OCR On-the-Go: Robust End-to-end Systems for Reading License Plates & Street Signs

Rohit Saluja* Ayush Maheshwari* Ganesh Ramakrishnan* Parag Chaudhuri* Mark Carman†
IITB-Monash Research Academy *IIT Bombay* *IIT Bombay* *IIT Bombay* *Politecnico di Milano*
Mumbai, India Mumbai, India Mumbai, India Mumbai, India Milan, Italy

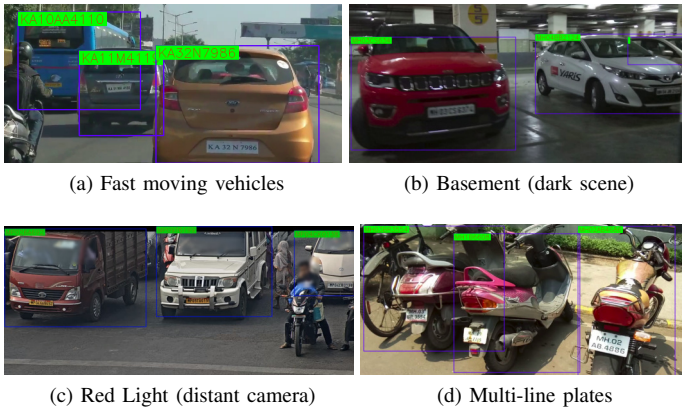
*{rohitsaluja, ayusham, ganesh, paragc}@cse.iitb.ac.in, †mark.carman@polimi.it

***Abstract*—We work on the problem of recognizing license plates and street signs automatically in challenging conditions such as chaotic traffic. We leverage state-of-the-art text spotters to generate a large amount of noisy labeled training data. The data is filtered using a pattern derived from domain knowledge. We augment the training and testing data with interpolated boxes and annotations that make our training and testing robust. We further use synthetic data during training to increase the coverage of the training data. We train two different models for recognition. Our baseline is a conventional Convolution Neural Network (CNN) encoder followed by a Recurrent Neural Network (RNN) decoder. As our first contribution, we bypass the detection phase by augmenting the baseline with an Attention mechanism in the RNN decoder. Next, we build in the capability of training the model end-to-end on scenes containing license plates by incorporating an inception based CNN encoder that makes the model robust to multiple scales. We achieve improvements of 3.75% at the sequence level, over the baseline model. We present the first results of using multi-headed attention models on text recognition in images and illustrate the advantages of using multiple heads over a single head. We observe gains as large as 7.18% from incorporating multi-headed attention. We also experiment with multi-headed attention models on French Street Name Signs dataset (FSNS) and a new Indian Street dataset that we release for experiments. We observe that such models with multiple attention masks perform better than the model with single-headed attention on three different datasets with varying complexities. Our models outperform state-of-the-art methods on FSNS and IIIT-ILST Devanagari datasets by 1.1% and 8.19% respectively.**

I. Introduction

Text spotting or optical character recognition (OCR) in scenes has many applications such as helping the visually impaired, helping travellers translate texts on signboards and also robotics. Reading the scenes end-to-end has an advantage of utilizing the global context in street boards or multi-line license plates, which enhances the learning of patterns. One of the important factors that separates a character level OCR system from an end-to-end OCR system is reading order. Attention is thus needed to i) locate the initial characters, read them and ii) keep the track of the correct reading order in form of change in characters, words, lines, paragraphs or columns (in multi-column texts). This observation forms the motivation of our work.

There has been a rising interest in end-to-end scene text spotting in images over the last decade [1]–[5]. Spotting text in scene images is typically performed in two steps, *viz.*, i) text localization/detection and ii) text recognition. Work specific to the localization task [6], has also been extended to real-time text detection [7], [8]. For the recognition task, CNNs are used for feature extraction, followed by RNNs for sequence classification [2]. Research in scene text spotting has seen improved solutions in terms of accuracy and speed [4], [5], but the problem of text spotting in the wild is complicated by variations in illumination and weather conditions. State-of-the-art recall, precision and F-measure scores on the COCO-Text dataset [9] are as low as 28.33, 68.42 and 40.07 respectively.



(a) Fast moving vehicles (b) Basement (dark scene)

(c) Red Light (distant camera) (d) Multi-line plates

Fig. 1. Sample chaotic scenes with predictions of our model (top left of each box). License plates at varying scales and orientations/locations (in the camera window) motivate the use of inception-based CNN and attention.

Determining the correct reading order over multiple text segments occurring in the same scene is another important problem that has received relatively little attention in the scene text literature. With the success of end-to-end models that can be trained without any supervision at the level of individual text-boxes [1], [10], a natural next step is to investigate if this success can be extended to determining the correct reading order. Thus we analyze results for determining the natural reading order over scenes with varying complexities. Our model, described in Section III-B, is able to successfully jump from one part of multi-line license plates (refer Figure 4 bottom) to another in the correct reading order.

A. Related Work

The particular problem of spotting license plates in scenes is useful in surveillance, toll collection, parking systems,

Type of Dataset	Duration/Quantity	Variations	Max seq. len.	μ, σ (I_{avg})
Day time traffic video Night time traffic video Night time traffic video with rain	36 hr 30 mins 15 hr 56 mins 2 hr 28 mins	motion blur, chaotic, illumination, single/dual line	10 (per plate)	86.45, 12.94
FSNS Dataset	966k train, 39k validation, & 43k test data-points	motion blur, multi-line, multiple views	37 (per board)	95.71, 26.01
Indic Street Dataset	1000 videos each covering a board from multiple orientations	motion blur, varying backgrounds, multi-lingual & multi-line	180 (per board)	111.64, 32.55

TABLE I
DATASETS USED FOR OUR EXPERIMENTS. I_{avg} STANDS FOR AVERAGE IMAGE INTENSITY.

Fig. 2. Top: FSNS sample. Bottom: Indic Street Board samples.

developing smart cities, *etc*. The problem involves several challenges, such as the ability to handle a diversity of fonts and patterns, robustness to geometrical transformations, composite backgrounds and camera positions as shown in Figure 1. Moreover in videos, motion blur and varying illumination can make it harder to recognize the numbers correctly. Features such as edges, colors, character images, texture, and the geometric properties of license plates are often used for detection [11]–[13]. To segment out the character images (for further recognition) in licence plates, methods based on connectivity, projections, geometrical rules, contours and/or template matching have been employed in the past [14]–[17]. Such methods, however, suffer from segmentation errors. There have been efforts to develop robust methods based on learning, to overcome such errors [18]. For example, sliding window based CNN classifiers, along with RNNs, have been used for recognizing characters in recent work on reading license plates [12]. However, methods based on sliding windows are computationally expensive since they require the rejection of a large fraction of non-character images.

Recent work on license plate recognition in a chaotic scenario [11] uses edge-based features and other geometrical properties to detect license plates in the scene. The false positive detections are then rejected by a CNN-based binary classifier. Further, a CNN-based character level recognizer, with a spatial transformer layer to overcome variations in brightness and tilt, is also employed.

Obtaining a large amount of labeled training data is an expensive and human-intensive process. Many state-of-the-art systems obtain large-scale noisy labels by combining a large amount of unlabeled data together with a modest amount of labeled data, (for example in image search for landmarks [19]). Training with noisy labels is extensively studied in the literature [20]. In the present work, we use a state-of-the-art text spotter to obtain noisy labels for a large number of video frames, and then clean the data using a pattern grammar. We augment the training data with synthetic data using SynthText [6], thus increasing the coverage of training. We then train a CNN-RNN model for recognition. Our end-to-end model utilizes an inception-based CNN followed by an attention-based RNN. Our key contributions are as follows: (1) A new methodology for labelling a large amount of training data, (refer Section I), that allows us to work with over 1000k real scene images containing license plates. (2) We present the first results of multi-head attention models on the task of text recognition. (3) We release a new multi-lingual data-set of 1000 videos (with text in Hindi, Marathi and English) each covering an Indic street board from different orientations. (4) To obtain synthetic data for License Plates and Indic scene text, we include License Plate patterns and Devanagari (and Latin) sequences in SynthText [6].

The multi-lingual data-set and supplementary material for our paper can be accessed from https://www.cse.iitb.ac.in/~rohitsaluja/project. The source code of our work is available at https://github.com/rohitsaluja22/OCR-On-the-go.

II. DATA DESCRIPTION, PROBLEM SCOPE & ANALYSIS

We use datasets of varying complexities as shown in Table I, working with various illumination and weather conditions for the problem of license plate recognition in chaotic scenes. The conditions and duration of the videos in our dataset are described in Table I (top). Our work can be applied to any scene text problem with labels that follow a particular pattern or grammar. We also perform our experiments with multi-headed attention mechanism on French Street Name Signs (FSNS) dataset and a new Indic Street Board dataset. For FSNS, each data-point consists of four images of a street board and a corresponding label. In Table I, we provide the quantitative description of the FSNS Dataset and a new Indic Street Board dataset that we have released. For Indic street boards, each of our videos covers a single street board from different orientations. Samples from these datasets are shown in Figure 2. We find that in addition to higher sequence length, the street-board images also have a higher mean and standard deviation of average image intensity I_{img}, as compared to the license plate images.

To generate labels for the dataset, we first obtain noisy predictions on every frame of each video (refer to Table I) using the state-of-the-art text spotter, *viz.*, DeepTextSpotter [5]. We then filter out instances in the dataset that do not follow the license plate grammar. For analysis, we calculate the accuracy (at the word-level) of the noisy predictions on an annotated set of 1000 samples. We observe that 73% of predictions that follow the license plate grammar are indeed correct. For data

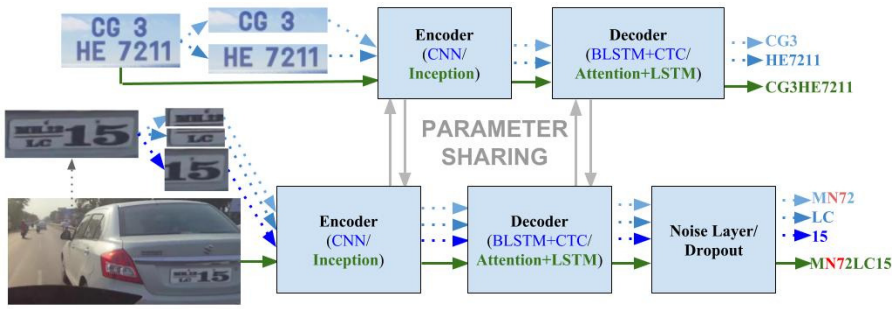


Fig. 3. We train our models on synthetic clean data (top) & real noisy labeled data (bottom). The license plate image flows in multiple splits through the *baseline model*, shown by dotted blue arrows. For the *end-to-end model*, the entire image flows through the model together, shown by green arrows.

augmentation, we further apply the filtered predictions across the video to correct other frames in the video as follows:

- If two successive (but not consecutive) filtered frames, f_i and f_k (where $i < k-1$), have identical predictions: $p_i = p_k$;
- and there exist intermediate frames $\{f_j\}$ $(i < j < k)$ in the original video, for which no prediction fits the grammar (or the DeepTextSpotter makes no prediction)
- then we assign the prediction $p_j = p_i (= p_k)$ to all the intermediate frames $\{f_j\}$.
- To obtain text-boxes for the intermediate frames $\{f_j\}$, we apply linear interpolation on text-boxes from the previous frame f_i and from the subsequent frame f_k.

III. System Architecture

As a baseline, we will use a conventional CNN-based encoder and RNN-CTC based decoder for recognizing cropped license plate images. However, as we will discuss further in Section III-B, the problem of multi-scale variation can be handled by using the inception-based CNN as encoder. Moreover, the challenge of reading the characters at different locations in a scene can be handled by the attention based RNN decoder, thus enabling end-to-end recognition in the scene text images.

A. Baseline model

As a baseline, we use the seven layer convolutional neural network (CNN) to extract the features from license plate images, followed by a two-layer bi-directional long short-term memory (BLSTM) for decoding the features, and a connectionist temporal classification (CTC) layer for aligning the decoded predictions. We use the TensorFlow implementation described in [2]. As shown in Figure 3 (follow dotted blue arrows), the model is trained on: (1) synthetic clean data: $\{X_s,Y_s\}$, with X_s being the synthetic license plate image (or sub-plate image in the case of a multi-line license plate), and Y_s being the corresponding clean labels, (2) real noisy data :$\{X_r,Y_r\}$, with X_r being the real license plate image (or sub-plate image) and Y_r being the corresponding noisy labels produced using the state-of-the-art DeepTextSpotter. Inspired by the literature on training neural networks with noisy data [20], our model shares parameters between the networks that are trained on the clean and noisy datasets respectively. We argue, however, that our case is different from previous work, owing to three distinctive properties:

1) Firstly, the set of synthetic input images $\{X_s\}$ are visually distinct from the set of real input images $\{X_r\}$, and are not as useful as the real images for the test data. We observed in our experiments that if in every epoch, training is consecutively performed on the two datasets, the model tends to overfit to the dataset that is used first. We therefore randomly shuffle the order in which the two datasets are used in each epoch.

2) Secondly, a dropout ("keep probability" = 0.5) is applied to the last stage of the decoder which acts to prevent overfitting even in the case of noisy labels. We argue that additional noise correction layer (as advocated by Hedderich and Klakow [20]) is not needed in this case as overfitting to noisy characters can simply be avoided by applying sufficient regularization through dropout [21].

3) Moreover, we observe in the literature [20] that the noise layer significantly helps in the case of less clean training data, and the results are not significantly improved by the addition of large amounts of clean data. Since, we train with a large amount of clean as well as noisy labeled real data, we avoid using the noise correction layer.

B. End to end model for scene text recognition

Our end-to-end model is developed over the tensorflow implementation of attention_ocr [22]. It is important to note that the vehicles appear at different scales in the scene as shown in Figure 1 (a,b,d). Thus, a powerful encoder is needed to capture the multi-scale variation across the license plates. Furthermore, as shown in Figure 1 (c), similarly scaled license plates are positioned at varying locations in the scene. Moreover, license plates exist at varying orientations in the scenes (Figure 1 (b,d)). Thus attention-based models are important to locate the character images in the scene. Our model has the following components (refer Figure 4):

1) As a powerful encoder, the inception based CNN learns to extract the features f from the input image [23]. One of the important parts of inception based network is that it has varying sized convolution layers in parallel, which helps in learning the text images at different resolutions.

2) With attention based LSTM as the powerful decoder, attention is learned over i) the features from one of the middle layer of the inception based network, ii) one-hot-encoded (OHE) vectors e_x and e_y for both x and y coordinates of

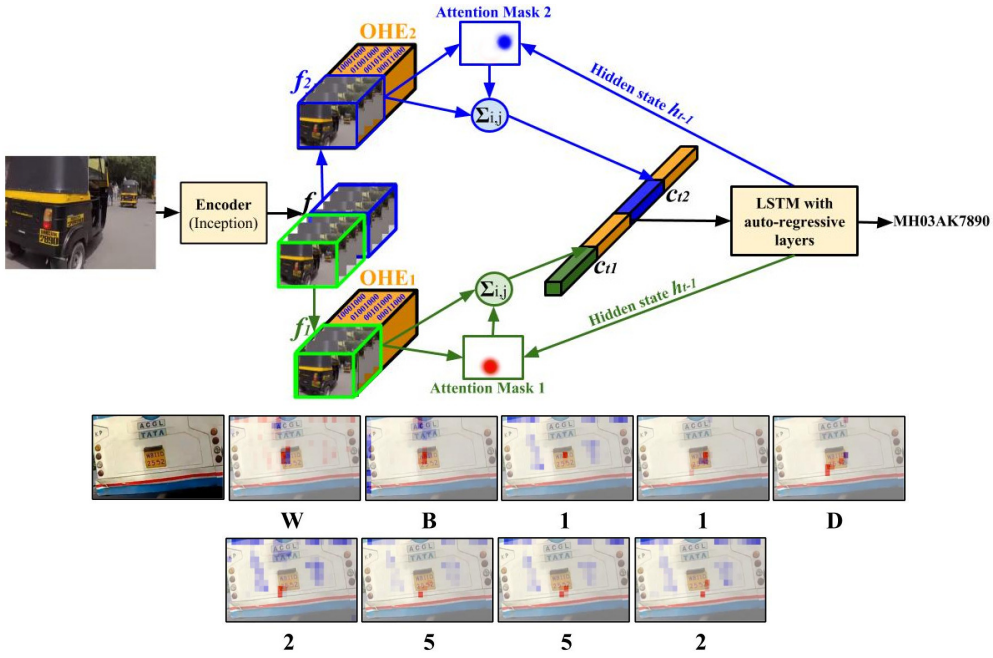


Fig. 4. Top: Two-headed split-attention based model. Bottom: Attention masks, note that the two masks (shown in red and blue) have unique coverage.

the features, and iii) hidden state of the LSTM at the previous time step of decoding. The OHE vectors for the coordinates provide location awareness to the network [10], thus making it possible to jump from the upper right character of the multi-line plate to the lower left character (refer Figures 1 d and 4).

3) An LSTM layer takes the context vector from the attention layer as well as the previous OHE output of the decoded sequence thus learning the language model (or license plate grammar in our case) via auto-regression.

We also experiment with a multi-head attention mechanism, which consists of several attention layers running in parallel [24]. We split the encoded features f into two or more parts and learn separate attention masks over each of them. Moreover, we keep each attention mask location aware by appending their respective features with one hot encoded vectors for both x and y coordinates of the feature map. The splitting of features reduces the computational overhead of the multi-head attention approach, and also allows the different "attention heads" to learn different information from the respective features. It also leads to an ensemble effect. Finally, context vectors obtained by applying the attention masks to corresponding splits (concatenated with OHE vectors), are concatenated together to form the input to the LSTM. Figure 4 provides an illustration of such a model for two-headed attention. As shown, the features f are split into two parts f_1 and f_2 (shown in green and blue colours). The context vectors c_{t1} and c_{t2} are finally concatenated to form the input of the LSTM layer. For reasons explained in Section III-A, we train by randomly switching between the models for clean synthetic data and noisy labeled data, and incorporate the dropout (with keep probability = 0.5) in the last layer of LSTM to avoid overfitting to noisy characters. The complete pipeline is shown in Figure 3.

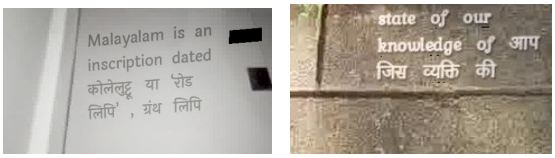


Fig. 5. Sample synthetic scenes with Devanagari & Latin scripts.

IV. Experiments

We experiment on three different datasets with varying complexities. Firstly, we work on license plate recognition in chaotic Indic scenes with noisy labelled real data as well as clean synthetic data. To obtain the real data with noisy labels we use DeepTextSpotter [5]. The DeepTextSpotter model is trained on the SynthText dataset [6], as well as the ICDAR datasets [3], [4]. Since it is not trained on license plate images, the overall performance is not satisfactory, whereas we obtain 73% word-level accuracy on the data that follows the license plate grammar. We work on 480x260 images for end-to-end experiments and all the license plate/sub-plate images are resized, with bilinear interpolation, to 32x100 images for input to the baseline model. To obtain clean data, we synthesize a large number of scene images with text from Indian license plates using SynthText [6]. For each license plate, we select a random 280x460 crop around it (covering the other license plate images with black pixels if they exist in the crop). For baseline, we obtain the license plate images similar to the one shown in Figure 3 (top-left). To obtain all the synthetic images for our experiments we use 18 freely available license plate fonts [25]. Using the method described in Section II, we obtain 1063k frames with license plates and corresponding predictions for training using 55 hours of video data. We train our models on 1063k frames with 64:16:20 train:val:test split. Additionally, for generalization across various Indian states,

Training Method	Character Accuracy with inception-v3 encoder	Sequence Accuracy with inception-v3 encoder	Character Accuracy with inc.-resnet-v2 encoder	Sequence Accuracy with inc.-resnet-v2 encoder
Baseline CNN-RNN model	96.74%	86.12%	96.74%	86.12%
E2E model with 1 head attention	97.48 %	89.87%	97.94 %	91.28%
E2E model with 2 head attention	**97.62%**	**91.06%**	98.06 %	91.56%
E2E model with 4 head attention	88.40 %	69.43%	98.12%	92.05%
E2E model with 8 head attention	-	-	**98.43%**	**93.30%**

TABLE II

EVALUATION ON LICENSE PLATE SCENES. E2E STANDS FOR END-TO-END. THE DATASET IS COMPLEX DUE TO 1) RECOGNITION BEING PERFORMED ON CONTINUOUS VIDEO FRAMES, 2) PRESENCE OF MOTION BLUR IN THE FRAMES WHERE WE INTERPOLATED THE ANNOTATIONS.

we use 187k synthetic scenes (only while training).

We avoid the use of synthetic dataset and dropout layer while training our models on FSNS dataset since it is large in quantity and contain clean annotated labels (as compared to our noisy-labelled license plate dataset). Moreover, each training and testing sample from FSNS dataset reuses encoder four times, once for each view. We further perform our experiments with mixed-6a layer from the inception-resnet-v2 as an encoder for all our datasets. Using this encoder we obtain the features of size 14x28x1088 for license plate images and Indic Street Board images, and 7x7x1088 for each view in FSNS images. For Indic Street Boards, we obtain around 79k frames, each of size 280x460, from the videos described in Table I. We use initial 50k frames for training, next 12k for validation and remaining 17k for testing. We also augment our training dataset with 700k synthetic scenes obtained from SynthText (modified to inlcude large multi-script sequences, refer Figure 5) with around 50 unicode fonts [26]. Additional experimental details, including hyper-parameters, are given in the supplementary material.

V. EVALUATION

In this section, we present the results of our experiments.

A. Visualization of Attention Masks

The predictions of our model for some of the complex test cases are shown in Figure 1. In, Figure 4 (bottom), we present results visualizing the multi-head attention masks (resized to image size with nearest neighbour interpolation) for text recognition. (Single-head results available in the supplementary material.) It is important to note that, specifically for license plate scenes, one of the attention masks moves over each and every character due to randomness of the chosen character at each position on the plate. In contrast, we observed that for other data sets that the attention mask often remains idle (on the edges of the image) after reading the initial few characters of highly frequent words in the language (refer attention masks in the literature [10]). This is probably due to 1) the implicit language model, 2) large receptive fields around each feature location, or 3) the mask not finding the character (due to occlusion, missing view) in the image and therefore remaining idle. We observed that the 47.26% of attention weights are focused on the edge of 100 sample images of FSNS dataset, whereas the fraction goes down to 19.69% (mainly due to borders in some of the videos) for the same number of license plate scenes. As shown in Figure 4 (bottom), both the masks (shown in red and blue) have unique coverage. Moreover, it can be observed that the first attention mask (in red) moves from the first character to the last character in the correct reading order (top-to-bottom followed by left-to-right) for the multi-line license plates. The second mask (in blue) probably explores the new lines, non-pattern text and the background regions. Moreover, the blue mask is highly scattered at locations where it could not find the license plate patterns. We observe that the attention masks for 4 (and more) headed attention models are more scattered as compared to 2 headed attention models. Though scattered, coverage of each mask is unique irrespective of the number of heads used in the models.

B. Evaluation on continuous License Plate Video Scenes

1) Effect on increasing number of heads: The results for our experiments with license plate scenes, are shown in Table II. As shown, the baseline model achieves the character level accuracy of 96.74% and sequence accuracy of 86.12%. With inception-v3 encoder, for the model with single head attention we achieve a gain in sequence level accuracy of 3.75%. The accuracy gains further increases to 4.94% with the two-headed attention model. Splitting the features further for four-headed attention decreases the performance with inception-v3 encoder. This happens probably because it becomes difficult to learn the proper attention masks with a lesser number of features (288/4 = 72). This also indicates that improvement gains for the model with two-headed attention are not due to an increase in the number of parameters of the model. The *residual* networks have been shown to improve performance in image classification tasks in literature [27]. The problem of decrease in accuracy with increase in heads is resolved by using better encoder with denser layer to derive features i.e. the mixed-6a layer (with feature depth = 1088) of inception-resnet-v2 network. Thus, with this encoder, we can observe the gain of 5.16% using single headed attention w.r.t. baseline, and the gain further increases to 7.18% using eight headed attention (no. of features with each attention being 136).

2) Effect on increasing inception parameters: As shown in Table II, both character level accuracy as well as sequence accuracy increases with the inception-resnet-v2 encoder as compared to inception-v3 encoder.

C. Evaluation on FSNS dataset and Indic Street Boards

We further experiment with multi-headed attention models on the French Street Name Signs (FSNS) dataset. We use the mixed-6a layer of the inception-resnet-v2 network as encoder for the reasons mentioned in previous subsection. The performance on the FSNS dataset is shown in Table III. As shown, the models with multi-headed attention masks perform

Training Method	Seq. Acc. on FSNS dataset	Seq. Acc. on IIIT-ILST
Baseline Model	Smith et al. [28]	CNN-RNN
Baseline Results	72.46%	42.90%#
E2E model w/t 1 head attention	84.20%*	46.27%
E2E model w/t 2 head attention	84.59%	47.63%
E2E model w/t 4 head attention	84.86%	50.36%
E2E model w/t 8 head attention	**85.30%**	**51.09%**

TABLE III
EVALUATION ON FSNS DATASET., *STATE-OF-THE-ART [10], AND IIIT-ILST DEVANAGARI DATASET, #STATE-OF-THE-ART [29].

better than the model with single-headed attention. Also, it is important to note that the performance improves with an increase in the number of heads from two to eight and our models outperform state-of-the-art results [10].

We trained our model on 750k Indic Street Board scenes described in Section IV. On standard IIIT-ILST Devanagari dataset of 1.1k images, each containing a word, we obtain the results shown in Table III. As shown, we outperform the state-of-the-art results for the dataset [29]. Moreover, the models with multiple masks perform better than the models with a comparatively lesser number of masks. Our preliminary results on the end-to-end test set of 17k frames from the multilingual dataset are just 35% accurate on character level and 13% accurate on the sequence level. This happens because the task of end-to-end recognition in Indic Street Board scenes is extremely challenging due to the presence of handwritten characters, two scripts (Devanagari and Latin) or three languages (Hindi, Marathi and English), and larger sequence length (180 w.r.t. 35 in FSNS dataset) in the Indic Street Boards. We are hopeful that by improving the techniques to train attention based models on large sequential multilingual data will improve the results further. Reducing the sequence length by using script grammar may also be an interesting area for future work.

More sample results and ablation studies (related to the performance on images with varying intensities) of our work are given in the supplementary material.

VI. CONCLUSION

We present an end-to-end trainable framework for reading text from scenes and illustrated its application in two scenarios: (i) recognizing license plates automatically in chaotic traffic conditions, a task for which we curated our own dataset and (ii) the existing publicly available FSNS and IIIT-ILST Devanagari datasets. We perform our experiments for license plate recognition on a large number of video frames. A salient point of our framework is that our models, when trained only on a combination of noisy labelled data and clean synthetic data and when appropriately tuned, set new benchmarks for the task. Moreover, we are the first to observe that multi-headed attention is more effective in reading scene text than the single headed attention.

REFERENCES

[1] C. Bartz, H. Yang, and C. Meinel, "Stn-ocr: A single neural network for text detection and text recognition," *arXiv:1707.08831*, 2017.

[2] 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 TPAMI*, vol. 39, no. 11, 2017.

[3] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. i Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazan, and L. P. De Las Heras, "Icdar 2013 robust reading competition," in *Proceedings of ICDAR*, 2013.

[4] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu *et al.*, "Icdar 2015 competition on robust reading," in *Proceedings of ICDAR*, 2015, pp. 1156–1160.

[5] M. Bušta, L. Neumann, and J. Matas, "Deep textspotter: An end-to-end trainable scene text localization and recognition framework," *Proceedings of ICCV*, 2017.

[6] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localisation in natural images," in *Proceedings of CVPR*, 2016.

[7] M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu, "Textboxes: A fast text detector with a single deep neural network." in *Proceedings of AAAI*, 2017.

[8] B. S. Minghui Liao and X. Bai, "TextBoxes++: A single-shot oriented scene text detector," *CoRR*, vol. abs/1801.02765, 2018.

[9] A. Veit, T. Matera, L. Neumann, J. Matas, and S. Belongie, "Coco-text: Dataset and benchmark for text detection and recognition in natural images," *arXiv:1601.07140*, 2016.

[10] Z. Wojna, A. N. Gorban, D.-S. Lee, K. Murphy, Q. Yu, Y. Li, and J. Ibarz, "Attention-based extraction of structured information from street view imagery," in *Proceedings of ICDAR*, vol. 1, 2017.

[11] V. Jain, Z. Sasindran, A. Rajagopal, S. Biswas, H. S. Bharadwaj, and K. R. Ramakrishnan, "Deep automatic license plate recognition system," in *Proceedings of ICVGIP*, 2016.

[12] H. Li and C. Shen, "Reading car license plates using deep convolutional neural networks and lstms," *arXiv:1601.05610*, 2016.

[13] S. Du, M. Ibrahim, M. Shehata, and W. Badawy, "Automatic license plate recognition (alpr): A state-of-the-art review," *IEEE Transactions on circuits and systems for video technology*, vol. 23, no. 2, 2013.

[14] Y. Yoon, K.-D. Ban, H. Yoon, and J. Kim, "Blob extraction based character segmentation method for automatic license plate recognition system," in *IEEE Intl. Conf. on Systems, Man, and Cybernetics*, 2011.

[15] S. Nomura, K. Yamanaka, O. Katai, H. Kawakami, and T. Shiose, "A novel adaptive morphological approach for degraded character image segmentation," *Pattern Recognition*, vol. 38, no. 11, 2005.

[16] Y. Zhang, Z. Q. Zha, and L. F. Bai, "A license plate character segmentation method based on character contour and template matching," in *Applied Mechanics and Materials*, vol. 333. Trans Tech Publ, 2013.

[17] S. Rasheed, A. Naeem, and O. Ishaq, "Automated number plate recognition using hough lines and template matching," in *Proceedings of the World Congress on Engineering and Computer Science*, vol. 1, 2012.

[18] J. Jiao, Q. Ye, and Q. Huang, "A configurable method for multi-style license plate recognition," *Pattern Recognition*, vol. 42, no. 3, 2009.

[19] L. S. Kennedy and M. Naaman, "Generating diverse and representative image search results for landmarks," in *Proceedings of WWW*, 2008.

[20] M. A. Hedderich and D. Klakow, "Training a neural network in a low-resource setting on automatically annotated noisy data," *Proceedings of ACL*, 2018.

[21] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, 2014.

[22] Tensorflow, "Attention Ocr Model," https://bit.ly/2BczGN3. Last accessed on March 7, 2019.

[23] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of CVPR*, 2016.

[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017.

[25] Fontspace, "18 Free LP Fonts," https://fontspace.com/category/license\%20plate. Last accessed on March 7, 2019.

[26] Devanagari, "50 Fonts," http://indiatyping.com/index.php/download/top-50-hindi-unicode-fonts-free. Last accessed on March 7, 2019.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of CVPR*, 2016.

[28] R. Smith, C. Gu, D.-S. Lee, H. Hu, R. Unnikrishnan, J. Ibarz, S. Arnoud, and S. Lin, "End-to-end interpretation of the french street name signs dataset," in *ECCV*, 2016.

[29] M. Mathew, M. Jain, and C. Jawahar, "Benchmarking scene text recognition in devanagari, telugu and malayalam," in *Proceedings of ICDAR*, vol. 7. IEEE, 2017.