DOI: 10.21917/ijivp.2021.0339

AN END-TO-END TRAINABLE CAPSULE NETWORK FOR IMAGE-BASED CHARACTER RECOGNITION AND ITS APPLICATION TO VIDEO SUBTITLE RECOGNITION

Ahmed Tibermacine and Selmi Mohamed Amine
Department of Computer Science, Biskra University, Algeria

Abstract

The text presented in videos contains important information for a wide range of vision-based applications. The key modules for extracting this information include detection of text followed by its recognition, which are the subject of our study. In this paper, we propose an innovative end-to-end subtitle detection and recognition system for videos. Our system consists of three modules. Video subtitle are firstly detected by a novel image operator based on our blob extraction method. Then, the video subtitle is individually segmented as single characters by simple technique on the binary image and then passed to recognition module. Lastly, Capsule neural network (CapsNet) trained on Chars74K dataset is adopted for recognizing characters. The proposed detection method is robust and has good performance on video subtitle detection, which was evaluated on dataset we constructed. In addition, CapsNet show its validity and effectiveness for recognition of video subtitle. To the best of our knowledge, this is the first work that capsule networks have been empirically investigated for Character recognition of video subtitles.

Keywords:

Capsule Networks, Convolutional Neural Networks, Subtitle Text Detection, Text Recognition

1. INTRODUCTION

Given huge available amount of video via recording and audiovisual broadcasting devices, many promising applications like automatic video retrieval and summarization has attracted wide interest among researchers from different fields in present era. Since the videos contain plenty of information, among which semantic information is supplied by the text in it. The text present in the videos is of two forms: Scene Text and Graphics Text. Video subtitles are the graphics text added externally to the video and express its content.

Different from traditional printed document, extraction video subtitle is complicated by complex backgrounds, diverse fonts, variation of illumination, low resolution and contrast between texts and backgrounds. Video subtitle extraction includes two indispensable subtasks: subtitle detection and subtitle recognition, which are the focus of the proposed model in this paper. The detection subtask aims to identify the accurate location of subtitle regions and separating clean text region from the complex background. Nemours methods have been proposed for this subtask which can roughly be classified into two categories. The first category uses textural properties to detect video text [1-4], whereas, the second one is based on the connected component of text [5]-[7].

Furthermore, the subtitle recognition subtask requires identifying subtitle text in cropped text image; this is a classification problem with high confusion because of letter case and homoglyph confusion.

The use of machine learning to text recognition is not a new research field. Formerly, general classifiers such as support vector machines have been used for such task [8]. Recently, continuous development of deep learning methods has provided efficient recognition model which further led to higher accuracies for text recognition. The major deep learning models used in text recognition are mostly based on convolutional neural networks (CNNs). However, the major challenge of CNNs is their inability to recognize pose, texture and deformations of an image [9] or parts of the image. In addition, the pooling operations in CNN lose valuable information and also do not encode relative spatial relationships between features. CNNs are also more prone to adversarial attacks such as pixel perturbations [10] resulting in wrong classifications [11].

Meanwhile, in the image recognition field, capsule networks [9] [12] proved to be effective at understanding spatial relationships in high levels of data due to the use of capsules. The capsule network is a structured model that addresses limitations of CNNs. Capsules in capsule networks encapsulate all important information about the state of the features they are detecting in form of vector. Capsules take into consideration the spatial relationships between features and learn these relationships via dynamic routing [10]. In this paper, we have applied this network to the video subtitle recognition, and debate that it also has advantages in this domain.

The main contributions of this work are three-fold. First, we propose an end-to-end subtitle detection and recognition system for Latin (English) languages. Second, for text detection, we propose robust method, which extract the region where subtitle text is present. Accurate and complete text regions can then be obtained after set of image preprocessing operations. The proposed method is resistant to noise and capable of detecting text in various languages and fonts against complex backgrounds. Third, we apply capsule networks with dynamic routing to video subtitle recognition and achieve comparable results to previous methods.

Rest of the paper is organized as follows: section 2 reviews related work, section 3 provides a detailed description about the technique (or the algorithm) and methodologies applied to carry out the extraction and recognition task. Subsequently, we discuss our results in section 4, and finally, section 5 samples the conclusions that have been obtained after the study.

2. RELATED WORKS

2.1 TEXT DETECTION

Over the last two decades, researchers have proposed various methods for detecting texts in scene images and videos, these methods can be classified into two categories: Texture based methods [1]-[4] and component-based methods [5]-[7]. The first category considers texts as a special type of texture and takes advantage of their textural properties to discriminate between text and non-text region in the images and video. These methods are computationally expensive as all locations should be scanned. Furthermore, these methods often address horizontal texts and are generally sensitive to rotation and scale change. The second category primarily extract candidate components using different techniques such as: extreme region extraction or color clustering, and then eliminate false text components using handcrafted rules [13] [14] or automatically trained classifiers [15]-[17]. Component based methods are more efficient, because the number of treated components is relatively little. Moreover, these methods are insensitive to rotation, scale change and font variation. These methods have become the most popular scene text detection methods over the past few years.

2.2 TEXT RECOGNITION

In recent years, considerable text recognition methods have been proposed, and these methods can be roughly assorted into three categories: sequence-based recognition methods and word-based recognition methods and character-based recognition methods.

Image-Based Sequence Recognition (IBSR) methods were the mainstream in the field of text recognition for the past few years. These methods can progressively output character sequence. Shi et al. [18] applied convolutional neural networks (CNNs) and recurrent neural networks (RNNs) to extract image features and model image context information. Shi et al. [19] introduced attention-based sequence-to-sequence transformation networks to recognize scene text.

Image-Based Word Recognition (IBWR) methods aim to recognize a word as a whole. The representative work is that Jaderberg et al. [20] employed deep neural network to recognize word images. The accuracy of this method is very high, and it achieved the best performance on multiple datasets. However, the deficiencies of IBWR methods are that they do not recognize words other than predetermined words, restricted by glossary and it is also unsuitable to recognize texts without spaces between words such as Chinese.

Image-Based Character Recognition methods (IBCR) usually detect and recognize each character individually, then combine characters into a character sequence. Early method [14] adopted artificially designed features to locate and recognize characters. Yao et al. [15] used Strokelets feature to detect a single character, and utilize the Histogram of Oriented Gradient to recognize each character. Al- sharif et al. [16] proposed to combine CNN with Hidden Markov Model to recognize character and they achieved high performance on multiple datasets. The advantages of this method over others are that is more flexible and not limited to the length and spatial distribution of the recognized word sequence.

2.3 CAPSULE NETWORKS

The capsule network used to recognize video subtitle in this paper is simplified model based on Sabour's model [9]. It has already been successfully used in other application such as astronomy [21], autonomous cars [22], machine translation [23], handwritten recognition [24]-[26], intent detection [27] [28], mood and emotion detection [29] [30]. The spatio-temporal nature of traffic data expressed in images lends itself to the application of CapsNets for predicting traffic speed [31] [32] and abnormal driving [33] on a complex road network. Another challenging problem solved by CapsNets is environmental sound detection [34] [35]. CapsNets have found important applications in health [36] [37] and other important areas [38]-[41].

3. PROPOSED MODEL

In this section, we will describe the end-to-end system in detail, the synthetic data generation pipeline and the CapsNet ensemble. As illustrated in Fig.1, the end-to-end system consists of three imperative modules including subtitle region extraction, subtitle segmentation and subtitle recognition.

The subtitle region extraction module can be divided into the following steps: detection, localization and extraction. Detection step aims to find text in a given frame. Localization concentrates on the accurate position of text in the image and generating bounding boxes around the text. Last step focuses on separating clean region of interest from the image.

Once a clean image of subtitle is obtained, three types of segmentation are carried out: Line level Segmentation (LS), Word level Segmentation (WS) and Character level Segmentation (CS). First level refers to the splitting of the image containing video subtitle written in the form of lines, into lines patches. Second level aim to segment the image containing single line of sequence of words, into set of word patches. The intent behind this this kid of segmentation is to treating each word separately in the remaining stages. This helps maintain the integrity of the words. In the last level, image containing a single word is segmented into set of individual character patches.

For recognition module, we adopt IBCR approach to detect and recognize each character individually, and then combine characters into a character sequence (word). More details about of each module are explained below.

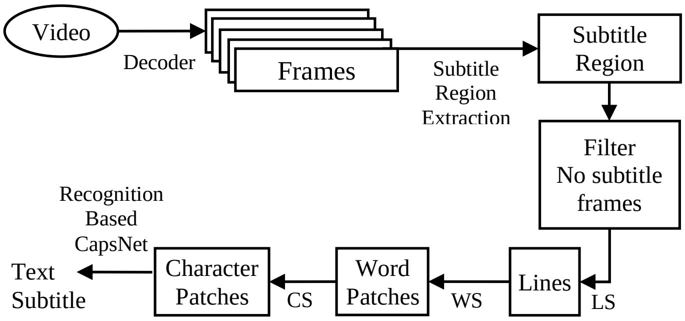


Fig.1. Flowchart for the proposed model

3.1 SUBTITLE REGION EXTRACTION

Initially, we decode video into frames. After that, we use an image processing approach using thresholding and contour filtering to obtain candidate subtitle regions of image, resulting in Fig.2(f). To realize, within reason of complexity and diversity of background, the video frame is converted to grayscale and Gaussian blurring (Fig.2(b)). Frame blurring is achieved by convolving the frame with a low-pass filter kernel consists of 9×9. It is useful for removing high frequency content from the frame and reduces detail. Then, the video frame is binarized to maintain the visibility and integrity of subtitle and keep text and background as separate as possible (Fig.2(c)). Because video contains a strong illumination gradient in different areas, we binarize frame by the Adaptive Gaussian Thresholding method, which dynamically determines the threshold of binarization. So, the threshold value $T(x,y)$ at pixel location (x,y) is then calculated using the formula given below:

$$T(x,y) = WA(x,y)\text{-}C \qquad (1)$$

where,

$WA(x,y)$ is Gaussian Weighted Average value of the $(b\times b)$ region.

$(b\times b)$ is a region around the pixel location, in this work b=9.

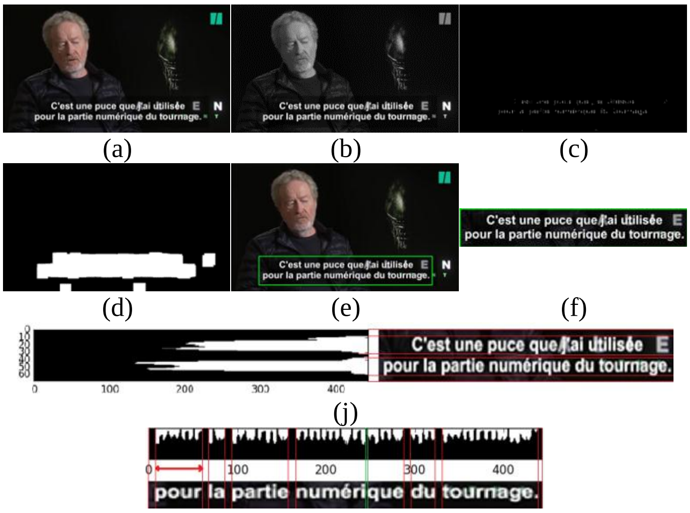
C is a constant which is subtracted from weighted average calculated and is fixed to 23.

Next we dilate to combine the adjacent text into a single contour (Fig.2(d)), by creating a rectangular structuring kernel (5×20) then dilate to form a single contour. The main effect of the dilation is to continuously increase the boundaries of subtitle regions. Thus areas of text expand in size while holes within those regions become smaller. The dilation of binary image A by structuring element B is defined by:

$$A\oplus B = \bigcup_{b\in B} A_b \qquad (2)$$

where, A_b is the translation of A by b. in this presented work, the dilation operation is repeated for 17 times.

From here we find all contours and choosing the one with the largest area as region of interest, when all counters areas are less then threshold (=40000) no subtitle region will be selected. If there is no subtitle region detected in the current video frame, the subtitle detection of the next video frame is directly performed. In the last, subtitle region are extracted (Fig.2(f)) and segmentation and recognition tasks will apply to it.


(a) (b) (c)

(d) (e) (f)

(j)

(h)

C'est une puce que j'ai utilisée
pour la partie numérique du tournage.
0
100
200
300
400

Fig.2. Proposed approach stages. Original (a), Grayscale (b) Thresholding (c), Dilation (d), Video subtitle localization (e), Subtitle region extraction (f), Horizontal projection (j) and Vertical projection (h)

3.2 SUBTILE SEGMENTATION

For all levels, we are going to use the Histogram Projection technique.

3.2.1 Line Level Segmentation:

To achieve the purpose of this segmentation level, we horizontally project the image. Rows that represent the gaps in-between the lines have high number of background pixels which correspond to higher peaks in the histogram. Rows which correspond to higher peaks in the histogram can be selected as the segmenting lines to separate the lines (Fig.2(j)).

3.2.2 Word Level Segmentation:

In this level, we vertically project the line image; columns which correspond to higher peaks in the histogram can be selected as the segmenting lines to separate the words (Fig.2(h), red lines). For segmenting words, higher peaks should be selected in such a way that they should span through a certain width (threshold) and neglects the thin gaps between the characters within the words. This is because there are higher peaks which correspond to the gaps between disconnected characters within a word, which we are not interested in this stage.

3.2.3 Character Level Segmentation:

Here, we use the same idea which we have used for word level segmentation by leveraging the small gap between the characters (Fig.2(h), green lines).

3.3 SUBTITLE RECOGNITION

In this section we tend to apply capsule networks to recognition of video subtitles and modifying it according to our purpose. The high-level view of our approach is depicted in Fig.3.

The architectural framework of our CapsNet model is composed of an encoder and a decoder, former of which comprises of stacked convolutional layers with rectified linear activation (ReLU) as the lower-level feature extractors, a PrimaryCaps layer for producing combinations of the above feature outputs and a CharCaps layer for the generation of the loss function and transformational weight matrix. Decoder constitutes of three Fully Connected layers (FC). The first two layers have the ReLU activation function while the last layer has the sigmoid activation unit. Both components effectively cooperate to recreate the original input image while handling with the accuracy and loss performance parameters.

The detailed technical functionality of each layer is as follows:

- *ReLU Convolutional Layer*: The layer has 256 kernels each with a bias term, stride of 1, size of 9×9×1 followed by the ReLU activation. The layer handles 20992 parameters and outputs 20×20×256 tensor.
- *PrimaryCaps layers*: The 8 capsule layer applies 9×9×256 convolutional kernels (with stride 2) to the 20×20×256 input volume while handling 1327168 parameters and outputs 6x6x8x8 tensor.
- *DataCaps Layers*: This 70 capsule layer takes as input the 6×6×8×8 tensor and as per inner workings of each capsule, a weight matrix is computed and 8 dimensional input spaces is mapped to the 16 dimensional capsule output space. The layer outputs a 16×70 matrix associated with 2620800 trainable parameters.

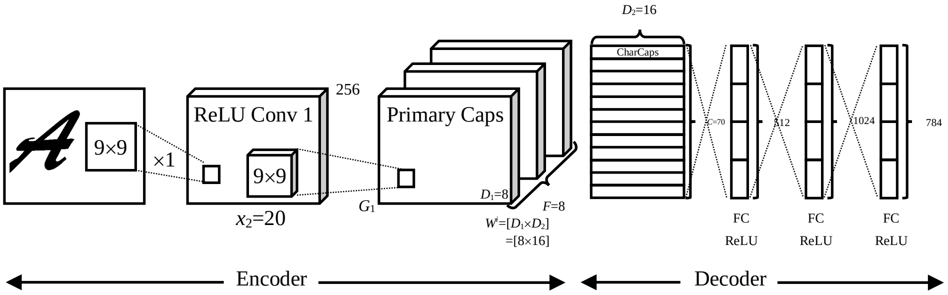


Fig.3. An overview of proposed framework: x_1 and x_2 depend on the size of the input image. G_1×G_2 are the dimensions of each primary capsule, and these values are computed automatically based on x_1 and x_2. D_1 and D_2 are the dimensions of the output vectors in primary and routing capsules, F is the number of channels in the primary capsule layer and C is the number of classes

- The first, second and third fully connected layer calculates the number of parameters based on bias which outputs a 512 vector,1024 vector, 784 vector respectively, and processing 537952 , 525312 , 803600 trainable parameters respectively. Thus, the total number of parameters in the capsule network is: 5835824.

The loss function is calculated for correct and incorrect DataCaps, primarily defined as 1 if the correct label corresponds with the character of this particular DataCap and 0 otherwise. A zero loss event is initiated either when a correct prediction occurs with probability greater than m^+ or when an incorrect prediction occurs with probability less than m^-. For each Datacaps capsule, k, the incurred loss is as follows:

$$L_k = T_k \max(0,m^+-||v_k||)^2 + \lambda(1-T_k)\max(0,||v_k||-m^-)^2 \qquad (3)$$

where T_k=1 if an image of class k is present and m^+ =0.9 and m^- =0.1, we use λ=0.5

The 8×16 transformation matrix W_{ij} maps the 8 dimensional capsule input space to a 16 dimensional capsule output space for each class j in relation to the capsule output of the previous layer u_i. The predicted vector $\hat{u}_{j|i}$ is expressed by a matrix operation between the weight matrix W_{ij} and u_i.

$$\hat{u}_{j|i} = W_{ij}u_i \qquad (4)$$

The final output v_j for class j is computed using novel vector-to-vector nonlinearity squashing function as:

$$v_j = \frac{\left\|S_j\right\|^2}{1+\left\|S_j\right\|^2}\frac{S_j}{\left\|S_j\right\|} \qquad (5)$$

where

$$S_j=\sum_i C_{ij}\hat{u}_{j|i} \qquad (6)$$

With C_{ij} coupling coefficients measuring the probability of primary capsule i probabilistically triggering capsule j. S_j representing the weighted sum shrinked by the squashing function.

4. EXPERIMENTS AND RESULTS

4.1 EXTRACTION SUBTITLE REGION

In this section, we discuss about the experimental analysis of proposed model for text detection. First of all, we discuss about the considered data set for experimental study. For the time being there is no uniform standard database for video subtitle, for this reason we have constructed a video database to test and evaluate the performance of our subtitle detection approach. The constructed dataset consists of more than 7,000 video frames of high and low resolution and high and poor quality. These frames are collected from various sources, including movies, cartoons, and documentaries. The video frames are different in the size, language, color, contrast ratio and font. They are also with text on simple background to text with complex background. We randomly sampled video frames and used them as test dataset.

For performance evaluation, we consider three matrices that are known as precision P, recall r and F-measure F, which is a single scalar that is the harmonic mean [3] [42]. Here, precision is defined as (number of extracted text areas/number of the whole extracted areas), recall (number of extracted text areas/number of the whole text areas) and f (2**precision***recall*/(*precision* + *recall*)). The proposed method in this work achieved a recall of 86.1%, a precision of 94.4% and F-measure of 88%. It also successfully extracts the subtitle regions, the problem with inconsistent contrast and color variations do not generate disturbance in localization subtitle regions. Images of any kind can be handled by our method regardless of resolution, contrast, blur and it gives promising results. Some true positive samples are shown in Fig.4.

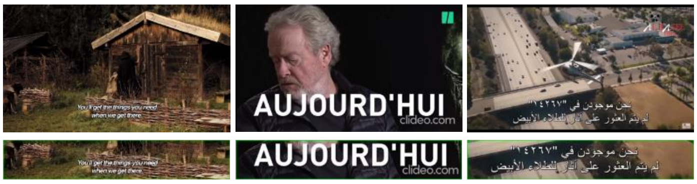

You'll get the things you need
when we get there
AUJOURD'HUI
clideo.com
نحن موجودن في "١٤٢٦٧"
لم يتم العثور على آثار للطلاء الأبيض
You'll get the things you need
when we get there
AUJOURD'HUI
clideo.com
نحن موجودن في "١٤٢٦٧"
لم يتم العثور على آثار للطلاء الأبيض

Fig.4. Examples of video subtitle detection results on various test sets. Detected text lines are bounded with green box

4.2 SUBTITLE RECOGNITION

In this section, we report the performance of our classifier based character recognition framework. We first introduce the datasets and then present the detailed recognition results.

The dataset used in this work is modified version of EnglishFnt dataset from Chars74K collection [43]. In this dataset, there are 70 classes consisting of all English alphabet letters in uppercase as well as lowercase, punctuation marks, along with the digits 0-9, where each class contains 700 images from different fonts. This dataset is considered to be challenging because an alphanumeric dataset that includes some labels is more prone to errors, e.g. classifying number zero to character “O” or vice versa.

The dataset is divided into training set and testing set, which contains 40000 and 9848 of images per class, respectively.

In this work, the number of primary capsules is set to 288. This is a small number compared to [9], which used 1,152 capsules for image classification. Our guess for this large difference is that the complexity of the produced feature map is lower in our case.

For training, we utilized the Adam optimizer [44] with exponentially decaying learning rates. We monotonically decreased the learning rate by decaying it by a factor of 0.99 in every epoch. We additionally utilized a batch size of 100, dropout rate of 0.5, batch size of 100 and the initial learning rate was initialized for all trainable parameters at 0.001. The loss parameter involves margin loss as computed for each capsule and reconstruction loss which is scaled down by 0.0005 to prevent domination.

Table.1. Obtained results with CapsNet and CNN model

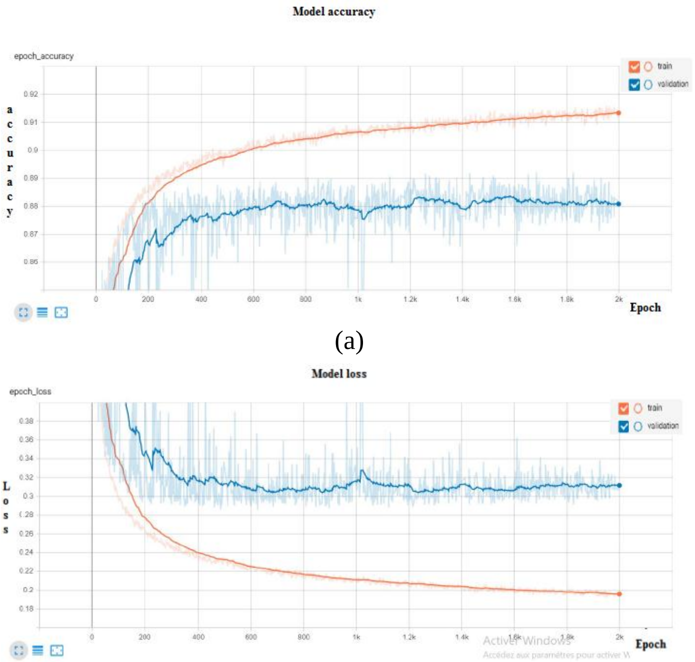
	Training Loss	Validation Loss	Training Accuracy	Validation Accuracy	Epoch
CapsNet	0.06	0.08	94.8%	92%	1700
CNN	0.2	0.31	91%	88%	2000

CNN was utilized as a baseline model for experimental comparisons. We carried out several tests in order to retain the best CNN architecture leading to the best classification rates. The CNN architecture consists of 6 convolution layers followed by a ReLU activation function. Max-pooling layer of size 2×2 is used between two consecutive convolution layers. Then, It is followed by flatten, dense, dropout layers ending with a Softmax activation layer with 70 outputs, which will indicate the classification percentages obtained for each class. The 1st and 2nd convolution layers apply 32 convolution kernels (3×3), the 3th and 4th layers apply 64 kernels (3×3), 5th and 6th apply 128 kernels (3×3). We use this optimization method with a momentum set to 0.9, a batch size of 100 and the base learning rate was initialized for all trainable parameters at 0.01.

Table.2. Accuracy, recall and F1-secore of character recognition (CapsNet)

Characters	**0**	**2**	**9**	**I**	**P**	**Y**	**x**	**y**	**?**
Accuracy	78%	99%	99%	89%	91%	92%	80%	96%	97%
Recall	83%	99%	98%	78%	91%	96%	83%	91%	100%
F1-score	81%	99%	98%	83%	91%	94%	81%	93%	99%

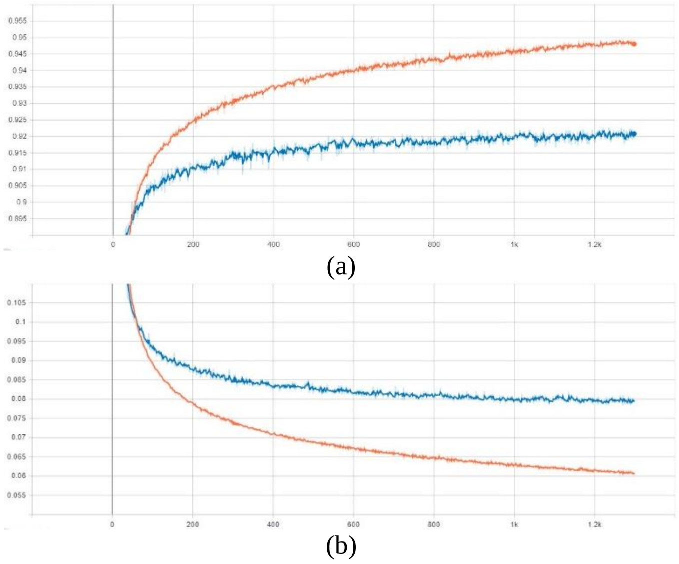
Our experimental results indicate that the CapsNet model when trained on Chars74K images achieves classification accuracy of 94.8%, which show significant improvement over the CNN model (91%), as shown in Table.1. Capsule network proved to be much more robust against complex data with higher number of classes. The improved performance boosts the application of capsule networks for the recognition of large variety of character with different font. We believe that this is due to the higher complexity of the second layer, which is a feature map utilizing convolutions. Therefore, the capacity to represent the information of the entities increases when training as a vector and it becomes possible to express various attributes of the entities. The Fig.4 and Fig.5 showing the accuracy and loss of training and validation set of Char47k dataset using CNN and CapsNet, respectively. The Table.2 show recognition accuracy, recall and F1-secore of some character from Char47k dataset.



(a)

(b)

Fig.5. CNN training graph. Accuracy (a) and loss (b) on train and validation set for Char74k



(a)

(b)

Fig.6. CapsNet training graph. Accuracy (a) and loss (b) on train and validation set for Char74k

5. CONCLUSION

In this paper, we propose an end-to-end subtitle extraction system specifically designed for videos. We introduce robust method which extracts the region where video subtitle is present. The proposed method is resistant to noise and is insensitive to backgrounds complexities, scale change and variation of font and languages. It capable of detecting accurate and complete text regions and there has been an appreciable increase in accuracy while recognizing the characters. This represents transference from scene text detection problem where advanced methods are designed to detect texts in a single image.

In this paper, we additionally investigated capsule networks with dynamic routing for video subtitle recognition. Capsule network is leveraged to diverse characters into tens of categories. The hierarchical structure and convolutional nature make it being able to extract robust high-level features. We compared the proposed model to CNNs, and demonstrated that capsule networks are indeed useful for character recognition based Chars74K datasets. Our model is trained on synthetic data, which give to our system the ability be retrained on other languages.

In future work, this system will be tested on videos in Chinese or other languages.

REFERENCES

[1] X. Chen and A.L. Yuille, "Detecting and Reading Text in Natural Scenes", *Proceedings of IEEE Computer Society Conférence on Computer Vision and Pattern Recognition*, pp. 1-12, 2004.

[2] C. Wolf and J.M. Jolion, "Extraction and Recognition of Artificial Text in Multimedia Documents", *Formal Pattern Analysis and Applications*, Vol. 6, pp. 309-326, 2004.

[3] S.M. Lucas, "Text locating Competition Results", *Proceedings of International Conference on Document Analysis and Recognition*, pp. 80-84, 2005.

[4] P. Viola and M. Jones, "Fast and Robust Classification using Asymmetric Adaboost and a Detector Cascade", *Advances in Neural Information Processing Systems*, Vol. 14, pp. 1311-1318, 2001.

[5] C. Yao, X. Bai, W. Liu, Y. Ma and Z. Tu, "Detecting Texts of Arbitrary orientations in Natural Images", *Proceedings of International Conference on Computer Vision and Pattern Recognition*, pp. 1083-1090, 2012.

[6] C. Yi and Y. Tian, "Text String Detection from Natural Scenes by Structure-Based Partition and Grouping", *IEEE Transactions on Image Processing*, Vol. 20, pp. 2594-2605, 2011.

[7] W. Huang, Z. Lin, J. Yang and J. Wang, "Text Localization in Natural Images using Stroke Feature Transform and Text Covariance Descriptors", *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1241-1248, 2013.

[8] D. Chen, H. Bourlard and J.P. Thiran, "Text Identification in Complex Background using SVM", *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1-14, 2001.

[9] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic Routing Between Capsules", *Advances in Neural Information Processing Systems*, Vol. 48, pp. 3856-3866, 2017.

[10] J. Su, D.V. Vargas and K. Sakurai, "One Pixel Attack for Fooling Deep Neural Networks", *IEEE Transactions on Evolutionary Computation*, Vol. 23, pp. 828-841, 2019.

[11] J. Su, D.V. Vargas and K. Sakurai, "Attacking Convolutional Neural Network using Differential Evolution", *IPSJ Transactions on Computer Vision and Applications*, Vol. 11, pp. 1-16, 2019.

[12] G.E. Hinton, A. Krizhevsky and S.D. Wang, "Transforming Auto-Encoders", *Proceedings of International Conference on Artificial Neural Networks*, pp. 44-51, 2011.

[13] X. Wang, L. Huang and C. Liu, "A New Block Partitioned Text Feature for Text Verification", *Proceedings of International Conference on Document Analysis and Recognition*, pp. 366-370, 2009.

[14] R. Minetto, N. Thome, M. Cord, N. J. Leite and J. Stolfi, "T-HOG: An Effective Gradient-Based Descriptor for Single Line Text Regions", *Pattern Recognition*, Vol. 46, pp. 1078-1090, 2013.

[15] X. Ren, K. Chen, X. Yang, Y. Zhou, J. He and J. Sun, "A New Unsupervised Convolutional Neural Network Model for Chinese Scene Text Detection", *Proceedings of International Conference on Signal and Information Processing*, pp. 428-432, 2015.

[16] W. Huang, Y. Qiao and X. Tang, "Robust Scene Text Detection with Convolution Neural Network Induced MSER Trees", *Proceedings of European Conference on Computer Vision*, pp. 497-511, 2014.

[17] A. Coates, B. Carpenter, C. Case, S. Satheesh, B. Suresh, T. Wang, D. J. Wu and A. Y. Ng, "Text Detection and Character Recognition in Scene Images with Unsupervised Feature Learning", *Proceedings of International Conference on Document Analysis and Recognition*, pp. 440-445, 2011.

[18] B. Shi, X. Bai and C. Yao, "An End-to-End Trainable Neural Network for Image-Based Sequence Recognition and Its Application to Scene Text Recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 39, pp. 2298-2304, 2016.

[19] B. Shi, X. Wang, P. Lyu, C. Yao and X. Bai, "Robust Scene Text Recognition with Automatic Rectification", *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 4168-4176, 2016

[20] M. Jaderberg, K. Simonyan, A. Vedaldi and A. Zisserman, "Synthetic Data and Artificial Neural Networks for Natural Scene Text Recognition", *Proceedings of International Conference on Computer Vision and Pattern Recognition*, pp. 1-7, 2014.

[21] R. Katebi, Y. Zhou, R. Chornock and R. Bunescu, "Galaxy Morphology Prediction using Capsule Networks", *Monthly Notices of the Royal Astronomical Society*, Vol. 486, pp. 1539-1547, 2019.

[22] A.D. Kumar, "Novel Deep Learning Model for Traffic Sign Detection using Capsule Networks", *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 432-443, 2018.

[23] M. Wang, J. Xie, Z. Tan, J. Su, D. Xiong and L. Li, "Towards Linear Time Neural Machine Translation with Capsule Networks", *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 551-557, 2018.

[24] B. Mandal, S. Dubey, S. Ghosh, R. Sarkhel and N. Das, "Handwritten Indic Character Recognition using Capsule Networks", *Proceedings of IEEE International Conference on Applied Signal Processing*, pp. 304-308, 2018.

[25] W. Zhao, J. Ye, M. Yang, Z. Lei, S. Zhang and Z. Zhao, "Investigating Capsule Networks with Dynamic Routing for Text Classification", *Proceedings of IEEE International*

Conference on Computer Vision and Pattern Recognition, pp. 881-889, 2018.

[26] J. Kim, S. Jang, E. Park and S. Choi, “Text Classification using Capsules”, *Neurocomputing*, Vol. 376, pp. 214-221, 2020.

[27] C. Xia, C. Zhang, X. Yan, Y. Chang and P. S. Yu, “Zero-Shot User Intent Detection via Capsule Neural Networks”, *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 661-673, 2018.

[28] C. Zhang, Y. Li, N. Du, W. Fan and P.S. Yu, “Joint Slot Filling and Intent Detection via Capsule Neural Networks”, *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 487-494, 2018.

[29] Y. Wang, A. Sun, J. Han, Y. Liu and X. Zhu, “Sentiment Analysis by Capsules”, *Proceedings of IEEE International Conference on World Wide Web*, pp. 1165-1174, 2018.

[30] H. Chao, L. Dong, Y. Liu and B. Lu, “Emotion Recognition from Multiband EEG Signals using CapsNet”, *Sensors*, Vol. 19, pp. 2212-2222, 2019.

[31] Y. Kim, P. Wang, Y. Zhu and L. Mihaylova, “A Capsule Network for Traffic Speed Prediction in Complex Road Networks”, *Proceedings of IEEE International Conference on Sensor Data Fusion: Trends, Solutions, Applications*, pp. 1-6, 2018.

[32] X. Ma, H. Zhong, Y. Li, J. Ma, Z. Cui and Y. Wang, “Forecasting Transportation Network Speed using Deep Capsule Networks with Nested LSTM Models”, *IEEE Transactions on Intelligent Transportation Systems (Early Access)*, pp. 1-12, 2020.

[33] M. Kim and S. Chi, “Detection of Centerline Crossing in Abnormal Driving using CapsNet”, *Journal of Supercomputing*, Vol. 75, pp. 189-196, 2019.

[34] T. Iqbal, Y. Xu, Q. Kong and W. Wang, “Capsule Routing for Sound Event Detection”, *Proceedings of IEEE International Conference on Signal Processing*, pp. 2255-2259, 2018.

[35] F. Vesperini, L. Gabrielli, E. Principi and S. Squartini, “Polyphonic Sound Event Detection by using Capsule Neural Networks”, *IEEE Journal of Selected Topics in Signal Processing*, Vol. 13, pp. 310-322, 2019.

[36] A. Pal, A. Chaturvedi, U. Garain, A. Chandra, R. Chatterjee and S. Senapati, “CapsDeMM: Capsule Network for Detection of Munro’s Microabscess in Skin Biopsy Images”, *Proceedings of IEEE International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 389-397, 2018.

[37] T. Iesmantas and R. Alzbutas, “Convolutional Capsule Network for Classification of Breast Cancer Histology Images”, *Proceedings of IEEE International Conference on Image Analysis and Recognition*, pp. 853-860, 2018.

[38] S. Prakash and G. Gu, “Simultaneous Localization and Mapping with Depth Prediction using Capsule networks for UAVS”, *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 191-198, 2018.

[39] L. Annabi and M. G. Ortiz, “State Representation Learning with Recurrent Capsule Networks”, *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 208-215, 2018.

[40] S. Garg, J. Alexander and T. Kothari, “Using Capsule Networks with Thermometer Encoding to Defend Against Adversarial Attacks”, Available at http://cs229.stanford.edu/proj2017/final-reports/5244416.pdf, Accessed at 2017.

[41] K. Duarte, Y. Rawat and M. Shah, “Videocapsulenet: A Simplified Network for Action Detection”, *Advances in Neural Information Processing Systems*, pp. 7610-7619, 2018.

[42] S.M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong and R. Young, “Robust Reading Competitions”, *Proceedings of International Conference on Document Analysis and Recognition*, pp. 682-687, 2003.

[43] T.E. De Campos, B.R. Babu and M. Varma, “Character Recognition in Natural Images”, *Proceedings of International Conference on Image and Video Formation, Preprocessing and Analysis*, pp. 1-8, 2009.

[44] D.P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization”, *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 981-989, 2014.