CATALIST: CAmera TrAnsformations for multi-LIngual Scene Text recognition

Shivam Sood[1] (✉)[0000−0002−3543−1754], Rohit Saluja[2][0000−0002−0773−3480], Ganesh Ramakrishnan[1][0000−0003−4533−2490], and Parag Chaudhuri[1][0000−0002−1706−5703]

https://catalist-2021.github.io
IIT Bombay, India
{ssood,ganesh,paragc}@cse.iitb.ac.in
IIIT Hyderabad, India
rohit.saluja@research.iiit.ac.in

Abstract. We present a CATALIST model that 'tames' the attention (heads) of an attention-based scene text recognition model. We provide supervision to the attention masks at multiple levels, *i.e.*, line, word, and character levels while training the multi-head attention model. We demonstrate that such supervision improves training performance and testing accuracy. To train CATALIST and its attention masks, we also present a synthetic data generator ALCHEMIST that enables the synthetic creation of large scene-text video datasets, along with mask information at character, word, and line levels. We release a real scene-text dataset of $2k$ videos, CATALIST_d with videos of real scenes that potentially contain scene-text in a combination of three different languages, namely, English, Hindi, and Marathi. We record these videos using 5 types of camera transformations - (i) *translation*, (ii) *roll*, (iii) *tilt*, (iv) *pan*, and (v) *zoom* to create transformed videos. The dataset and other useful resources are available as a documented public repository for use by the community.

Keywords: Scene Text Recognition · Video Dataset · OCR in the Wild · Multilingual OCR · Indic OCR · Video OCR

1 Introduction

Reading the text in modern street signs generally involves detecting the boxes around each word in the street signs and then recognizing the text in each box. Reading street signs is challenging because they often appear in various languages, scripts, font styles, and orientations. Reading the end-to-end text in scenes has the advantage of utilizing the global context in street signs, enhancing the learning of patterns. One crucial factor that separates a character-level OCR system from an end-to-end OCR system is reading order. Attention is thus needed to locate the initial characters, read them, and track the correct reading order in the form of change in characters, words, lines, paragraphs, or columns (in multi-column texts). This observation forms the motivation for our work.

Fig. 1: Sample video frames from $\text{CATALIST}_{\text{d}}$

Obtaining large-scale multi-frame video annotations is a challenging problem due to unreliable OCR systems and expensive human efforts. The predictions obtained on videos by most OCR systems are fluctuating, as we motivate in Section 3. The fluctuations in the accuracy of the extracted text may also be due to various external factors such as partial occlusions, motion blur, complex font types, distant text in the videos. Thus, such OCR outputs are not reliable for downstream applications such as surveillance, traffic law enforcement, and cross-border security system.

In this paper, we demonstrate that the photo OCR systems can improve by guiding the attention masks based on the orientations and positions of the camera. We improve an end-to-end attention-based photo-OCR model on continuous video frames by taming the attention masks in synthetic videos and on novel controlled datasets that we record for capturing possible camera movements.

We begin by motivating our work in Section 3. We base a video scene-text recognition model (referred to as CATALIST) on partly supervised attention. Like a teacher holding a lens through which a student can learn to read on a board, CATALIST exploits supervision for attention masks at multiple levels (as shown in Figure 3). Some of the attention masks might be interpreted as covering different orientations in frames during individual camera movements (through separate masks). In contrast, others might focus on the line, word, or character level reading order. We train CATALIST using synthetic data generated using a non-trivial extension of SynthText [6]. The extension allows for the generation of text videos using different camera movements while also preserving character-level information. We describe the CATALIST model which 'tames' the attention (heads) in Section 4.1. We demonstrate that providing direct supervision to attention masks at multiple levels (*i.e.*, line, word, and character levels) yields improvement in the recognition accuracy.

To train CATALIST and its attention masks, we present a synthetic data generator ALCHEMIST[1] that enables the synthetic creation of large scene-text

[1] ALCHEMIST stands for synthetic video generation in order to tame **A**ttention for **L**anguage (line, word, character, *etc.*) and other camera-**CH**ang**E**s and co**M**binat**I**ons for **S**cene **T**ext.

video datasets, along with mask information at character, word and line levels. We describe the procedure to generate synthetic videos in Section 4.2.

We also present a new video-based real scene-text dataset, $\text{CATALIST}_{\text{d}}$ in Section 4.3. Figure 1 shows the sample video frames of the dataset. We create these videos using 5 types of camera transformations - (i) *translation*, (ii) *roll*, (iii) *tilt*, (iv) *pan*, and (v) *zoom*. We provide the dataset and experimental details in Section 5. We summarize the results in Section 6 and conclude the work in Section 8.

2 Related Work

We now introduce the approaches to tackle various issues in the field of photo OCR. Works specific to text localization are proposed by Gupta et al. [6]. Liao et al. [11,13] augments such work to real-time detections in the end-to-end scenes. Karatzas et al. [9] and Bušta et al. [3] present better solutions in terms of accuracy and speed. The problem of scene-text spotting, however, remains complicated owing to variations in illumination, capturing methods and weather conditions. Moreover, the movement of the camera (or objects containing text) and motion blur in videos can make it harder to recognize the scene-text correctly. There has been a rising interest in end-to-end scene-text recognition in images over the last decade [2,16,10,9,3]. Recent text-spotters by Bušta et al. [3,4] include deep models that are trained end-to-end but with supervision at the level of text as well as at the level of words and text-boxes. The two recent breakthroughs in this direction, which work directly on complete scene images without supervision at the level of text boxes, are:

1. STN-OCR by Bartz et al. [2]: A single neural network for text detection and text recognition. The model contains a spatial transformation network that encodes the input scene image. It then applies a recurrent model over the encoded image features to output a sequence of grids. Combining the grids and the input image returns the series of word images present in the scene. Another spatial transformer network process the word images for recognition. This work does not need supervision at the level of detection.
2. Attention-OCR by Wojna et al. [19]: This work employs an inception network (proposed by Szegedy et al. [18]) as an encoder and an LSTM with attention as a decoder. The work is interesting because it does not involve any cropping of word images but works on the principle of soft segmentation through attention. The attention-OCR model performs character-level recognition directly on the complete scene image thus utilizing the global context while reading the scene. This model has an open-source TensorFlow (a popular library for deep learning by Abadi et al. [1]) implementation.

Both these works experiment on French Street Name Signs (FSNS) dataset, on which Attention-OCR performs the best. The Attention-OCR model also outperforms another line-level segmentation-based method (refer to work by Smith et al. [17]) on the FSNS dataset. Recently, the OCR-on-the-go model

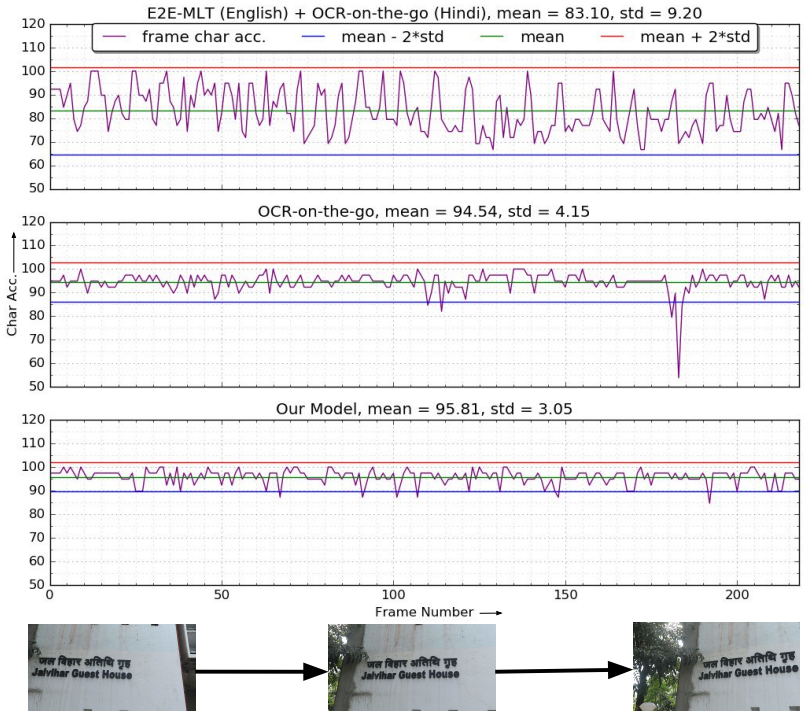


Fig. 2: Frame wise accuracy of 3 text-spotters on a simple video exhibiting *pan*

outperforms these models on the FSNS dataset using a multi-head attention mechanism [15]. In this work, we set new benchmarks for reading Indian street signs in a large number of video frames. The FSNS dataset contains around $10M$ images annotated with end-to-end transcriptions similar to ours. Different large-scale datasets are available in English. Uber-Text by Zhang et al. [21] include over $0.1M$ images annotated at line-level, captured from 6 US cities. Reddy et al. [14] annotate 1000 video clips from the BDD dataset [20] at line-level. We provide end-to-end transcriptions for our dataset similar to FSNS. Additionally, we also share noisy annotations at word-level and paragraph-level for each frame.

3 Motivation

We motivate our work of training the scene-text spotting models on the real (as well as synthetic) videos captured via continuous camera movements. Various end-to-end scene-text spotters, such as the ones proposed by Bušta et al. [4,3], train on synthetic as well as augmented real data to cover different capturing perspectives/orientations. The problem, however, is that during the training phase, such models do not exploit all the continuous perspectives/orientations

captured by the camera movement (or scene movement). Thus the OCR output fluctuates when tested on all/random video frames. Also, to deploy such models on real-time videos, two scenarios may occur. Firstly, the multi-frame consensus is desirable to improve OCR accuracy or interactive systems. Secondly, since it is computationally expensive to process each frame for readability, it is not possible to verify the quality of the frame to be OCR-ed. In any of these scenarios, the recognition system needs to work reasonably well on continuous video frames.

We present the frame level accuracy of E2E-MLT proposed by Bušta et al. [4] on an 8 second video clip with a frame size of 480×260 in the first plot of Figure 2 (with sample frames shown at the bottom). Since the model does not work for Hindi, we recognize the Hindi text using *OCR-on-the-go* model [15] . As shown, the E2E-MLT model produces the most unstable text on a simple video (from the test dataset) with the average character accuracy of 83.1% and the standard deviation of 9.20. The reason for this is that E2E-MLT, which does not train on continuous video frames, produces extra text-boxes on many of them during the detection phase. Thus extra noise characters or strings are observed during recognition. For instance, the correct text "Jalvihar Guest House" appears in 18 frames, the text "Jalvihar arG Guest House" appears in 10 frames, and the text "Jalvihar G Guest House" appears in 9 frames. The text "Jalvihar G arGu Guesth R H House" appears in one of the frame.

The instability in the video text, however, reduces when we use the *OCR-on-the-go* model by Saluja et al. [15] to read these video frames. As shown in the second plot of Figure 2, we achieve the (higher) average character accuracy of 94.54% and (lower) standard deviation of 4.15. This model works on the principle of end-to-end recognition and soft detection via unsupervised attention. The instability further reduces, as shown in the third plot of Figure 2, when we train our CATALIST model on the continuous video datasets proposed in this work.

4 Methodology

We use end-to-end attention-based encoder-decoder model proposed by Wojna et al. [19]. For better inference of attention masks, and improved recognition, we use the multi-head version of this model, proposed by Saluja et al. [15]. In Figure 3, we present the CATALIST model, that uses multi-task learning to update attention masks. Each mask is updated based on two loss functions. For end-to-end supervision, we use cross-entropy loss. To train attention heads, we use dice loss [12] between the predicted masks and the segmented masks obtained using text-boxes from SynthText proposed by Gupta et al. [6]. We also transform the synthetic images, along with text-boxes, to form videos which we describe in the end of this section.

4.1 The CATALIST model

As shown in Figure 3, the powerful inception-based encoder proposed by Szegedy et al. [18], which performs multiple convolutions in parallel, enhances the ability

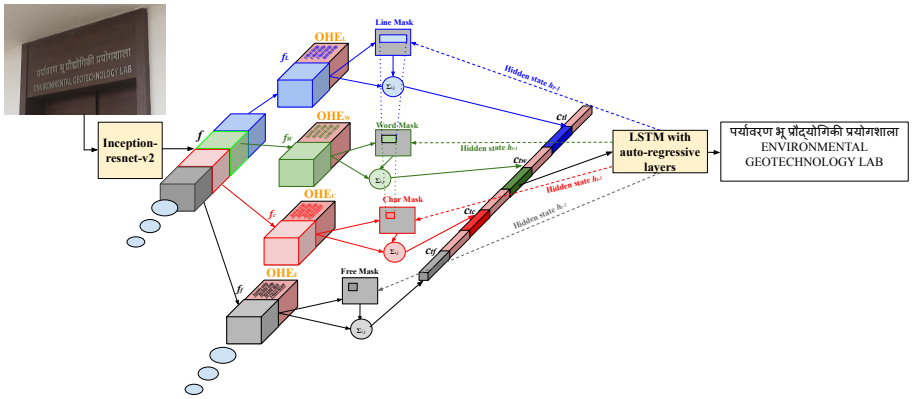


Fig. 3: CATALIST tames attention mask at multiple levels of granularity. The first three masks, namely line, word, and char mask, are supervised. The remaining attention masks are set free. Figure shows the first four attention masks.

to read the text at multiple resolutions. We extract the features f from the input image using the inception-based encoder. Moreover, the multi-head attention mechanism in our model exploits: i) the splits of feature f into f_L, f_w, f_c, f_f[2], etc. (refer Figure 3), ii) one-hot-encoded (OHE) vectors (OHE_L, OHE_w, OHE_c, OHE_f, etc.[3]) for both x and y coordinates of each feature split, iii) hidden layer at the previous decoding step (h_{t-1}) of an LSTM (decoder). To learn the attention at multiple levels of granularity, we provide supervision to the first three masks in the form of the line, word, and character level segmented binary images. The remaining masks are set free to assist/exploit end-to-end recognition/supervision. Thus we refer to the first three of them as line mask, word mask, and char mask in Figure 3. We also hard-code the word mask to remain inside the line mask, and the character mask to remain inside the word mask. The context vectors (c_L, c_w, c_c, c_f, etc.), which are obtained after applying the attention mechanism, are fed into the LSTM to decode the characters in the input image.

It is important to note that for each input frame, the features f and splits remain fixed, whereas the attention masks move in line with the decoded characters. Thus, we avoid using simultaneous supervision for all the character masks (or word masks or line masks) in a frame. Instead, we use a sequence of masks (in the form of segmented binary images) at each level for all the video frames. We accomplish this by keeping the word-level as well as the line-level segmented images constant and moving the character level segmented images while decoding the characters in each word. Once the decoding of all the characters in a

[2] f_L represents the features used for producing line masks, f_w represents features used for word masks, f_c represents features used for character masks, and f_f represents features used for free attention masks

[3] for the corresponding features f_L, f_w, f_c, f_f, etc.

word is complete, the word level segmented image moves to the next word in the line, and the character level image keeps moving as usual. Once the model has decoded all the characters in a line of text, the line (and word) level segmented image moves to the next line, and the character level segmented image continues to move within the word image.

4.2 The ALCHEMIST videos

We generate synthetic data for training the attention masks (as well as the complete model) using our data generator, which we refer to as ALCHEMIST. ALCHEMIST enables the synthetic creation of large scene-text video datasets. ALCHEMIST overlays synthetic text on videos under 12 different transformations described in the next section. By design, we preserve the information of the transformation performed, along with information of the character, word, and line positions (as shown in Figure 7). This information in the synthetic data provides for fairly detailed supervision on the attention masks in the CATALIST model. We build ALCHEMIST as an extension of an existing fast and scalable engine called SynthText proposed by Gupta et al. [6].

Methodology: According to pinhole camera model, a (2-d) point x (in homogeneous coordinate system) of image captured by a camera is given by equation 1.

$$x = K[R|t]X \tag{1}$$

Here K is the intrinsic camera matrix, R and t are rotation and translation matrices respectively, and X is a (3-d) point in *real world coordinates* in an homogeneous coordinate system.

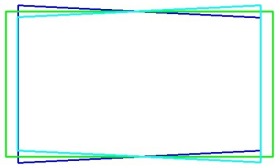


Fig. 4: For videos with camera *pan*, we find Homography between the corners of a rectangle and 4 points equidistant from them (which form one of the blue trapeziums).

For generating synthetic videos, we first select a fixed crop within the synthetic image (as denoted by the green rectangle in Figure 5). We then warp the corners of the crop by finding a planar homography matrix H (using algorithm given by Hartley et al. [7]) between the corner coordinates and four points equidistant from corners (direction depends on the kind of transformation as explained later). For Figure 4 (and Figure 5), we find the planar homography matrix H between corners of one of the blue trapezium and the green rectangle. Thus, instead of a 2D point x in the homogeneous coordinate system as

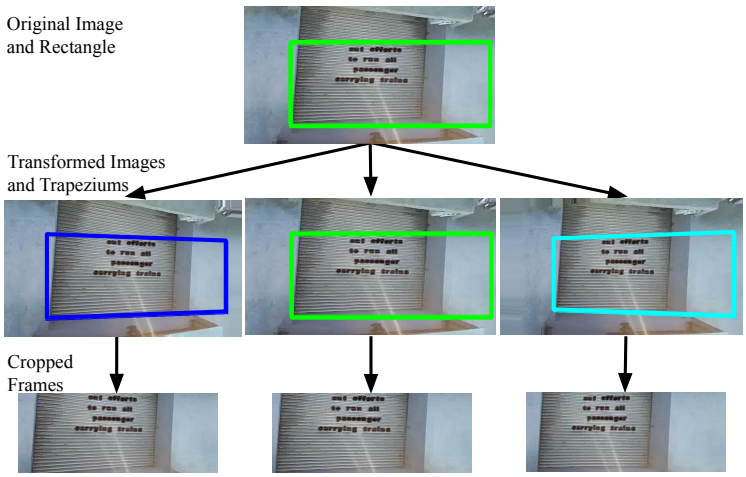


Fig. 5: Generating video with camera *pan* (3 frames at the bottom for dark-blue, green and light-blue perspectives respectively) from an image (at the top)

explained earlier, we get a translated point x_{new} defined in equation 2:

$$x_{new} = HK[R|t]X \tag{2}$$

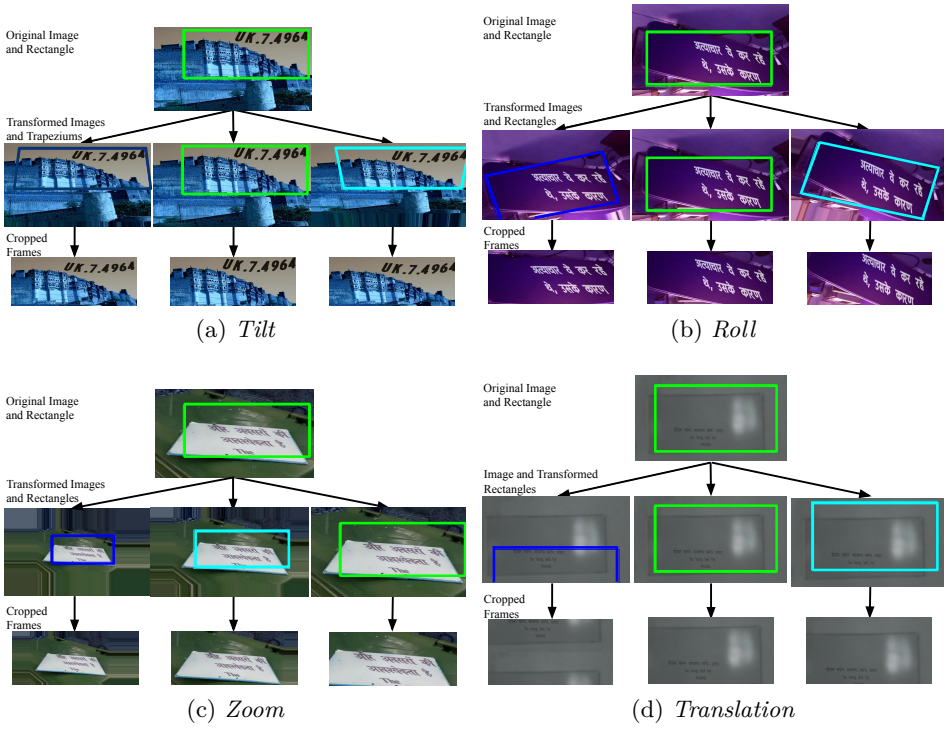
Here, H is the known homography. The above equation is simplified from the equation below:

$$x_{new} = KT[R|t]X = KTK^{-1}K[R|t]X \tag{3}$$

Here T is the unknown transformation matrix. We then warp the complete image using H and crop the rectangular region (refer green rectangle in Figure 5), to obtain the video frames. To find all the homography matrices for a video with camera *pan*, we consider the corners of the trapezium moving towards the rectangle corners. Once the homography matrix becomes the *identity matrix*, we move the corners of the trapezium away from the rectangle in the opposite direction to the initial flow (to form the mirrors of the initial trapeziums, e.g. light-blue trapezium in Figure 5).

The process for generating videos with camera *tilt* is similar to that of *pan*. The only difference is that the trapeziums in videos with camera *tilt* have vertical sides as parallel (as shown in Figure 6a) whereas the trapeziums in videos with camera *pan* have horizontal sides as parallel. For the videos with camera *roll*, we utilize the homography matrices between the corners of the rectangles rotating around the text center and the base (horizontal) box, as shown in Figure 6b.

For videos with camera *translation*, we use the regions a moving rectangle beginning from one text boundary to the other and generate the frames, as shown in Figure 6d. We make sure that the complete text, with rare partial occlusion of boundary characters, lies within each frame of the videos.



(a) *Tilt* (b) *Roll*

(c) *Zoom* (d) *Translation*

Fig. 6: Generating video with camera (a) *tilt*, (b) *roll*, (c) *zoom* and (d) *translation (frames at the bottom)*

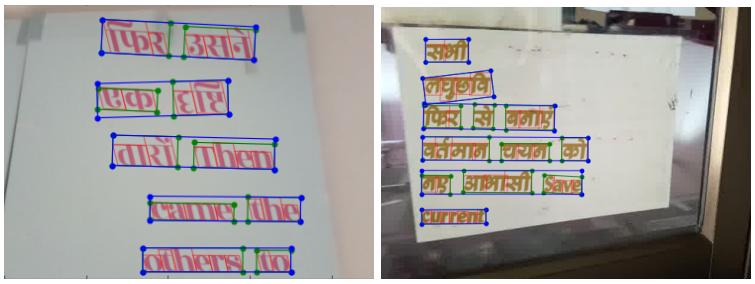


Fig. 7: Sample frames from the synthetic videos with multi-level text-boxes

We also use the homography H to transform the multi-level text-boxes in the cropped image. Figure 7 depicts sample video frames with text-boxes at the line, word, and character[4] levels – shown in blue, green, and red, respectively.

[4] For Devanagari (the script used for Hindi and Marathi), we carefully consider the boxes at the level of joint-glyphs instead of characters since rendering characters

Table 1: Distribution of videos in the CATALIST$_d$ dataset

S.No.	Transformation Type	Number of Videos
1.	Translation	736
2.	Roll	357
3.	Tilt	387
4.	Pan	427
5.	Zoom	402

4.3 The CATALIST$_d$ videos

We now present a new video-based scene-text dataset, which we refer to as CATALIST$_d$. Every video in CATALIST$_d$ contains scene-text, potentially in a combination of three different languages, namely, English, Hindi, and Marathi. For each such scene-text, we create 12 videos using 12 different types of camera *transformations*, broadly categorized into 5 groups:- (i) four types of *translation*, that could be left, right, up and down, (ii) two types of *roll*, including clockwise and anti-clockwise, (iii) two types of *tilt* which could be up-down or down-up motion, (iv) two types of *pan*, that is left-right and right-left , and (v) two types of *zoom* which could be in or out. We use a camera with a tripod stand to record all these videos to have a uniform control.

We summarize the distribution of different types of videos in Table 1. It is important to note that there are four types of translations, whereas there are only two types for all other transformations. We capture these videos at 25 fps with a resolution of 1920×1080.

5 Experiments

We synthesize around 12000 videos using ALCHEMIST data generator, which we use only for training the models. We use 50 Unicode fonts[5] and 18 license plate fonts[6] to render text in these videos. Here the duration and frame-rate for each video are 5 seconds and 25 fps, respectively. Moreover, we record a total of around $2k$ real videos (uniformly divided across 12 camera transformations) using a camera mounted over tripod stand for CATALIST$_d$ dataset. The setup allows smooth camera movements for *roll*, *tilt*, *pan* and *zoom*. We record the horizontal translation videos with the camera and tripod moving on a skateboard. Other translation videos, which exhibit top to bottom and reverse movements, have jitter because our tripod does not allow for smooth translation while recording such videos. We use a train:test split of 75:25, and carefully avoid letting any testing labels (as well as redundancy of the scenes) be present in the training data. We additionally record around $1k$ videos using handheld mobile phones and

individually (to obtain character level text-boxes) hamper glyph substitution rules that form the joint glyphs in Devanagari.

[5] http://indiatyping.com/index.php/download/top-50-hindi-unicode-fonts-free

[6] https://fontspace.com/category/license%20plate

use them for training the models. Finally, we also make use of the 640 videos shared by Saluja et al. [15]. We refer to the complete training dataset described above as $\text{CATALIST}_{\text{ALL}}$ in the next sections.

Fig. 8: A sample video frame from ICDAR'15 competition with text-boxes sorted using our algorithm

We further add the ICDAR'15 English video dataset of 25 training videos (13,450 frames) and 24 testing videos (14,374 frames) by Karatzas et al. [9] to the datasets. For each frame in the ICDAR'15 dataset, we first cluster the text-boxes into paragraphs and then sort the paragraph text-boxes from top-left to bottom-right. A sample video frame with the reading order mentioned above and the text-boxes sorted using our algorithm are shown in Figure 8. We visually verify that the reading order remains consistent throughout their appearance and disappearance in the videos. The reading order, changes when a new piece of text appears in the video or an old piece of text disappears from the video.

Although we record the controlled videos with a high resolution of 1920×1080, we work with the frame size of 480×260 for all videos owing to the more limited size of the videos captured on mobile devices, as well as for to reduce training time on a large number of video frames. To take care of resolution as well as to remove the frames without text, we extract the 480×260 sized clips containing the mutually exclusive text regions in the videos from the ICDAR'15 dataset. Features of size $14 \times 28 \times 1088$ are extracted from the mixed-6a layer

of inception-resnet-v2 [18]. The maximum sequence length of the labels is 180, so we unroll the LSTM decoder for 180 steps. We train all the models for 15 epochs.

Table 2: Test Accuracy on different datasets.

S. No.	Training Model	Training Data	Test Data	Char. Acc.	Seq. Acc.
1.	OCR-on-the-go (8 free masks)	OCR-on-the-go[7]	OCR-on-the-go 200 test videos	35.00 [15]	1.30
2.	CATALIST model (8 free masks)	$\text{CATALIST}_{\text{ALL}}$[8]		65.50	7.76
3.	CATALIST model (3 superv., 5 free masks)	$\text{CATALIST}_{\text{ALL}}$		**68.67**	**7.91**
4.	CATALIST model (8 free masks)	$\text{CATALIST}_{\text{ALL}}$	491 $\text{CATALIST}_{\text{d}}$ videos	**73.97**	6.50
5.	CATALIST model (3 superv., 5 free masks)			73.60	**7.96**
6.	CATALIST model (8 free masks)	$\text{CATALIST}_{\text{ALL}}$	24 ICDAR'15 Competition videos	34.37	**1.70**
7.	CATALIST model (3 superv., 5 free masks)			**35.48**	0.72

6 Results

We now present the results of the CATALIST model on the different datasets described in the previous section. It is important to note that we use a single CATALIST model to jointly train on all the datasets ($\text{CATALIST}_{\text{ALL}}$) at once.

Results on the OCR-on-the-go dataset In the first three rows of Table 2, we show the results on the test data used for *OCR-on-the-go* model by Saluja et al. [15]. The first row shows the results of this work. As shown in row 2, there is a dramatic improvement in character accuracy by 30.50% (from 35.0% to 65.5%) as well as sequence accuracy by 6.46% (1.30% to 7.76%), due to proposed CATALIST model as well as the ALCHEMIST and $\text{CATALIST}_{\text{d}}$ datasets we have have created. Adding the multi-level mask supervision to the CATALIST model further improves the accuracies by 3.17% (from 65.50% to 68.67%) and 0.15% (from 7.76% to 7.91%).

[7] 640 real videos + $700k$ synthetic images
[8] $3.7k$ real videos + $12k$ synthetic videos

Results on the CATALIST$_{\mathbf{d}}$ dataset As shown in the fourth and fifth row of Table 2, the gain of 1.46% (6.50 to 7.96) is observed in the sequence accuracy of the CATALIST model, when we use the mask supervision. We, however, observe a slight gain of 0.37% in character level accuracy when all the masks are set free (*i.e.*, trained without any direct supervision).

Results on the ICDAR'15 competition dataset We observe a gain of 1.11% (from 34.37% to 35.48%) in character-level accuracy on the ICDAR'15 competition dataset due to mask supervision. The end-to-end sequence accuracy for this dataset is as low as 1.70% for the model with all free masks and further lowers (by 0.98%) for the model with the first 3 masks trained using direct semantic supervision. We observe that the lower sequence accuracy for this dataset is due to the complex reading order in the frames.

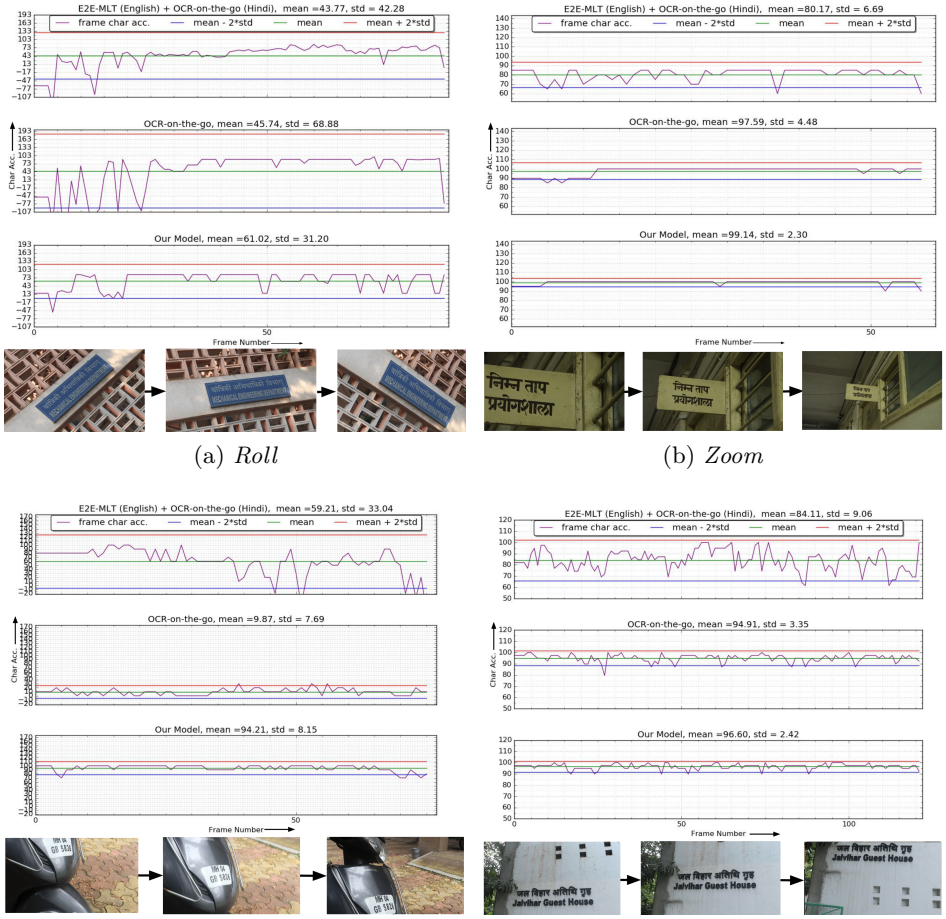
7 Frame-wise accuracies for all transformations

In Figure 2, we presented the frame-level accuracy of E2E-MLT (with the Hindi text recognized using *OCR-on-the-go* model), *OCR-on-the-go* model, and the present work on an 8 second video exhibiting *pan*. In this section, we present the frame-level accuracy of the above mentioned text-spotters for the other transformations: *roll*, *zoom*, *tilt*, and *translation*. The accuracy plots for a video with 88 frames (at 25 fps) exhibiting *roll* (clockwise) is shown in Figure 9a. We use the formulae in Equation 4 for calculating the character accuracy taking noise characters into consideration.

$$Accuracy = 100 * \frac{length(GT) - edit\_distance(P, GT)}{length(GT)} \tag{4}$$

Here, GT denotes the ground truth sequence and P is the predicted sequence. For some of the frames (with large amounts of transformations), predicted sequence contains a lot of noise characters. As a result, the *edit_distance* between predicted sequence and ground truth sequence may go higher than the length of ground truth sequence. Thus we get negative accuracy for some of the frames in Figure 9a. As shown, our model has the highest mean and lowest standard deviation for this video as well. It demonstrates the importance of the CATALIST model trained with mask supervision on continuous video frames. Furthermore, it is essential to note that all the models perform poorly at the start of this video due to larger amounts of rotation as compared to the later parts of the video.

In Figure 9b, we present similar plots for a video with 58 frames exhibiting *zoom* (out). The signboard in the video only contains Hindi text. The E2E-MLT model, however, outputs some English characters due to script mis-identification. Owing to this, the overall accuracy of the topmost plot (E2E-MLT + *OCR-on-the-go*) in Figure 9b is most unstable. Our model again achieves the highest mean and lowest standard deviation across all the video frames.



(a) *Roll* (b) *Zoom*

(c) *Tilt* (d) *Translation*

Fig. 9: Frame-wise accuracy of 3 text-spotters on videos exhibiting (a) *roll*, (b) *zoom*, (c) *tilt* and (d) *translation*

The plots for a video with 75 frames exhibiting *tilt* (up-down) is shown in Figure 9c. As shown, contrary to other figures, the *OCR-on-the-go* model performs poorly on this video. The reason for this is that the model perhaps overfits to its license plates dataset. E2E-MLT generalizes well with respect to *OCR-on-the-go* model, however, our model has the highest average accuracy. In Figure 9d, we present similar plots for a video with 121 frames exhibiting *translation* (upward). As discussed earlier in Section 5, the video clips recorded with the vertical camera movements in the setup possess jitter because the tripod does not allow for smooth translation while recording such videos. Our model, however, outputs

the text with the highest accuracy and lowest standard deviation for the video we present in Figure 9d.

8 Conclusion

In this paper, we presented CATALIST, a multi-task model for reading scene-text in videos and ALCHEMIST, a data generator that produces the videos from text images. These synthetic videos mimic the behaviour of videos captured with five different camera movements. We also presented the $\text{CATALIST}_{\text{d}}$ dataset of around two thousand real videos recorded with the camera movements mentioned above. By training the CATALIST model on both real and synthetic videos, we set new benchmarks for the task of reading multi-lingual scene-text in Hindi, Marathi, and English. The multi-level mask supervision improved either character or sequence (or both) accuracy on three different datasets with varying complexities.

9 Future Work

The camera movement information in $\text{CATALIST}_{\text{d}}$ dataset is ideal for Capsule Network [8,5]. Unlike conventional CNNs, capsule networks are viewpoint invariant. The transformation information can help capsules with video scene-text detection by helping the network learn about camera movements.

Acknowledgment: We thank Shubham Shukla for dataset collection and annotation efforts.

References

1. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al.: Tensorflow: A System for Large-Scale Machine Learning. In: 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16). pp. 265–283 (2016)
2. Bartz, C., Yang, H., Meinel, C.: Stn-ocr: A single neural network for text detection and text recognition. arXiv preprint arXiv:1707.08831 (2017)
3. Bušta, M., Neumann, L., Matas, J.: Deep textspotter: An end-to-end trainable scene text localization and recognition framework. International Conference on Computer Vision (2017)
4. Bušta, M., Patel, Y., Matas, J.: E2E-MLT-an Unconstrained End-to-End Method for Multi-Language Scene Text. In: Asian Conference on Computer Vision. pp. 127–143. Springer (2018)
5. Duarte, K., Rawat, Y.S., Shah, M.: Videocapsulenet: A simplified network for action detection. arXiv preprint arXiv:1805.08162 (2018)
6. Gupta, A., Vedaldi, A., Zisserman, A.: Synthetic Data for Text Localisation in Natural Images. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2315–2324 (2016)

7. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge university press (2003)
8. Hinton, G.E., Sabour, S., Frosst, N.: Matrix capsules with em routing. In: International conference on learning representations (2018)
9. Karatzas, D., Gomez-Bigorda, L., Nicolaou, A., Ghosh, S., Bagdanov, A., Iwamura, M., Matas, J., Neumann, L., Chandrasekhar, V.R., Lu, S., et al.: Icdar 2015 competition on Robust Reading. In: 2015 13th International Conference on Document Analysis and Recognition (ICDAR). pp. 1156–1160. IEEE (2015)
10. Karatzas, D., Shafait, F., Uchida, S., Iwamura, M., i Bigorda, L.G., Mestre, S.R., Mas, J., Mota, D.F., Almazan, J.A., De Las Heras, L.P.: ICDAR 2013 Robust Reading Competition. In: 2013 12th International Conference on Document Analysis and Recognition. pp. 1484–1493. IEEE (2013)
11. Liao, M., Shi, B., Bai, X., Wang, X., Liu, W.: Textboxes: A fast text detector with a single deep neural network. In: AAAI. pp. 4161–4167 (2017)
12. Milletari, F., Navab, N., Ahmadi, S.A.: V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: 2016 fourth international conference on 3D vision (3DV). pp. 565–571. IEEE (2016)
13. Minghui Liao, B.S., Bai, X.: TextBoxes++: A single-shot oriented scene text detector. CoRR **abs/1801.02765** (2018)
14. Reddy, S., Mathew, M., Gomez, L., Rusinol, M., Karatzas, D., Jawahar, C.: RoadText-1K: Text Detection & Recognition Dataset for Driving Videos. In: 2020 IEEE International Conference on Robotics and Automation (ICRA). pp. 11074–11080. IEEE (2020)
15. Saluja, R., Maheshwari, A., Ramakrishnan, G., Chaudhuri, P., Carman, M.: Robust End-to-end Systems for Reading License Plates and Street Signs. In: 2019 15th IAPR International Conference on Document Analysis and Recognition (ICDAR). pp. 154–159. IEEE (2019)
16. Shi, B., Bai, X., Yao, C.: 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 **39**(11) (2017)
17. Smith, R., Gu, C., Lee, D.S., Hu, H., Unnikrishnan, R., Ibarz, J., Arnoud, S., Lin, S.: End-to-end interpretation of the french street name signs dataset. In: European Conference on Computer Vision. pp. 411–426. Springer (2016)
18. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2818–2826 (2016)
19. Wojna, Z., Gorban, A.N., Lee, D.S., Murphy, K., Yu, Q., Li, Y., Ibarz, J.: Attention-Based Extraction of Structured Information from Street View Imagery. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR). vol. 1, pp. 844–850. IEEE (2017)
20. Yu, F., Xian, W., Chen, Y., Liu, F., Liao, M., Madhavan, V., Darrell, T.: BDD100K: A Diverse Driving Video Database with Scalable Annotation Tooling. arXiv preprint arXiv:1805.04687 **2**(5), 6 (2018)
21. Zhang, Y., Gueguen, L., Zharkov, I., Zhang, P., Seifert, K., Kadlec, B.: Uber-text: A Large-Scale Dataset for Optical Character Recognition from Street-Level Imagery. In: SUNw: Scene Understanding Workshop-CVPR. vol. 2017 (2017)