGRU network. They trained the models using the connectionist temporal classification (CTC) loss function on Thai ancient manuscripts. The result showed that the tiny 3CNN+BiGRU obtained the character-level error rate (CER) value of 11.9%. Yan and Xu [2] proposed using the residual network

DOI: 10.24507/icicel.16.06.647



(a) (b)

FIGURE 1. Examples of a subtitle text appearing in the video: (a) The subtitle text appearing at the bottom of the video and (b) the subtitle text appearing in the area of interest (source: Youtube video)

(ResNet) architecture, Bi-GRU, and CTC to recognize Chinese and English subtitle texts in video images. It obtained an accuracy of 92.3% on the ICDAR2003 and 89.2% on the ICDAR2013.

Gan et al. [3] proposed a 1D-CNN and temporal convolutional recurrent network, called 1D-TCRN, to recognize in-air handwritten Chinese text on a large-scale IAHCT dataset. In this model, the two 1D residual convolution blocks were applied. These two blocks were connected as sequence layers. Hence, this architecture was then connected with LSTM and CTC layers. This network could recognize 2,565 characters of Chinese handwritten text. In [4], the subtitle left/right boundary detection discovered text window regions, called the CNN ensemble algorithm. First, a sliding window slides through the text windows computing the deep features using the CNN architecture and sending these features to classify as text or not-text using a support vector machine algorithm. The CNN ensemble algorithm can determine the text region and recognize the characters at the same time.

Zhang et al. [5] invented a scale-aware hierarchical attention network, called SaHAN, for scene text recognition. This network included two schemes: encoder and decoder. For the encoder, a deep pyramid convolutional recurrent neural network was proposed to create the multi-scale features. The smallest features were then converted to 1D vectors to learn semantic information in the bi-directional LSTM (Bi-LSTM). For the decoder, the semantic information and multi-scale features were transferred to the hierarchical attention decoder. It included two stages in the hierarchical attention decoder: 1D and 2D. Hence, the output of the 1D attention decoder was trained by the GRU and predicted the sequence label.

This paper aims to experiment with subtitle recognition that transforms the subtitle text image into text format. We propose CNN and LSTM architectures for recognition of Thai and English video subtitle images. The contributions of this paper can be summarized as follows.

- We propose the CNN and LSTM architectures, namely CNN-LSTM architecture for text line recognition. For the CNN architecture, we modify the VGGs architectures and then compare the experimental results with the method proposed by Chamchong et al. [1]. The experimental results show that our CNN-LSTM architecture obtains a lower character-level error value than Chamchong et al. [1].
- This paper aims to provide the new standard Thai and English languages video subtitle dataset for subtitle text recognition. The video subtitle dataset contains 4,224 images and includes 157 characters.

Outline of the paper. This paper is organized in the following way. Section 2 presents the proposed CNN-LSTM architecture. Section 3 describes video subtitle dataset used in

the experiments. The experimental results are explained in Section 4. The conclusion is presented in Section 5.

2. **Proposed CNN-LSTM Architecture.** This section explains the framework of video subtitle recognition. Two main architectures, CNN and LSTM, together called CNN-LSTM architecture, are proposed. The deep features are extracted using the VGGNet architecture and the Bi-LSTM network and transformed using the softmax function into a probability distribution. Hence, this architecture learns the subtitle text images and predicts the sequence of the text using the CTC as a loss function. The detail of the proposed CNN-LSTM architecture is as follows.

2.1. **Convolutional neural network.** CNN is a well-known deep learning technique used to address many research domains, such as image classification, image segmentation, and speech recognition. In recent years, many CNN architectures have been proposed, including EfficientNet [6], InceptionResNet [7], and NASNet [8]. The VGG architecture is involved in our proposed architecture. The details of the VGG architecture are as follows.

In 2015, Simonyan and Zisserman [9] from the Visual Geometry Group (VGG), University of Oxford, proposed a stack of a convolutional neural network called VGGNet, which consists of 16-19 weight layers that were computed with small convolution filters. The input of the network was a fixed size of 224×224 pixels. In each weight layer, the size of the weight layer was downsampled using the max-pooling process. The smallest size of the weight layer was 7×7 pixels. Then, the weight layers were followed by the three fully connected (FC) layers. The FC layers had 4,096, 4,096, and 1,000 channels, respectively. The softmax function was the last layer of the network. The VGG architecture is shown in Figure 2.

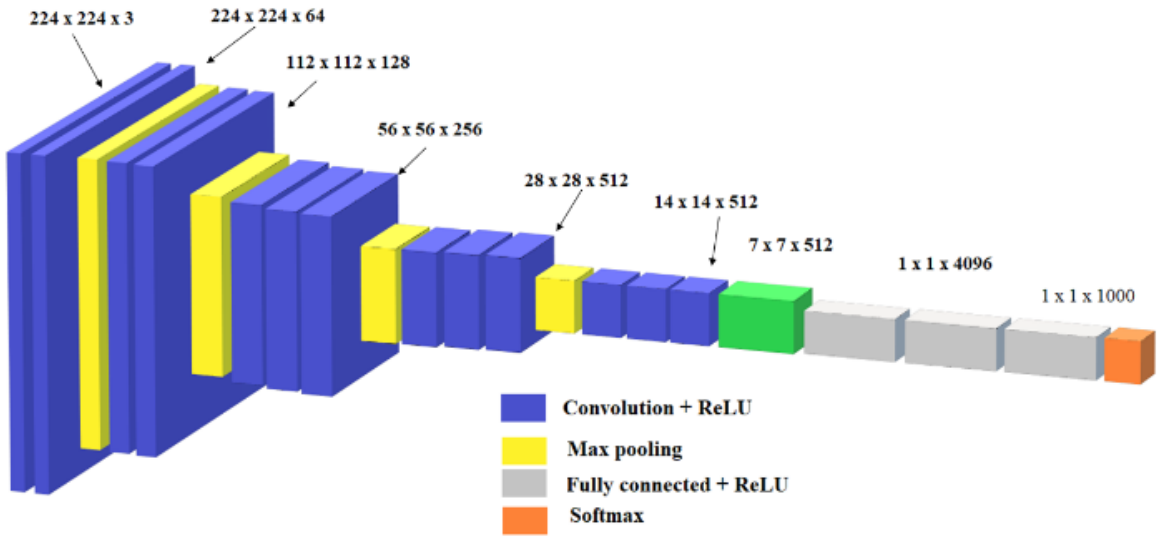


FIGURE 2. (color online) Illustration of the VGG architecture

2.2. **Long short-term memory.** Hochreiter and Schmidhuber [10] proposed a new type of RNN, namely LSTM, to address the vanishing gradient problem found when training with the RNN network and long sequence data. Therefore, the LSTM architecture included the feedback connection to proceed with long sequence data, such as speech and video. It includes gates, such as input gate, output gate, and forget gate, to control the information flow. As a result, it can learn from the sequence data and keep or throw the data away if the data is not essential. An illustration of the LSTM network is shown in Figure 3.

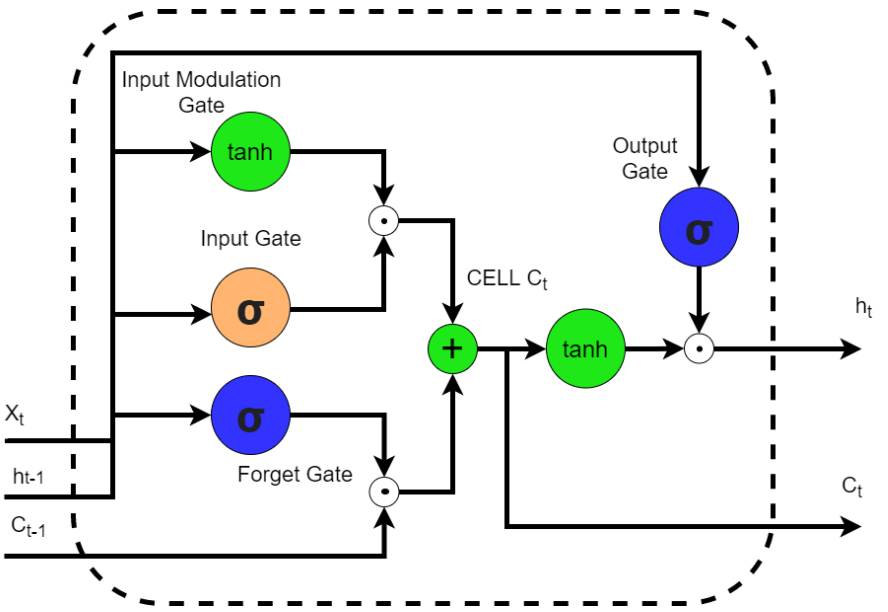


FIGURE 3. Illustration of the LSTM network

2.3. **Bi-directional LSTM.** The Bi-LSTM [11] network is an extension of the LSTM network. It is a combination of two independent LSTM networks (forward state and backward state) to process sequence data in two directions. The outputs of the two states are not attached to inputs of the reverse direction states. The Bi-LSTM network is shown in Figure 4.

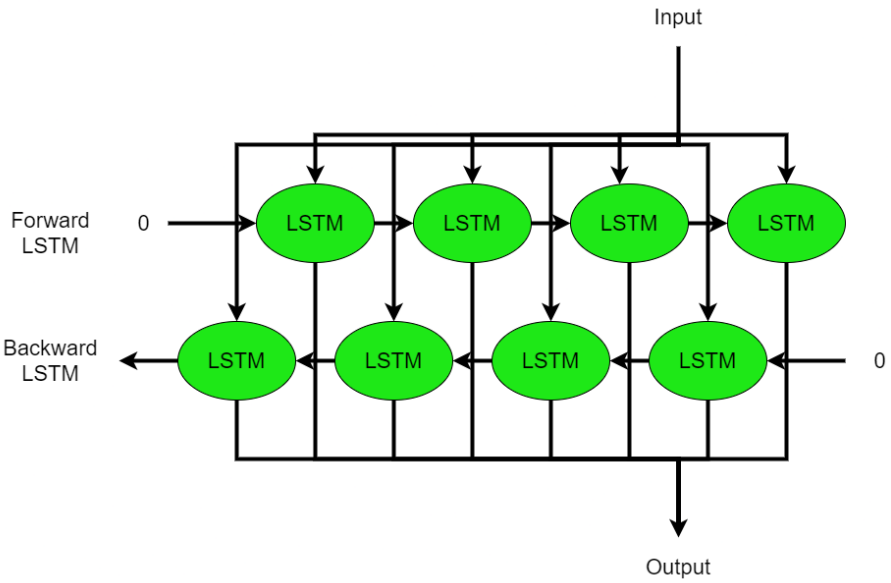


FIGURE 4. Illustration of the Bi-LSTM network

2.4. **Connectionist temporal classification.** CTC [12] is a loss function used for training the LSTM networks to address sequence problems. It allows predicting the continuous output. Recently, it was proposed to classify sequence data, including handwritten text and speech. The CTC function examines the blank or no label, so it does not translate the blank or no label into other labels.

The detail of the CNN-LSTM architecture proposed to recognize video subtitle images is shown in Table 1.

TABLE 1. Our proposed CNN-LSTM architecture

Stage	Operators	Resolution	Channels	Layers
1	Conv 3×3	32×379	64	2
2	Max-pooling 2×2			1
3	Conv 3×3	16×189	128	2
4	Max-pooling 2×2			
5	Conv 3×3	8×94	256	3
6	Max-pooling 2×2			
7	Conv 3×3	4×94	512	3
8	Max-pooling 2×1			
9	Bi-LSTM	94	256	2
10	Dense & Softmax Function	157		1
11	CTC Loss Function			

3. **Video Subtitle Dataset.** The video subtitle images used in this experiment were collected from 24 videos shared on Facebook and Youtube. The subtitle text included Thai and English languages, including Thai characters, Roman characters, Thai numerals, Arabic numerals, and special characters with 157 characters in total, as shown in Table 2.

TABLE 2. Set of characters for Thai and English subtitle text recognition

Type of characters	Characters
Thai consonant	ก ข ฃ ค ฅ ฆ ง จ ฉ ช ซ ฌ ญ ฎ ฏ ฐ ฑ ฒ ณ ด ต ถ ท ธ น บ ป ผ ฝ พ ฟ ภ ม ย ร ล ว ศ ษ ส ห ฬ อ ฮ
Thai vowel	ะ ั า ำ ิ ี ึ ื ุ ู ฺ เ แ ๅ
Thai tone	่ ้ ๊ ๋
Thai punctuation marks	ๆ ฯ ์
Thai numeral	๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ ๐
Roman character	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z a b c d e f g h i j k l m n o p q r s t u v w x y z
Arabic numeral	1 2 3 4 5 6 7 8 9 0
Special character	. , ! ( ) - $ ฿ & : ? * “ ‘ (space)

In the data-preprocessing step, we converted all 24 videos to images and obtained 2,700 images with subtitle text. The size of the subtitle text image was 1280×720 pixels and it was stored in JPG format. Further, we generated the ground truth from 4,224 subtitle

FIGURE 5. Examples of subtitle text images and labels

images using the labelImg program. Also, the labels were then assigned to each subtitle image. Examples of subtitle text images and labels are shown in Figure 5. Note that the number before the label is the order of the subtitle text image.

4. **Experimental Results.** In this section, we employed the TensorFlow framework running on Google Colab with GPU support for all the experiments. The video subtitle dataset was divided into training and test sets using the 5-fold cross-validation method. The CER was applied for the evaluation. The lowest CER value was deemed the best result.

4.1. **Character-level error rate.** The CER [13] is a metric used to evaluate speech recognition, machine translation, and handwritten text recognition. The output of the text sequence predicted from the classifier can have a different length when compared with the reference text sequence. The CER value is calculated by the following equation:

$$CER = \frac{I + S + D}{N} \tag{1}$$

where I is the number of character insertions, S is the number of characters substituted D is the number of characters deleted, and N is the total number of characters in the reference text.

4.2. **Experiments of the CNN-LSTM architectures.** For the CNN-LSTM architectures, we created four CNN-LSTM models (Models A, B, C, and D) based on VGG16 and VGG19. We connected Bi-LSTM and Bi-GRU and computed models using the CTC loss function, as shown in Table 3. For the experiment, we then evaluated four CNN-LSTM models with different batch sizes of 32, 64, and 128.

Table 3. CNN-LSTM architectures applied in subtitle video recognition

Model A	**Model B**	**Model C**	**Model D**
16 weight layers	16 weight layers	19 weight layers	19 weight layers
Input (gray image), 32×379			
Conv 3×3, 64	Conv 3×3, 64	Conv 3×3, 64	Conv 3×3, 64
Conv 3×3, 64	Conv 3×3, 64	Conv 3×3, 64	Conv 3×3, 64
Max-pooling 2×2, stride 2			
Conv 3×3, 128	Conv 3×3, 128	Conv 3×3, 128	Conv 3×3, 128
Conv 3×3, 128	Conv 3×3, 128	Conv 3×3, 128	Conv 3×3, 128
Max-pooling 2×2, stride 2			
Conv 3×3, 256	Conv 3×3, 256	Conv 3×3, 256	Conv 3×3, 256
Conv 3×3, 256	Conv 3×3, 256	Conv 3×3, 256	Conv 3×3, 256
Conv 3×3, 256	Conv 3×3, 256	Conv 3×3, 256	Conv 3×3, 256
		Conv 3×3, 256	Conv 3×3, 256
Max-pooling 2×1, stride 1			
Conv 3×3, 512	Conv 3×3, 512	Conv 3×3, 512	Conv 3×3, 512
Conv 3×3, 512	Conv 3×3, 512	Conv 3×3, 512	Conv 3×3, 512
Conv 3×3, 512	Conv 3×3, 512	Conv 3×3, 512	Conv 3×3, 512
		Conv 3×3, 512	Conv 3×3, 512
Max-pooling 2×1, stride 1			
Bi-LSTM	Bi-GRU	Bi-LSTM	Bi-GRU
Bi-LSTM	Bi-GRU	Bi-LSTM	Bi-GRU
Dense & Softmax Function, 157			
CTC Loss Function			

Table 4 showed the experimental results with four CNN-LSTM architectures and three different batch sizes on the video subtitle dataset. The experiments showed that model C with a batch size of 128 achieved the best performance with a CER value of 9.36%.

TABLE 4. Character-level error rates of CNN-LSTM models on the video subtitle dataset

Batch size	**CER value (%)**			
	Model A	**Model B**	**Model C**	**Model D**
32	22.61	18.57	17.84	13.50
64	12.37	15.34	19.39	17.12
128	18.38	10.11	**9.36**	16.53

4.3. **Comparison with other text recognition architectures.** In this section, we compared two deep learning models based on architectures from Chamchong et al. [1], as shown in Table 5.

TABLE 5. The architectures based on Chamchong et al.

Model CC1	**Model CC2**
6 weight layers	6 weight layers
Input (gray image), 32×379	
Conv 3×3, 16	Conv 3×3, 16
Max-pooling 2×2	
Conv 3×3, 32	Conv 3×3, 32
Max-pooling 2×2	
Conv 3×3, 32	Conv 3×3, 32
Max-pooling 5×1	
Bi-GRU	Bi-LSTM
Bi-GRU	Bi-LSTM
Dense & Softmax Function, 157	
CTC Loss Function	

We experimented with three batch sizes consisting of 32, 64, and 128 to obtain the best performance on the subtitle video dataset. From the results in Table 6, it can be seen that model CC1, with the batch size of 32, achieved the best CER value of 17.35%.

TABLE 6. Character-level error rates of two models based on Chamchong architecture on the video subtitle dataset

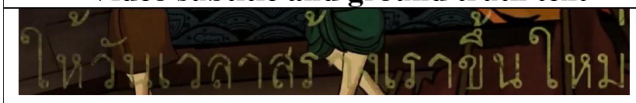
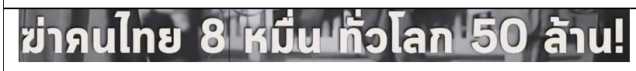
Batch size	**CER value (%)**	
	Model CC1	**Model CC2**
32	**17.35**	24.14
64	20.68	22.87
128	22.39	26.65

The best CNN-LSTM models that we found from Tables 4 and 6 were then selected to compare again with a 5-fold cross-validation technique. In Table 7, the results showed that our CNN-LSTM architecture (model C) with a batch size of 128 outperformed the deep learning model (model CC1) with a batch size of 32. Our CNN-LSTM architecture showed a better CER value of 10% more than with model CC1. However, model CC1 trained quite fast in only 23 minutes. It trained the model three times faster than our CNN-LSTM architecture. The predicted texts decoded from our CNN-LSTM (model C) and Chamchong (model CC1) architectures are shown in Table 8.

TABLE 7. Experimental results between our proposed CNN-LSTM architecture and previous work

Model	Batch size	CER value	Training time (min.)
C	128	**9.36±0.91**	115.43
CC1	32	19.13±1.37	23.12

TABLE 8. Examples of the predicted text from our proposed CNN-LSTM (Model C) and Chamchong (Model CC1) architectures

Video subtitle and ground truth text	Predicted text
 **Ground Truth Text:** ให้วันเวลาสร้างเราขึ้นใหม่	**Model C:** ให้ว อาสร้างเราขึ้นใหม่ **Model CC1:** ให้วรร้านเรานี่นใหม่
 **Ground Truth Text:** ฆ่าคนไทย 8 หมื่น ทั่วโลก 50 ล้าน!	**Model C:** ฆ่าคนไทย 8 หมื่น ทั่วโลก 50 ล้าน!) **Model CC1:** ข่าคนไทย 2หมื่น กั้วโลก 50 ล้านป้
 **Ground Truth Text:** ทฤษฎีสมคบคิด MH370	**Model C:** ทฤษฎีสมคบคิด MH370 **Model CC1:** ทฤษฎีสมคบคิด MH370
 **Ground Truth Text:** State Quarantine	**Model C:** State Quarantine **Model CC1:** State Quarantine

5. **Conclusions.** In this paper, we evaluated the deep learning method, called CNN-LSTM architecture, on a video subtitle dataset. First, we created four CNN architectures based on VGG16 and VGG19 networks (Models A-D). Subsequently, the last layer of each model was combined with Bi-LSTM or Bi-GRU and trained using the CTC loss function. There were 16 weight layers in total. Model C combined with Bi-LSTM provided a CER value of 9.36%. Second, we recreated two CNN models (Models CC1 and CC2) based on Chamchong et al. [1]. Model CC1, which included only six weight layers, obtained a CER value of 17.35%. Finally, in consideration of the best experiments, Model C and Model CC1 were then selected. We experimented with the 5-fold cross-validation method. The experimental results showed that our proposed CNN-LSTM (Model C) also obtained a low CER value compared to Model CC1 architecture. Due to training time, however, Model CC1 was much faster than our proposed architecture. This was because of the number of weight layers.

In future work, to improve the performance of video subtitle recognition, we will experiment with hybrid CNN architecture [14] and data augmentation techniques. We are interested in using computational linguistic methods for automatic spelling correction.

Acknowledgment. This research project was financially supported by Mahasarakham University.

REFERENCES

[1] R. Chamchong, W. Gao and M. D. McDonnell, Thai handwritten recognition on text block-based from Thai archive manuscripts, *The International Conference on Document Analysis and Recognition (ICDAR)*, pp.1346-1351, 2019.

[2] H. Yan and X. Xu, End-to-end video subtitle recognition via a deep residual neural network, *Pattern Recognition Letters*, vol.131, pp.368-375, 2020.

[3] J. Gan, W. Wang and K. Lu, In-air handwritten Chinese text recognition with temporal convolutional recurrent network, *Pattern Recognition*, vol.97, DOI: 10.1016/j.patcog.2019.107025, 2020.

[4] Y. Xu et al., End-to-end subtitle detection and recognition for videos in East Asian languages via CNN ensemble, *Signal Processing: Image Communication*, vol.60, pp.131-143, 2018.

[5] J. Zhang, C. Luo, L. Jin, T. Wang, Z. Li and W. Zhou, SaHAN: Scale-aware hierarchical attention network for scene text recognition, *Pattern Recognition Letters*, vol.136, pp.205-211, 2020.
[6] M. Tan and Q. V. Le, EfficientNet: Rethinking model scaling for convolutional neural networks, *The 36th International Conference on Machine Learning (ICML)*, pp.10691-10700, 2019.
[7] C. Szegedy, S. Ioffe, V. Vanhoucke and A. A. Alemi, Inception-v4, Inception-ResNet and the impact of residual connections on learning, *The 31st AAAI Conference on Artificial Intelligence (AAAI)*, pp.4278-4284, 2017.
[8] B. Zoph, V. Vasudevan, J. Shlens and Q. V. Le, Learning transferable architectures for scalable image recognition, *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.8697-8710, 2018.
[9] K. Simonyan and A. Zisserman, Very deep convolutional networks for large-scale image recognition, *The 3rd International Conference on Learning Representations (ICLR)*, pp.1-14, 2015.
[10] S. Hochreiter and J. Schmidhuber, Long short-term memory, *Neural Computation*, vol.9, no.8, pp.1735-1780, 1997.
[11] A. Graves and J. Schmidhuber, Framewise phoneme classification with bidirectional LSTM and other neural network architectures, *Neural Networks*, vol.18, nos.5-6, pp.602-610, 2005.
[12] A. Graves, S. Fernández, F. Gomez and J. Schmidhuber, Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks, *The 23rd International Conference on Machine Learning (ICML)*, pp.369-376, 2006.
[13] P. Dreuw, G. Heigold and H. Ney, Confidence- and margin-based MMI/MPE discriminative training for off-line handwriting recognition, *International Journal on Document Analysis and Recognition*, vol.14, no.3, pp.273-288, 2011.
[14] Y. T. Hafiyan, Afiahayati, R. D. Yanuaryska, E. Anarossi, V. M. Sutanto, J. Triyanto and Y. Sakakibara, A hybrid convolutional neural network-extreme learning machine with augmented dataset for DNA damage classification using comet assay from buccal mucosa sample, *International Journal of Innovative Computing, Information and Control*, vol.17, no.4, pp.1191-11201, 2021.