

Large Scale Scene Text Verification with Guided Attention

Dafang He^{1*}, Yeqing Li^{2*}, Alexander Gorban², Derrall Heath², Julian Ibarz², Qian Yu², Daniel Kifer¹, C. Lee Giles¹

The Pennsylvania State University¹, Google Inc².
 {duh188,giles}@ist.psu.edu, dkifer@cse.psu.edu,
 {yeqing,dheath,gorban,julianibarz,qyu}@google.com

Abstract. Many tasks are related to determining if a particular text string exists in an image. In this work, we propose a new framework that learns this task in an end-to-end way. The framework takes an image and a text string as input and then outputs the probability of the text string being present in the image. This is the first end-to-end framework that learns such relationships between text and images in scene text area. The framework does not require explicit scene text detection or recognition and thus no bounding box annotations are needed. It is also the first work in scene text area that tackles such a **weakly labeled problem**. Based on this framework, we developed a model called *Guided Attention*. Our designed model achieves better results than several state-of-the-art scene text reading based solutions for a challenging Street View Business Matching task. The task tries to find correct business names for storefront images and the dataset we collected for it is substantially larger, and more challenging than existing scene text dataset. This new real-world task provides a new perspective for studying scene text related problems.

Keywords: Scene Text · Verification · End to End Model · Attention · Weakly Labeled Dataset.

1 Introduction

Our streets are full of text such as street names, street numbers, store names, etc. These text are important clues for understanding the real world. With this rapidly growing new source of data, it has become more and more important to learn how to extract useful text information out of images. This task is usually referred to as scene text reading.

Many researchers divide the scene text reading problem into two sub-problems: text detection [2,3,12,26,5,13] and text recognition [6,8,15]. In order to build such an applicable system, the two components are usually combined sequentially into a complex system even though they are developed separately. In many applications, we don't need to precisely transcribe every single word in an image - we

* The two authors contribute equally

⁰ The work is done while Dafang is in Internship at Google.



Fig. 1: An example application of scene text verification task. Our model takes a business storefront image and a potential business name, and then directly outputs how likely the storefront image matches the business name. The red text means it is the ground truth for this image and our model gives it a high score.

only need to tell if some text prominently exists in an image and we call this problem: **scene text verification**. If we use scene text reading to formulate it, we will be facing several challenges: (1) Building a robust text understanding system is difficult, especially considering the possible differences in the domains of images in real usage and in training. A detector trained on currently available public datasets (e.g. ICDAR 2015 [10]) may have high possibility of failure on the new domain of images the system is deployed on. (2) Scene text is normally highly artistic. Heuristic based text matching is usually applied afterwards for verification, and often error are introduced when connecting these pieces. (3) A large amount of fully annotated data is needed in order to train a detector, which is expensive to obtain. In this work, we argue that we do not need separate text detection and text recognition systems for text verification problem.

We have observed that, in a lot of situations, an explicit text transcriber is unnecessary. Often a context is available, which can be used to narrow down the attention (text candidates). For example, if a person wants to find the reviews of a restaurant, they may want to just take a picture of the restaurant and we should be able to provide the reviews for them. In such cases, the algorithm only needs to pay attention to the name of the restaurant in order to identify it. It does not need to read any other text on the street, such as a 20% off advertising on clothes, or the opening hours on the door. Furthermore, a list of restaurant candidates could be obtained based on the geo-location so as to serve as candidates. Therefore, the actual problem we are trying to solve is instead: how likely is it that a sequence of words is in the image? In answering this question, an undiscriminating transcriber may be harmful since it could provide extra transcriptions that are irrelevant to the question (i.e., noise), and/or provide incorrect transcriptions which confuse the later process.

In order to address the previous concerns, we propose a new, end-to-end framework which takes input the image and the text and predict the possibility that the text string is contained in an image as in Fig. 1. The framework is expected to give a high probability if the input sequence is in the image, and thus it tries to verify the existence of a text string. It could also be regarded as trying to find a match between an image and a text string. Many applications could be built when we are able to give such a unified text-image score. For example, In addition to the restaurant case, Google Street View contains countless images taken on the street. Such a model could enable it to identify the business name from storefront images for back-end data processing.

We design a model called *Guided Attention* based on such framework. It does not need to do any explicit text detection or recognition. Instead it uses an attention mechanism that is guided by the input text string and decides whether the string is in the image or not.

This is the first work that tries to solve this problem in an end-to-end manner, and we study it in the context of business matching—given a storefront image, predict which business it represents. We collect millions of business storefront images through Google Maps API¹ for this task. Each image is associated with a list of candidate business names and the algorithm needs to find correct candidate among them.

We call this the Street View Business Matching (SVBM) dataset, and experiments show the effectiveness of our new framework in solving this challenging real-world problem. Our contributions are:

1. **New Problem** We study a new problem: scene text verification in an image. It verifies the existence of a certain text string and is closely related to scene text reading. It could be tackled with traditional method based on scene text detection and scene text recognition. It could also been seen as a sub-problem for scene text retrieval.
2. **New Dataset** We collect a large-scale dataset(SVBM) which contains 1.5M images for experiment. It is substantially larger than any existing public scene text datasets(e.g ICDAR2015 contains only 1500 images). The dataset contains storefront images from google street view, and algorithm needs to verify the existence of the business name in the images. It contains various of different languages and is much more challenging than existing public scene text datasets.
3. **New Framework** We propose an end-to-end trainable framework to this problem that does not do or require any explicit text detection or recognition. The framework is completely different from traditional method, and only requires image-text pairs for training, as opposed to a fully annotated dataset (e.g., bounding box labels), which are more expensive to create than simple yes/no labels. This greatly reduces the burden of dataset collection. We also propose a model based on this new framework and it achieves better performance in terms of precision/recall curve than existing text reading based solutions on SVBM. This new framework brings new insight into scene text area, and could inspire further research works.
4. **Extensive Experiments** We have done extensive experiments to study different properties of the framework. Our major experiments are on SVBM dataset, and we also evaluate the trained model in two public datasets: UberText [25] and FSNS [16]. These experiments confirm that our Guided Attention model is better suited at solving this task than a more traditional scene text reading based solution and by combing the proposed method and traditional method, we can achieve an even better performance in this task. We have also done ablation studies that show how important some design aspects of our model impact performance.

¹ <https://developers.google.com/maps/>

2 Related Work

Scene text detection is usually considered the most challenging part of building an end-to-end system, and many techniques have been developed to solve it. There are usually two approaches. The first approach treats the problem as a semantic segmentation problem, and several further steps are performed to obtain each individual text instance. This category includes methods that use fully convolutional neural networks (FCNs) to do semantic segmentation on text to extract text blocks [26,5,27]. The second category uses modified region proposal networks (RPNs) to detect text [18,13]. Both approaches achieve state-of-the-art performance on public datasets.

The challenges of scene text recognition are mainly due to the fact that text could be distorted in an image. Most regular (straight) text could be effectively read by first extracting CNN features and then using a RNN to predict the sequence. [6,14] used such techniques and combined them with CTC loss [4] to create powerful models for transcribing straight text. [21,15,22] tried to read curved scene text by using an attention mechanism, and spatial attention [21,22] has achieved the state-of-the-art performance.

Scene text retrieval [11,9], which aims at retrieving images based on text content, is closely related to scene text verification. The verification task could be seen as a subtask for scene text retrieval, as it only cares about the existence of text and no ranking is needed. Retrieval performance will benefit from a better verification model.

An end-to-end scene text reading system would combine the detection and recognition steps sequentially, and a fully annotated dataset is needed to train it. Instead, our proposed framework takes as input a character string and an image, and then verifies the existence of the string in the image in an end-to-end manner. Our framework is unique in that no explicit text detection or recognition is involved, and thus, there is also no need for a fully annotated dataset. The annotation for our framework is thus much easier to obtain since we only need images with corresponding lists of candidate strings and their labels.

Our work also has a loose connection to visual question answering[1] or image text matching[23]. However, our framework also has several unique properties: (1) input text is character sequence instead of words. (2) Simple binary classification is used as evaluation metric. (3) Order of input words shouldn't affect the results. For example, if a sequence of words "Street View Image" is in an image, then "Image Street View" should also be considered as positive.

In experiments, we evaluate our framework and model on the SVBM dataset and two public datasets: UbetText [25], FSNS [16]. SVBM dataset contains image-text pairs without bounding box annotations and it makes training a scene text detector impossible. The model has to learn from both the image and text in order to give a final prediction.

3 Method

3.1 Model Architecture

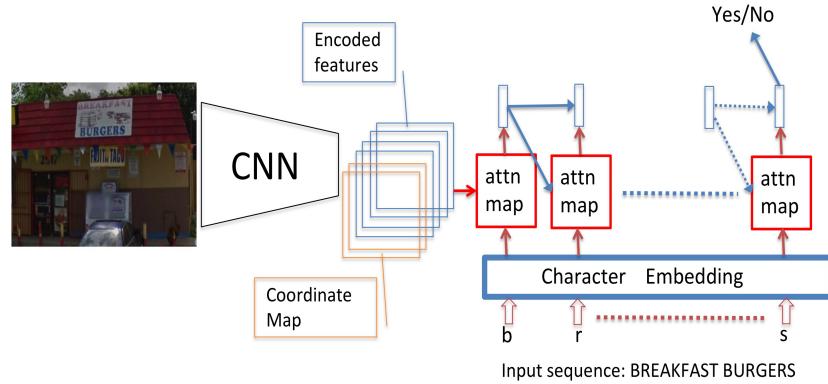


Fig. 2: The architecture of our scene text verification model. The model takes in a sequence of characters and an image, and predict the probability of the input text sequence being present in the input image.

The architecture is shown in Fig. 2. Our model consists of two major components: (1) a CNN-based image encoder network with coordinate encoding, (2) a guided attention decoder which selectively pools features from the encoded feature map and generates the final result.

CNN Encoder with Coordinate Map We trimmed InceptionV3 [17] to construct our image encoder, which builds a deep CNN by stacking some carefully designed sub-network structure with different scales.

Let I be the input image, and CNN encoded visual feature is denoted as $f_v = \text{CNN}(I)$. In order to capture the spatial information for each location in the feature map, we follow [21] to create a coordinate map f_{xy} for each image. The equation of such coordinate encoding could be expressed as $f_{xy} = E\text{ocode}(i, j)$. i, j denotes the x, and y indices of each spatial location in the feature map. Function *Encode* computes the one-hot encoding for each positng i, j , and we concatenate the coordinate map f_{xy} with original cnn feature f_v in depth dimension. We use \tilde{f} to denote the features augmented with position information. By combining such position information with each feature vector, the following decoding process could refer to them for better attention calculation and achieves better decoding results. Such scheme has been adopted by [21,22] and has been proved to be effective in scene text related tasks. Fig 3 illustrates the coordinate encoding.

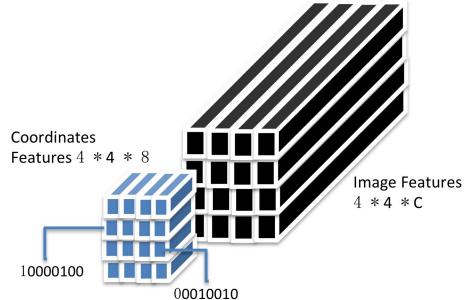


Fig. 3: Illustration of encoding the coordinates as a one-hot feature vector f_{xy} and concatenate it with the original visual features f_v . A feature map with $H \times W \times C$ will have corresponding coordinate feature map with size $H \times W \times (H+W)$

Guided Attention Decoder In the next step, the model tries to decode useful information from the cnn features \tilde{f} guided by the input character string. Let N be the number of characters of the input string and $S = S_1, \dots, S_N$ be the character level one-hot encoding of it. The embeddings of the characters are represented as $S^e = S_1^e, \dots, S_N^e$, which is learned end-to-end during training. Our goal is to compute

$$p_{valid} = \mathbb{P}(y = 1 | S_1, \dots, S_N, I), \quad (1)$$

where $y \in \{0, 1\}$ is the indicator of the existence of the text.

We use LSTM [7] as our recurrent function to encode sequential features. Let us denote h_t as the hidden state of the LSTM in time step t , then the update function of hidden state could be expressed as Eq. 2:

$$h_t = \text{LSTM}(h_{t-1}, S_t^e, \text{CT}(x_t)), \quad (2)$$

where $\text{CT}(x_t)$ represents the context vector generated in time step t . It can be computed based on Eq. 3:

$$\text{CT}(x_t) = \sum_{i=1}^W \sum_{j=1}^H \alpha_{(i,j)}^t * \tilde{f}_{(i,j)}, \quad (3)$$

where $\alpha_{(i,j)}^t$ represents the attention map. It is computed based on Eq. 4. $e_{(i,j)}^t$ represents how relevant is the feature $\tilde{f}_{(i,j)}$ to the current character embedding S_t^e . In this work, we choose to use attention function as in [19] in Eq. 5:

$$\alpha_{(i,j)}^t = \frac{\exp(e_{(i,j)}^t)}{\sum_{u=1}^W \sum_{v=1}^H \exp(e_{(u,v)}^t)}, \quad (4)$$

$$e_{(i,j)}^t = f_{attn}(h_{t-1}, \tilde{f}_{(i,j)}), \quad (4)$$

$$f_{attn}(h_{t-1}, \tilde{f}_{(i,j)}) = v^T \tanh(W h_{t-1} + U \tilde{f}_{(i,j)}), \quad (5)$$

where W, V are weight matrix that could be learned.

Let us denote $y = y_1, \dots, y_N$ as the output sequence probability and $y' = y'_1, \dots, y'_N$ as the groundtruth labels. Each y_i is the prediction based on information till character S_i^e . In training, we use cross entropy as our loss function, and we only calculate the loss based on the output of the last time step for each image and candidate pair as in Eq. 6:

$$\text{loss} = -\frac{1}{M} \sum_{i=1}^M (y'_i \log(y_{n_i}^i) + (1 - y'_i) \log((1 - y_{n_i}^i))), \quad (6)$$

where n_i is the length of the i th candidate and M is the number of training pairs.

3.2 Model Training and Sub-batch Architecture

In our problem setting, each image usually contains several positive (random shuffling of positive words) and a list of negative text.

In order to save the computation, for each image we have M parallel text input for it and the CNN tower will only need to be computed once. The number M is equal to $N_p + N_n$ where N_p, N_n represent the number of positive and negative examples sampled for each image, respectively. The total loss thus becomes a weighted cross entropy based on the ratio between positive examples and negative examples. In our experiment, we usually set N_p to be 1, and N_n to be 4. So there are 5 parallel recurrent sequences for each convolutional tower.

3.3 Hard Negative Mining(HNM)

During model training, the sampling scheme of the N_n number of negative training examples plays an important role for better performance. In our problem, the hardness of a negative candidate could be empirically determined by the edit distance of it with the corresponding positive candidates. Fig. 4 illustrates this idea.



Fig. 4: Example showing how we can determine the hardness of a negative example based on the corresponding positive candidates. The different color represents the different hardness levels.

We also observe that, in the SVBM dataset, most negative candidates are easy cases as they differ from the positive text by quite a lot. However, the negative samples that really confuse the network are from those hard cases. So we incorporate hard negative example mining as a fine tuning process for our model. We first define the hardness of an example based on Eq. 7.

$$\text{Hardness}(\hat{n}eg) = 1 - \min_{\forall p\hat{o}s \in \hat{P}} \frac{\text{edit_dis}(\hat{n}eg, p\hat{o}s)}{\max(\text{len}(p\hat{o}s), \text{len}(\hat{n}eg))} \quad (7)$$

\hat{P} represents the positive example set for a specific image. $p\hat{o}s$ is one positive sample and $\hat{n}eg$ represents the negative sample that we want to compute. The function `edit_dis` calculates the edit distance between the positive sample $p\hat{o}s$ and the negative sample $\hat{n}eg$. The hardness of a negative sample is determined by the minimal edit distance between the negative candidate and all the positive samples. The higher the score, the harder the negative sample.

The training process including HNM is thus as follows: (1) Train the network from scratch with evenly sampled positive and negative text for each image. (2) Finetune the network with evenly sampled positive text and evenly sampled negative text whose hardness score is larger than T . In our experiments, we set $T = 0.3$ to keep relatively harder text without removing too much of them. Note that we did this for both training set and testing set, so in the second phase, the testing examples are harder. During the first phase, the classification accuracy reached 95%, but in the beginning of the second phase, the classification accuracy dropped to 88%. This means that the harder examples, based on our definition, are actually more difficult for our model.

4 Experiments

The major experiments are on the SVBM dataset which contains millions of images, and we use it to study different properties of our model as well as the problem of scene text verification. We have also done two experiments on UberText and FSNS datasets with the model trained on SVBM dataset and baseline methods.

4.1 Dataset Description and Experiment Setting

SVBM The SVBM dataset is based on Street View images, and each image represents one business storefront. The storefront images could be obtained by the method proposed in [24], and all the images have associated geo-locations (for e.g, lat/lng) so that we can retrieve a list of candidate business names by filtering out far-away business. The number of candidates depends on the business density and the search radius. The goal is to find the correct business name among the list of candidates. No text bounding boxes are provided.

The dataset contains 1.39M training images and around 147K testing images. They are collected from various countries such as US, Brazil, England and etc. with various different languages. Each image has a number of candidates ranging from 10 to over 500. One of them is the correct business name, and all others are incorrect. As a preprocessing step, we convert all business names into lower case. So the character set contains a total of 37 alphanumeric characters (including space) plus characters in other languages which have high frequency in training set. We evaluate the dataset based on precision recall curve on a per-candidate basis.

In training, we use rmsprop as our optimization method, and the learning rate is set to 0.001. The batch size is set to 32, and the input image is resized and padded to 400 * 400 with random noise for simplicity. Each image is associated with 5 candidate text during training, and thus the batch size for attention decoder is 160(32 * 5). We use 70 cpu machines for training and it takes over 20 days to converge to an initial checkpoint.

UberText and FSNS There is currently no public dataset that is suitable for our purpose. We choose two public datasets that are relatively larger with a little bit different evaluation schemes. This study aims at demonstrating that by combining the proposed model with other traditional approaches, a better performance can be achieved for the text verification problem.

UberText [25] is a newly released dataset for scene text related problems. It is currently the largest fully annotated public dataset and contains 10K testing images. We evaluate it using the model trained on SVBM to show that our model can generalize well to other datasets. We choose the words that are of type: “business names” in the dataset and only evaluate the recall of these positive text in the verification problem.

FSNS [16] contains french street signs. The images are much easier than that in SVBM because the text are focused and clear. We randomly sample 49 text as negative text for each image for evaluation purpose.

4.2 Qualitative Results

In this section, we give several visual results illustrating the different properties of our trained guided attention model. More visual examples are in Fig. 9.



The Little Barbers	0.9625	The Little Barbers	0.9625	NHS	0.8947
Little Barbers	0.9524	Little Barber The	0.8749	PRIVATE	0.07
The	0.4268	Barber Little The	0.9249	DENTIST	
Little	0.4317	Barber The Little	0.9517	google	0.2026
Barber	0.9624	Little The Barber	0.9614		
Barber shop	0.1306				

Fig. 5: A qualitative evaluation on several specific behaviors of our model when we give it the same input image, but different input text. On the left is the input image. The first section in the table on the right shows results when we input a subset of ground truth text. The 2nd section in the table demonstrates when we have randomly shuffled groundtruth text as input. The last section represents results when the input text are not the corresponding business name, but are also in the image.

Subset of Text During training and subsequent evaluations, we use the full business name. However, it is interesting and important to study the property

of our model when we use subsets of the business name, or a slightly different business name to see how it performs. This is because that during annotation, we may not obtain exactly the same text as the actual business name. For example, “The Little Barbers” might be annotated as “Little Barbers”, and our model is expected to still give a high score for that. The first section in the table of Fig. 5 shows an example image and several such input text strings with their predicted probabilities.

We can see that, when we use informative subset text (For e.g, “Little Barbers”) from the ground truth as input text to the model, the model still gives a pretty high score. This meets our expectation as we care about the existence of the string (words) in the image. However, there are also several interesting findings: (1) If we use non-informative words as input(e.g: “The”), the model gives a relatively low score. (2) If we use text string that contains other words that are not in the image (For e.g, “Barber Shop”), the model gives a low score. These findings are interesting, and we believe that the model is looking for more informative words and makes decisions based on that. “The” is common in business names, so the model gives a lower score if we only use that as input text. “Barber shop” contains other words, which could possibly indicate that it’s some other business’ name. So our model gives a low score to it even though it contains the word “Barber” that is in the image.

Random Shuffled Text As we have discussed before, our model should be able to ignore the shuffling of input words. The 2nd section of the table in Fig. 5 shows the results when we try to input shuffled text with the same image into the model. This property is also important, as the order of words in annotation has no relationship with the spacial ordering of those words in image, and thus the trained model should ignore the word order, and only focus on the existence of the collection of words.

We can see that our model is somewhat invariant to the random shuffling of words as all the of them received high scores. This is the property that we expect, and it is important to text verification.

Non-Business Text The last section of the table in Fig. 5 shows an example where we have text inputs that are not the specific business name in the ground truth. We can see that the model still gives it a high score since it is in the image, and it also meets our expectation. There is also a failure example when we have “PRIVATE DENTIST” as input. This might happen when text are too small w.r.t the image size, so the attention could not capture it well. In addition to that, since our model is trained on business related text, we believe that this also caused the failure of this non-business text.

Hard Negative Mining Fig. 6 shows two examples illustrating the performance of the model trained before and after hard negative mining. We can see that after hard negative mining, the probability margins between the positive

	Before Hard Negative Mining	After Hard Negative Mining
Peluqueria estetica stela	0.9324	0.8924
peluqueria toni	0.7613	0.3293
francisco hidalgo tello	0.2643	0.1375
ferreteria mheva sl	0.2232	0.1328
puertodental sl	0.1989	0.1276

	Before Hard Negative Mining	After Hard Negative Mining
ceip santa catalina	0.9324	0.8604
aulario santa catalina	0.9057	0.1294
trastero 16	0.0781	0.1186
ropa africana	0.0618	0.0970
verde limon	0.0606	0.0868
peluqueria vanitas	0.0605	0.0819

Fig. 6: Two visual examples illustrating the performance gain after mining hard negative candidates for training. We observe that after HNM, the gap between the best prediction and the second best prediction has usually been increased.

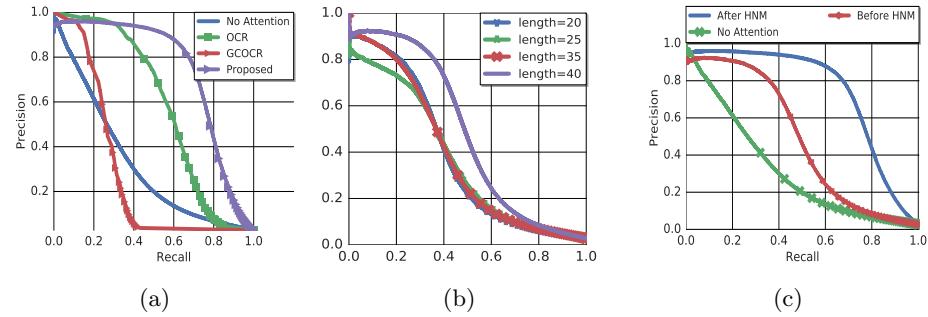


Fig. 7: (a) The Precision-Recall curve of our model compared with other baselines. (b) The Precision-Recall curve of our model w.r.t. different maximum length of text. (c) The Precision-Recall curve comparison before and after HNM.

and best negative sample for the two images have increased by 40%. This makes the model trained with HNM much more robust and the quantitative results of such comparison are in the followings.

4.3 SVBM Quantitative Evaluation

Baseline Models We compare several baseline models with our approach: (1) Google Cloud OCR(GCOCR)¹. (2) Attention OCR(OCR) [21]. (3) Show and Tell Model [20] with binary classification output. See supplementary material for details of the third model.

Model (1) tries to detect and read all the text indiscriminately. Then we do text based matching with the candidates to find the best candidate. This is one typical paradigm for this problem, but the model is trained on other fully annotated dataset. Model (2) is trained on the SVBM dataset directly.

¹ <https://cloud.google.com/vision/docs/ocr>

We simply force it to transcribe the whole image into the positive business name it represents. Text based matching is performed afterwards. Model (3) is a modification based on [19] (by changing the output as a binary classification). It could also be regarded as removing the attention mechanism in our framework. So we call it “no attention”.

Comparison w.r.t baselines We first show the comparison of our final model against other baseline methods. Fig. 7a shows the Precision/Recall curve. We can see that our end-to-end model(after HNM) outperforms all other baselines by a large margin. This address our two points: (1) text detector and recognizer trained on other fully annotated data couldn’t achieve good results in our SVBM dataset because they couldn’t accurately find and transcribe the business name. (2) Our task could be learned in an end-to-end way, and it outperforms transcription based method(OCR). In the following two evaluations, we show different settings that can improve the performance.

Evaluation w.r.t Maximum Length The maximum length of the character sequence could be tuned as an hyperparameter. This is an interesting aspect of our scene text verification problem. It is because when we need to decide which candidate business name the storefront represents, sometimes we only need, for example, the first 20 characters. In the example Fig 4, we only need the first 15 characters to determine which candidate is the positive one.

Note that this maximum length N only affects those candidates with length longer than it. We simply cut the longer text to keep the first N characters. This is also a difference between our problem w.r.t traditional VQA problem, since usually we will not cut the question in VQA task.

The Precision/Recall curve w.r.t the maximum length is in Fig. 7b. We can see that the peak is when the maximum length set to 40 for our task. However, other lengths also achieve reasonable performance. Besides, the value is dataset dependant. This experiment aims at illustrating an unique property of our problem and is important in deployment since the shorter the sequence, the faster the model could run. So there is a trade-off between the performance and efficiency. We use 40 as our final model choice.

Evaluation w.r.t Hard Negative Mining Whether to use hard negative mining makes a big difference in terms of the performance of the trained model. In Fig. 7c we show the results of the models trained before and after hard negative mining. Together we also show the results produced by a model with “no attention”. We can see that our attention mechanism is important to get good performance for our scene text verification tasks. By further adding hard negative mining process, the performance has been further improved, especially in the high accuracy region of the curve. This is also a difference between our problem with standard VQA problem since we can empirically define the “hardness”.

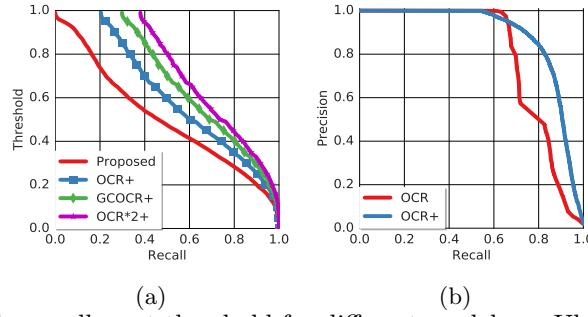


Fig. 8: (a) The recall w.r.t threshold for different models on UberText. We only evaluate positive text here. (b) P-R curve for OCR [21] and the ensembled model on FSNS.

4.4 UberText and FSNS Quantitative Evaluation

In this quantitative study, we study the problem that whether we can combine this model with traditional approaches to better solve text verification problem. The compared ensembled approaches are following: (1) OCR+: Ensemble of Attention OCR [21] and the proposed approach by taking the maximum of the scores output by the two models as the final score. This way of ensemble is used in all of the following compared approaches. (2) GCOOCR+: Simple ensemble of GCOOCR¹ and the proposed approach. (3) OCR*2+: Simple ensemble of GCOOCR¹, OCR [21] and the proposed approach.

Fig. 8a, 8b show the results for UberText and FSNS, respectively. The results show that when combining our model with other traditional approaches, we can achieve a better results than the original model itself. It also demonstrates that the “knowledge” learned from different models are not the same, and our model could also serve as complementary resources of information for traditional approaches.

5 Conclusion and Acknowledgement

In this work, we proposed a new problem: verifying the existence of a text string in an image and collected a large-scale dataset(SVBM) for evaluating the performance on this problem. Instead of traditional approaches based on scene text reading, we propose an end-to-end framework which takes both the image and the text as inputs and gives a unified result. We experimentally proved that model designed based on this framework achieved a better performance in SVBM for matching the business with the storefront image. It could also be combined with traditional methods into more powerful models as shown in the experiments in two public datasets. This work does not aim at giving the most sophisticated architecture, but proving that an end-to-end solution could be developed for such task to achieve a better performance without the need to fully annotate images at the text level.

This work was majorly done while Dafang is at internship at Google. We also thank NSF grant CCF 1317560 for support.

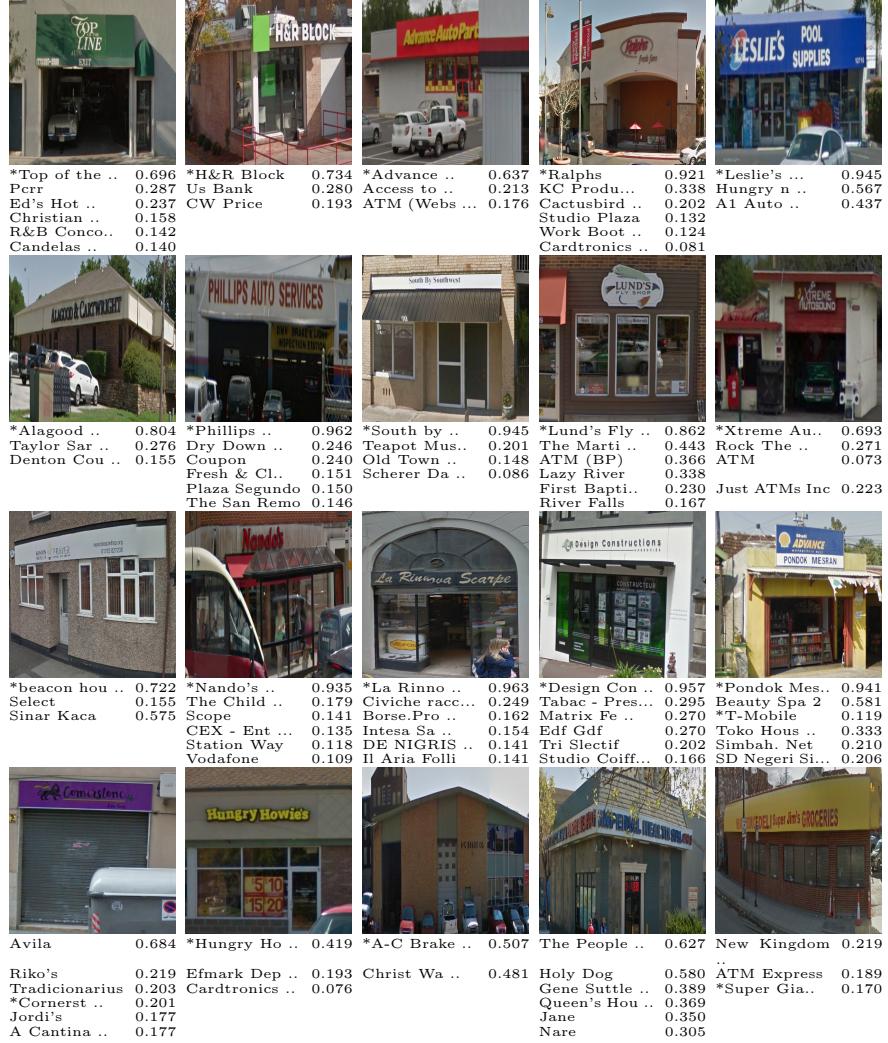


Fig. 9: Example images from the test set. We only visualize the top predicted candidates. The first three rows show the examples that we give good predictions (high score for positive candidate, low score for negative ones). The last row shows several failure examples. They are either caused by wrong prediction from our model, or by the problem of extremely unclear or non-exists text. It also shows that the dataset is challenging as it contains many practical problems such as extreme distortion. Such problems make it hard to correctly detect and transcribe the text. Our model, without the need to explicitly detect text, can achieve a better performance.

References

1. Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Lawrence Zitnick, C., Parikh, D.: Vqa: Visual question answering. In: Proceedings of the IEEE international conference on computer vision. pp. 2425–2433 (2015)
2. Chen, X., Yuille, A.L.: Detecting and reading text in natural scenes. In: Computer Vision and Pattern Recognition,CVPR. Proceedings of the IEEE Computer Society Conference on. vol. 2, pp. II–366. IEEE (2004)
3. Epshtain, B., Ofek, E., Wexler, Y.: Detecting text in natural scenes with stroke width transform. In: Computer Vision and Pattern Recognition,CVPR. Proceedings of the IEEE Computer Society Conference on. pp. 2963–2970. IEEE (2010)
4. Graves, A., Fernández, S., Gomez, F., Schmidhuber, J.: Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: Proceedings of the 23rd international conference on Machine learning. pp. 369–376. ACM (2006)
5. He, D., Yang, X., Liang, C., Zhou, Z., Ororbi, II, A.G., Kifer, D., Lee Giles, C.: Multi-scale fcn with cascaded instance aware segmentation for arbitrary oriented word spotting in the wild. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017)
6. He, P., Huang, W., Qiao, Y., Loy, C., Tang, X.: Reading scene text in deep convolutional sequences. In: AAAI Conference on Artificial Intelligence (2016)
7. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
8. Jaderberg, M., Simonyan, K., Vedaldi, A., Zisserman, A.: Reading text in the wild with convolutional neural networks. International Journal of Computer Vision **116**(1), 1–20 (2016)
9. Karaoglu, S., Tao, R., Gevers, T., Smeulders, A.W.: Words matter: Scene text for image classification and retrieval. IEEE Transactions on Multimedia **19**(5), 1063–1076 (2017)
10. 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: Document Analysis and Recognition (ICDAR), 2015 13th International Conference on. pp. 1156–1160. IEEE (2015)
11. Mishra, A., Alahari, K., Jawahar, C.: Image retrieval using textual cues. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3040–3047 (2013)
12. Neumann, L., Matas, J.: Real-time scene text localization and recognition. In: Computer Vision and Pattern Recognition,CVPR. Proceedings of the IEEE Computer Society Conference on. pp. 3538–3545. IEEE (2012)
13. Shi, B., Bai, X., Belongie, S.: Detecting oriented text in natural images by linking segments. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017)
14. 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 (2016)
15. Shi, B., Wang, X., Lyu, P., Yao, C., Bai, X.: Robust scene text recognition with automatic rectification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4168–4176 (2016)
16. 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)

17. 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)
18. Tian, Z., Huang, W., He, T., He, P., Qiao, Y.: Detecting text in natural image with connectionist text proposal network. In: Proceedings of the 11th European Conference on Computer Vision. pp. 56–72. Springer (2016)
19. Vinyals, O., Kaiser, L., Koo, T., Petrov, S., Sutskever, I., Hinton, G.: Grammar as a foreign language. In: Advances in Neural Information Processing Systems. pp. 2773–2781 (2015)
20. Vinyals, O., Toshev, A., Bengio, S., Erhan, D.: Show and tell: A neural image caption generator. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3156–3164 (2015)
21. Wojna, Z., Gorban, A., Lee, D.S., Murphy, K., Yu, Q., Li, Y., Ibarz, J.: Attention-based extraction of structured information from street view imagery. arXiv preprint arXiv:1704.03549 (2017)
22. Xiao Yang, Dafang He, Z.Z.D.K.C.L.G.: Learning to read irregular text with attention mechanisms. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17. pp. 3280–3286 (2017). <https://doi.org/10.24963/ijcai.2017/458>, <https://doi.org/10.24963/ijcai.2017/458>
23. Yan, F., Mikolajczyk, K.: Deep correlation for matching images and text. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3441–3450 (2015)
24. Yu, Q., Szegedy, C., Stumpe, M.C., Yatziv, L., Shet, V., Ibarz, J., Arnoud, S.: Large scale business discovery from street level imagery. arXiv preprint arXiv:1512.05430 (2015)
25. 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 2017. Hawaii, U.S.A. (2017)
26. Zhang, Z., Zhang, C., Shen, W., Yao, C., Liu, W., Bai, X.: Multi-oriented text detection with fully convolutional networks. In: Computer Vision and Pattern Recognition, CVPR. Proceedings of the IEEE Computer Society Conference on (June 2016)
27. Zhou, X., Yao, C., Wen, H., Wang, Y., Zhou, S., He, W., Liang, J.: East: An efficient and accurate scene text detector. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017)