AdaDNNs: Adaptive Ensemble of Deep Neural Networks for Scene Text Recognition

Chun Yang[†], **Xu-Cheng Yin**[†*], **Zejun Li**[†], **Jianwei Wu**[†],
Chunchao Guo[‡], **Hongfa Wang**[‡], and **Lei Xiao**[‡]
[†] Department of Computer Science and Technology, University of Science and Technology Beijing, Beijing, China
[‡] TEG, Tencent Co. LTD, Shenzhen, China
[*] Corresponding author: xuchengyin@ustb.edu.cn

Abstract

Recognizing text in the wild is a really challenging task because of complex backgrounds, various illuminations and diverse distortions, even with deep neural networks (convolutional neural networks and recurrent neural networks). In the end-to-end training procedure for scene text recognition, the outputs of deep neural networks at different iterations are always demonstrated with diversity and complementarity for the target object (text). Here, a simple but effective deep learning method, an adaptive ensemble of deep neural networks (AdaDNNs), is proposed to simply select and adaptively combine classifier components at different iterations from the whole learning system. Furthermore, the ensemble is formulated as a Bayesian framework for classifier weighting and combination. A variety of experiments on several typical acknowledged benchmarks, i.e., ICDAR Robust Reading Competition (Challenge 1, 2 and 4) datasets, verify the surprised improvement from the baseline DNNs, and the effectiveness of AdaDNNs compared with the recent state-of-the-art methods.

Scene text is widely used as visual indicators for navigation and notification, and text recognition from scene images and videos is one key factor for a variety of practical applications with reading in the wild (Ye and Doermann 2015; Yin et al. 2016; Tian et al. 2017), such as assisting for visually impaired people (Goto and Tanaka 2009; Sanketi, Shen, and Coughlan 2011), real-time translation (Shi and Xu 2005; Fragoso et al. 2011), user navigation (Minetto et al. 2011), driving assistance systems (Wu, Chen, and Yang 2005), and autonomous mobile robots (Létourneau et al. 2003).

Scene text (cropped word) recognition methods can be generally grouped into **segmentation-based word recognition** and **holistic word recognition**. Typical segmentation-based approaches over-segment the word image into small segments, combine adjacent segments into candidate characters, classify them using convolutional neural networks (CNNs) or gradient feature-based classifiers, and find an approximately optimal word recognition result (Bissacco et al. 2013; Jaderberg, Vedaldi, and Zisserman 2014). Because of complex backgrounds and diverse distortions, character segmentation is another more challenging task. Thereby, holistic word recognition approaches with deep neural networks are more impressive for text reading in the wild. **Word spotting**, the direct holistic approach, usually calculates a similarity measure between the candidate word image and a query word (Jaderberg et al. 2016; Gordo 2015). **Sequence matching**, the indirect holistic approach, recognizes the whole word image by embedding hidden segmentation strategies. For example, Shi et al. constructed an end-to-end training deep neural network for image-based sequence recognition (scene text recognition) (Shi, Bai, and Yao 2017).

However, there are a variety of grand challenges for scene text recognition (see samples in Fig. 1), even with recent deep neural networks (DNNs), where additional characters will be probably identified for text distortions and complex backgrounds, some characters are wrongly recognized for changing illuminations and complex noises, and characters are sometimes missed for low resolutions and diverse distortions.

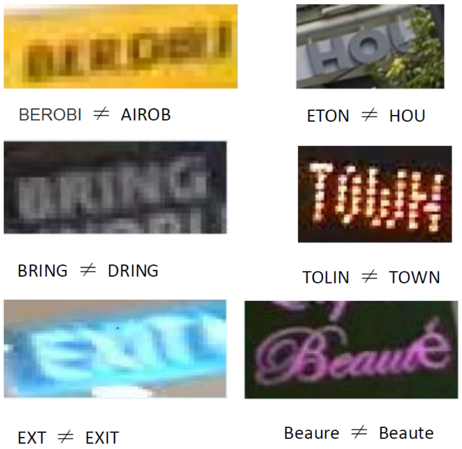

Figure 1: Some challenging examples (from 2015 ICDAR Robust Reading Competition Challenge 4 dataset) of scene text images which are incorrectly recognized by the baseline DNNs (see related descriptions in Experiments). The captions show the recognized text (left) versus the ground truth (right): additional characters, wrong characters and missing characters in target words.

Stochastic Gradient Descent (SGD) (Bottou 2010) and its variants have become the defacto techniques for optimizing DNNs, where SGD always leads to local minima, even though the popularity of SGD can be attributed to its ability to avoid spurious saddle-points and local minima (Dauphin et al. 2014). There are a plenty number of (more than million) possible local minima in DNNs (Kawaguchi 2016), and local minima with flat basins are supposed to generalize better in the learning system (Keskar et al. 2017). As a result, although different local minima often have similar error rates, the corresponding neural networks in DNNs tend to make different mistakes. This diversity and complementarity can be exploited via classifier ensemble (Huang et al. 2017).

There are two major ways for ensemble of deep neural networks. On the one hand, **different learning systems** with DNNs are first trained independently, and then the final system is a trivial ensemble of these different deep learning architectures via majority voting or averaging. For example, most high profile competitions in ImageNet [1] and Kaggle [2] are won by such ensemble techniques. Because of the huge computation complexity, this ensemble becomes uneconomical and impossible for most researchers in the universities and even in the small companies. On the other hand, **one learning system** with DNNs is first trained, and then the final ensemble selects and combines **neural network components** [3] in this only one system without incurring any additional training cost. Huang et al. proposed such an ensemble technique, called as Snapshot Ensembling, where a specific optimization strategy is designed to train DNNs and "model snapshots" (neural network components) in all cycles are combined for the final ensemble in the learning procedure (Huang et al. 2017). However, how to design the specific and effective optimization algorithms for DNNs is also a challenge.

In this paper, we propose a new and adaptive ensemble of deep neural networks (**AdaDNNs**) in the most simplest way, i.e., given trained neural networks (of all iterations) from a learned DNNs system [4], a subset of neural network components are simply selected and adaptively combined to perform the final predictions. And the ensemble is formally formulated as a Bayesian framework for classifier weighting and combination. We argue that because of the diversity and complementarity in DNNs with SGD, AdaDNNs via ensembling with diversity can improve robust performance of the final learning system. On the same time, because of the high accuracy of components in DNNs, AdaDNNs via combination with accurate neural network components can improve precision performance of the final classification system. A variety of experiments on several acknowledged benchmarks, i.e., ICDAR Robust Reading Competition (Challenge 1, 2 and 4) datasets, have shown that the **simple but effective** AdaDNNs improves largely from the baseline DNNs. Moreover, our proposed approach has the top performance compared with the latest state-of-the-art methods.

Related Work

Recognizing text in scene videos attracts more and more interests in the fields of document analysis and recognition, computer vision, and machine learning. The existing methods for scene text (cropped word) recognition can be grouped into segmentation-based word recognition and holistic word recognition. In general, segmentation-based word recognition methods integrate character segmentation and character recognition with language priors using optimization techniques, such as Markov models (Weinman et al. 2014) and CRFs (Mishra, Alahari, and Jawahar 2012; Shi et al. 2013). In recent years, the mainstream segmentation-based word recognition techniques usually over-segment the word image into small segments, combine adjacent segments into candidate characters and classify them using CNNs or gradient feature-based classifiers, and find an approximately optimal word recognition result using beam search (Bissacco et al. 2013), Hidden Markov Models (Alsharif and Pineau 2014), or dynamic programming (Jaderberg, Vedaldi, and Zisserman 2014).

Word spotting (Manmatha, Han, and Riseman 1996), a direct holistic word recognition approach, is to identify specific words in scene images without character segmentation, given a lexicon of words (Wang and Belongie 2010). Word spotting methods usually calculate a similarity measure between the candidate word image and a query word. Impressively, some recent methods design a proper CNN architecture and train CNNs directly on the holistic word images (Jaderberg et al. 2014; Jaderberg et al. 2016), or use label embedding techniques to enrich relations between word images and text strings (Almazan et al. 2014; Gordo 2015). Sequence matching, an indirect holistic word recognition approach, recognizes the whole word image by embedding hidden segmentation strategies. Shi et al. constructed an end-to-end train deep neural network for image-based sequence recognition (scene text recognition), where a convolutional recurrent neural networks framework (CRNN) is designed and utilized (Shi, Bai, and Yao 2017). In this paper, a similar CRNN architecture is used in AdaDNNs for recognizing scene text sequently and holistically.

Classifier ensemble can be mainly divided into two categories. The first one aims at learning multiple classifiers at the feature level, where multiple classifiers are trained and combined in the learning process, e.g., Boosting (Freund and Schapire 1997), Bagging (Breiman 1996), and Rotation Forest (Rodriguez, Kuncheva, and Alonso 2006). The second tries to combine classifiers at the output level, where the results of multiple available classifiers are combined to solve the targeted problem, e.g., multiple classifier systems (classifier combination) (Zhou 2012; Yin et al. 2014). AdaDNNs in this paper follows the second one. Namely, given multiple classifiers (neural network components sequently learned in DNNs), AdaDNNs is constructed by combining intelligently these component classifiers within a Bayesian-based formulation framework.

[1] `www.image-net.org`.

[2] `www.kaggle.com`.

[3] The neural network component means the resulting DNN of each iteration in the whole training procedure.

[4] Here, the DNNs system can be trained with conventional optimization algorithms (Bottou, Curtis, and Nocedal 2016), or even with the specific algorithms, e.g., Snapshot Ensembling (Huang et al. 2017).

Adaptive Ensemble of Deep Neural Networks

As we have known, both SGD and batch optimization can lead to different local minima in DNNs, and neural network components are always with diversity and complementarity. Conventionally, there are tens of thousands of iterations and also neural network components in the learning system of DNNs. Considering the acceptable computation complexity in the testing procedure, one thing is to quickly select a small subset of neural network components in different training iterations. At the same time, considering the high accuracy requirement, another thing is to adaptively combine this subset of neural network components and construct a final classification system. In the following, the unified framework of AdaDNNs is first formulated. Next, the detail procedure of AdaDNNs is then described.

Unified Framework

To formulate the ensemble decision, the individual classifier decisions can be combine by majority voting, which sums the votes for each class and selects the class that receives most of the votes. While the majority voting is the most popular combination rule, a major limitation of majority voting is that only the decision of each model is taken into account without considering the distribution of decisions.

In particular, all the possible models in the hypothesis space could be exploited by considering their individual decisions and the correlations with other hypotheses. Here, we use a Bayesian-based framework to combine classifiers. Given a sample x and a set H of independent classifiers, the probability of label y can be estimated by a Bayesian Model (BM) as

$$P(y|H,x) = \sum_{h_i \in H} P(y|h_i,x)P(h_i|x) \tag{1}$$

where $P(y|h_i,x)$ is the distribution of describing the correlation between decision y and $h_i(x)$, and $P(h_i|x)$ denotes the posterior probability of model h_i. The posterior $P(h_i|x)$ can be computed as

$$P(h_i|x) = \frac{P(D|h_i)P(h_i)}{\sum_{h_i \in H} P(D|h_i)P(h_i)} \tag{2}$$

where $P(h_i)$ is the prior probability of classifier h_i and $P(D|h_i)$ is the model likelihood on the training set D. Here, $P(h_i)$ and $\sum_{h_i \in H} P(D|h_i)P(h_i)$ are assumed to be a constant in Eq. 2. Therefore, BM assigns the optimal label y to y^* according to the following decision rule, i.e.,

$$\begin{aligned} P(y^*) &= argmax_y P(y|H,x) \\ &= argmax_y \textstyle\sum_{h_i \in H} P(y|h_i,x)P(h_i|x) \\ &= argmax_y \textstyle\sum_{h_i \in H} P(y|h_i,x)P(D|h_i) \\ &= argmax_y 2\textstyle\sum_{h_i \in H} P(y|h_i,x)P(D|h_i) - P(D) \\ &= argmax_y \textstyle\sum_{h_i \in H}(2P(y|h_i,x)-1)P(D|h_i) \\ &= argmax_y \textstyle\sum_{h_i \in H} W(y,h_i(x))P(D|h_i) \end{aligned} \tag{3}$$

where $W(y,h_i(x))$ is a function of y and $h_i(x)$. By multiplying the scaling factor $\lambda > 0$, $W(y,h_i(x))$ can have a different range in $\mathbb{R}$.

There are two key issues for optimizing Eq. 3. The first one is the calculation of $W(y,h_i(x))$. As mentioned above, $P(y|h_i,x)$ is the distribution of describing the correlation between decision y and $h_i(x)$. Thus, $W(y,h_i(x))$ can be derived from y, $h_i(x)$ and the distance between y and $h_i(x)$. Here, $W(y,h_i(x))$ is assumed to be computed as

$$W(y,h_i(x)) = I(y=h_i(x)) + U(y) * V(y,h_i(x)) \tag{4}$$

where both $U(*)$ and $V(*,\bullet)$ are functions. $I(y=h_i(x))$ returns 1 when $y=h_i(x)$; otherwise, $I(y=h_i(x))=0$.

For the scene text recognition task, on the one hand, with a given dictionary [5], $U(y)$ can be calculated as

$$U(y) = \left\{ \begin{array}{ll} 1 & y \in Dict \\ 0 & y \notin Dict \end{array} \right. \tag{5}$$

On the other hand, the correlation between y and $h_i(x)$ can be assumed by the function V of Cost Levenshtein Distance (CLD). In the traditional Levenshtein Distance, the cost of any two different characters is always 1. However, in spelling correction, the cost of two characters with similar shape tends to have a smaller distance. In this paper, we statistics the frequencies of different character pairs at the same location from the label and the hypothesis on the validation set (bootstrapped from the training set in Experiments), and calculate the cost of two different characters (a and b) as

$$cost(a,b) = 1 - P(a|b) \tag{6}$$

Note that if both y and $h_i(x)$ are from the given dictionary, then they will have a competitive relationship with each other. Thus, $V(y,h_i(x))$ can be calculated with

$$V(y,h_i(x)) = \left\{ \begin{array}{ll} F(-CLD(y,h_i(x))) & h_i(x) \in Dict \\ F(CLD(y,h_i(x))) & h_i(x) \notin Dict \end{array} \right. \tag{7}$$

where F is a function of the CLD between y and $h_i(x)$. By a heuristic approach, the values of F can be empirically assigned at the multiple integral points, and the values at other points can be calculated by the piecewise linear interpolation. An example of F is shown in Fig. 2. In general, F has a small range, e.g., $[1.5, 1.5]$ in Fig. **??**. So, the obtained weights from Eq. 4 are convenient for linear combination of classifiers.

The second issue is about generating voting candidates (more probable labels of the hypotheses). Obviously, the ground truth doesn't always appear in the decisions made by H. It is necessary to find an effective way to generate good candidates from all the decisions, i.e., to find a more probable label $y_i(x)$ from the existed initial label $y_i^0(x)$ of hypothesis h_i. Generally speaking, a good candidate means it has a small edit distance with most of the hypotheses. Following this idea, we propose an algorithm to semantically generate voting candidates (see Algorithm 1).

[5] In our experiments, a $90k$ word dictionary from (Jaderberg et al. 2016) is used as the given dictionary.

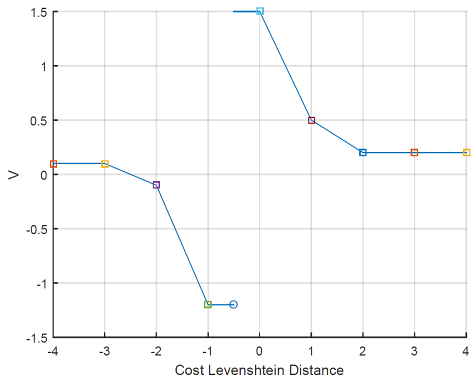

Figure 2: An example of F of describing the relationship between V (in Y-axis) and the Cost Levenshtein Distance (in X-axis).

```
Algorithm 1: Generating Voting Candidates.
Input:
    $H = \{h_1, h_2, ..., h_L\}$: the base classifier set, $|H| = L$.
    $Y_0$: the initial decisions made by $H$.
    $ED$: the measurement function of the pairwise distance.
    $\theta$: the upper bound of the distance between the candidate
        and the hypothesis.
Output:
    $Y$: the voting candidate set.
Parameter:
    $H^\star$: a subset of H, $\forall h_i^\star, h_j^\star \in H^\star, ED(h_i^\star, h_j^\star) \leq 2\theta$.
Procedure:
    $Y = \emptyset$.
For each $H^\star \subset H$;
  For each $y \in Y_0$:
    If $max_{h_i^\star \in H^\star} ED(y, h^\star(x)) \leq \theta$:
      $Y = Y \cup \{y\}$.
  End
End
```

In Algorithm 1, the searching process of $H^\star$ is an implicit computational way for $P(D|h_i)$. In our experiments, a special simple case of algorithm 1 is used, where during the voting candidates generation process, Y_0 is initialized only by H, the upper bound is set from θ to inf, and $P(D|h_i)$ is assumed to be a constant.

AdaDNNs Algorithm

Within the above framework, the procedure of AdaDNNs for scene text recognition includes three major steps, i.e., base classifiers generation, classifier combination, and ensemble pruning.

Base Classifiers Generation Ensembles work best if the base models have high accuracy and do not overlap in the set of examples they misclassify. Deep neural networks (DNNs) are naturally used as a base classifier generator for ensembles. On the one hand, DNNs have dramatically improved the state-of-the-art in many domains, such as speech recognition, visual object recognition and object detection, by being composed of multiple processing layers to learn representations of data with multiple levels of abstraction. On the other hand, during the training phase of one individual deep neural network, two snapshots with different minibatch orderings will converge to different solutions. Those snapshots often have the similar error rates, but make different mistakes. This diversity can be exploited by ensembling, in which multiple snapnots are average sampling and then combined with majority voting.

Focusing on scene text recognition, the CRNN model (Shi, Bai, and Yao 2017) is used to generate base classifiers (neural network components) as our text recognizer. CRNN uses CTC (Graves et al. 2006) as its output layer, which estimates the sequence probability conditioned on the input image, i.e. $P(h|x)$, where x is the input image and h represents a character sequence.

Classifier Combination for AdaDNNs The core of AdaDNNs is to calculate y^* (by Eq. 3), i.e., the calculation of F, which is a function of distance between y and $h_i(x)$. Here, F is represented by the set of values at the multiple integral points. These values are assigned with the highest recognition rate on the validation set. The detail procedure of the AdaDNNs ensemble is shown in Algorithm 2.

```
Algorithm 2: AdaDNNs (classifier combination).
Input:
    $H = \{h_1, h_2, ..., h_L\}$: the base classifier set, $|H| = L$.
    $Dict$: the given dictionary.
    $F$: a function of distance between $y$ and $h_i(x)$.
Parameter:
    $Y$: the voting candidates set generated by Algorithm 1.
Output:
    $y^*$: the label of prediction.
Procedure:
    Initialize $Y$ by $H$ and $Dict$.
For $y \in Y$:
    Calculate $P(y|H, x)$ through Eq. 2.
End
    Calculate $y^*$ through Eq. 3.
```

AdaDNNs Pruning In classifier ensemble, pruning can generally improve the ensemble performance. Here, we use Genetic Algorithm (GA) to pruning the ensemble. GA is a meta heuristic inspired by the process of natural selection that belongs to the larger class of evolutionary algorithms. GAs are commonly used to generate high-quality solutions for optimization and search problems by relying on bio-inspired operators such as mutation, crossover and selection.

In AdaDNNs pruning, firstly, a population of binary weight vectors is randomly generated, where 1 means the classifier is remained. Secondly, the population is iteratively evolve where the fitness of a vector w is measured on the validation set V, i.e., $f(w) = R_w^V$ (R stands for the recognition rate). Finally, the ensemble is correspondingly pruned by the evolved best weight vector w^*.

Experiments

To evaluate the effectiveness of the proposed AdaDNNs method, a variety of experiments for text (cropped word) recognition are conducted on acknowledged benchmark datasets. We first focused on the most challenging task, i.e., incidental scene text recognition (ICDAR Robust Reading Competition Challenge 4), trained our AdaDNNs learning system (on both the synthetic dataset from (Jaderberg et al. 2016) and the training set of Challenge 4), and performed comparative experiments. Then, we also conducted experiments of this learned AdaDNNs on other text recognition tasks, i.e., focused scene text recognition and born-digital text recognition (ICDAR Robust Reading Competition Challenge 1 and 2), and checked the generalization of AdaDNNs. Here, the baseline DNNs model, CRNN, is same to the one in (Shi, Bai, and Yao 2017). The official metrics in ICDAR 2011/2013/2015 Robust Reading Competition (Shahab, Shafait, and Dengel 2011; Karatzas et al. 2013; Karatzas et al. 2015) are used.

Experiments with Incidental Scene Text Recognition

The ICDAR 2015 Robust Reading Competition Challenge 4 database (Karatzas et al. 2015) is a widely used and highly competitive benchmark database for scene text recognition within complex situations in the recent 3 years. The public dataset includes a training set of $1,000$ images and a test set of 500, with more than $10,000$ annotated text regions (cropped words). Because of complex backgrounds, various illuminations and diverse distortions, this incidental scene text recognition topic is a very challenging task. In our experiments, a variety of methods are conducted and compared, i.e., the baseline DNNs, AdaDNNs, AdaDNNs pruning, the winning participation method in the official competition (marked as **bold words**), and the latest top submissions of the Robust Reading Competition (RRC) website [6] in 2017 (marked as *italic words*).

Table 1: Comparative results on 2015 ICDAR Challenge 4 dataset (incidental scene text recognition), where the comparative results are from the RRC website.

Date	Method	T.E.D	C.R.W	T.E.D.	C.R.W.
			(%)	(upper)	(upper)
2017/7/6	*Baidu IDL v3*	211.59	**80.02**	171.15	82.33
2017/7/6	*HIK_OCR_v3*	**191.25**	78.29	158.84	80.12
2017/6/29	*HKU-VisionLab*	258.59	72.03	212.17	74.19
2015/4/1	**MAPS**	1,128.01	32.93	1,068.72	33.90
-	Baseline DNNs	384.76	60.18	303.77	64.90
-	AdaDNNs	251.98	76.31	185.36	80.55
-	AdaDNNs Pruning	224.7	79.78	**147.11**	**84.21**

As can be seen from Table 1, our proposed AdaDNNs is much better than the baseline DNNs. For example, for the measure of "C.R.W (upper)", AdaDNNs has a surprised improvement, i.e., from 64.90% to 80.55%. That is to say, the adaptive ensemble of DNNs in a simple but effective strategy can largely improved the performance from the original baseline DNNs. Moreover, compared with the latest top submissions (e.g., "Baidu IDL v3" and "HIK_OCR_v3"), our method, AdaDNNs Pruning, has the best performance with "C.R.W (upper)", i.e., 84.21%.

We also perform experiments on the COCO-text dataset (Veit et al. 2016), a similar challenging but large-scale incidental scene text dataset. Images in this dataset are from the MS COCO dataset that contain text ($63,686$ images with $173,589$ text regions). ICDAR2017 Robust Reading Challenge on COCO-Text is holding and will be released in ICDAR 2017. So, the comparative results of AdaDNNs, AdaDNNs Pruning and the baseline DNNs are only on the validation set; they are 58.08%, 66.07%, and 66.27%, respectively. Some scene text recognition samples for COCO-text are shown in Fig. 3.

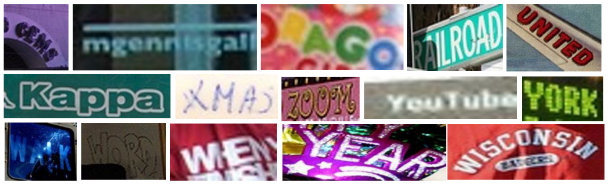

Figure 3: Challenging samples of scene text from COCO-text which are correctly recognized (with C.R.W. upper) by AdaDNNs: "GEMS", "mgennisgal", "RGAO", "RAILROAD", "UNITED", "Kappa", "XMAS", "ZOOM", "YouTube", "YORK", "WALK", "WPRD", "WHEN", "YEAR", and "WISCONSIN".

Experiments with Focused Scene Text Recognition and Born-Digital Text Recognition

In order to investigate the generalization of AdaDNNs, we directly use the trained AdaDNNs system above (for 2015 ICDAR Challenge 4), and perform experiments on 2013 ICDAR Challenge 2 (cropped word recognition) dataset. The Challenge 2 dataset contains $1,015$ ground truths cropped word images. In our experiments, a variety of methods are conducted and compared, i.e., the baseline DNNs, AdaDNNs, AdaDNNs pruning, the winning participation method in the official competition (marked as **bold words**), the top three results in published papers, and the latest top submissions of the RRC website in 2017 (marked as *italic words*).

Table 2: Comparative results on 2013 ICDAR Challenge 2 dataset (focused scene text recognition), where the comparative results without publications are from the RRC website.

Date	Method	T.E.D	C.R.W	T.E.D.	C.R.W.
			(%)	(upper)	(upper)
2017/8/14	*TencentAILab*	**42**	**95.07**	**39.35**	**95.34**
2017/7/28	*Tencent Youtu*	48.12	92.42	40.37	93.42
2017/2/24	*HIK_OCR*	64.95	90.78	42.31	93.33
2016	CNN (Jaderberg et al. 2016)	–	–	–	90.8
2016	RARE (Shi et al. 2016)	–	–	–	88.6
2017	CRNN (Shi, Bai, and Yao 2017)	–	–	–	89.6
2013/4/6	**PhotoOCR**	122.75	82.83	109.9	85.30
-	Baseline DNNs	306.43	75.34	282.35	78.63
-	AdaDNNs	193.51	83.20	170.13	86.67
-	AdaDNNs Pruning	182.7	85.21	164.38	88.13

Similarly, AdaDNNs is much better than the baseline DNNs, e.g., the measure of "C.R.W (upper)" increases from 78.63% to 86.67%. Surprisedly, only trained for another task (Challenge 4), the AdaDNNs (AdaDNNs Pruning) has a competitive performance on a new dataset (Challenge 2 dataset), compared with the recent published methods (e.g., CRNN (Shi, Bai, and Yao 2017)), and even with the latest submission results.

Apart from the above experiments on text recognition from scene images (ICDAR Robust Reading Competition Challenge 2 and 4), we also directly perform the learned

[6] http://rrc.cvc.uab.es.

AdaDNNs on the born-digital images track (Challenge 1). Though born-digital images are not scene images, they have similar challenging issues for text recognition, e.g., complex backgrounds, low resolution and various colors. We also compare AdaDNNs (AdaDNNs Pruning) with the baseline DNNs, the winning participation method in the official competition (marked as **bold words**), and the latest top submissions of the RRC website (marked as *italic words*). The similar conclusions are drawn. Firstly, AdaDNNs improves largely compared with the baseline DNNs (from 84.50 to 92.22 for "C.R.W (upper)"). Secondly, AdaDNNs has a comparative performance with the latest submission results (e.g., "Dahua_OCR_v1" with 92.49% in 2017/9/1).

Table 3: Comparative results on ICDAR Challenge 1 dataset (born-digital text recognition), where the comparative results are from the RRC website.

Date	Method	T.E.D	C.R.W	T.E.D.	C.R.W.
			(%)	(upper)	(upper)
2017/8/22	*Tecent Youtu*	**17.51**	**96.80**	13.67	**97.29**
2017/7/21	*TecentAILab*	18.77	96.18	**12.91**	97.22
2017/9/1	*Dahua_OCR_v1*	57.47	91.31	42.87	92.49
2013/4/6	**PhotoOCR**	103.41	82.21	87.19	85.41
-	Baseline DNNs	87.7	82.42	72.32	84.50
-	AdaDNNs	55.44	89.58	39.61	92.22
-	AdaDNNs Pruning	55.64	89.53	39.81	92.17

We fully believe that if AdaDNNs (AdaDNNs Pruning) performs re-training on ICDAR Challenge 1 and Challenge 2 datasets, the performance will correspondingly be improved and obtain a more impressive results compared with the latest submission systems. This is also a near issue for our future work.

Conclusion and Discussion

A variety of DNNs based methods have been proposed and are still being investigated in the literature for scene text recognition because of the grand challenges, e.g., complex backgrounds, various illuminations and diverse distortions. In order to fully take advantage of the complementary diversity and the high accuracy of neural network components in DNNs, an adaptive ensemble of deep neural networks (AdaDNNs) is proposed to simply select and adaptively combine neural networks in the whole training procedure. Comparative experiments of scene text (cropped word) recognition showed that AdaDNNs achieves a remarkable increase in the final performance (more than 10%) compared with the baseline DNNs.

Note that the DNNs methods have dramatically improved the state-of-the-art in object detection, object recognition, speech recognition and many other domains. Consequently, a near future issue is to evaluate the efficacy of AdaDNNs with state-of-the-art DNNs on object recognition and speech recognition. For example, experiments for object detection and recognition of AdaDNNs with Snapshot Ensembling (Huang et al. 2017), ResNet (He et al. 2016), and DenseNet (Huang, Liu, and Weinberger 2017) can be performed and compared in the next step.

References

[Almazan et al. 2014] Almazan, J.; Gordo, A.; Fornes, A.; and Valveny, E. 2014. Word spotting and recognition with embedded attributes. *IEEE Trans. Pattern Analysis and Machine Intelligence* 36(12):2552–2566.

[Alsharif and Pineau 2014] Alsharif, O., and Pineau, J. 2014. End-to-end text recognition with hybrid HMM maxout models. In *Proceedings of International Conference on Learning Representations (ICLR'14)*.

[Bissacco et al. 2013] Bissacco, A.; Cummins, M.; Netzer, Y.; and Neven, H. 2013. PhotoOCR: Reading text in uncontrolled conditions. In *Proceedings of International Conference on Computer Vision (ICCV'13)*, 4321–4328.

[Bottou, Curtis, and Nocedal 2016] Bottou, L.; Curtis, F. E.; and Nocedal, J. 2016. Optimization methods for large-scale machine learning. *CoRR* abs/1606.04838.

[Bottou 2010] Bottou, L. 2010. Large-scale machine learning with stochastic gradient descent. In *Proceedings of the 19th International Conference on Computational Statistics (COMPSTAT'10)*, 177–186.

[Breiman 1996] Breiman, L. 1996. Bagging predictors. *Machine Learning* 24:122–140.

[Dauphin et al. 2014] Dauphin, Y. N.; Pascanu, R.; Gülçehre, Ç.; Cho, K.; Ganguli, S.; and Bengio, Y. 2014. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014 (NIPS'14)*, 2933–2941.

[Fragoso et al. 2011] Fragoso, V.; Gauglitz, S.; Zamora, S.; Kleban, J.; and Turk, M. 2011. Translatar: A mobile augmented reality translator. In *Proceedings of 2011 IEEE Workshop on Applications of Computer Vision (WACV'11)*, 497–502.

[Freund and Schapire 1997] Freund, Y., and Schapire, R. 1997. A decision-theoretic generalization of on-line learning and an application to Boosting. *Journal of Computer and System Sciences* 55(1):119–139.

[Gordo 2015] Gordo, A. 2015. Supervised mid-level features for word image representation. In *Proceedings of 2015 IEEE International Conference on Computer Vision and Pattern Recognition (CVPR'15)*, 2956–2964.

[Goto and Tanaka 2009] Goto, H., and Tanaka, M. 2009. Text-tracking wearable camera system for the blind. In *Proceedings of International Conference on Document Analysis and Recognition (ICDAR'09)*, 141–145.

[Graves et al. 2006] Graves, A.; Fernández, S.; Gomez, F. J.; and Schmidhuber, J. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006*, 369–376.

[He et al. 2016] He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'16)*, 770–778.

[Huang et al. 2017] Huang, G.; Li, Y.; Pleiss, G.; Liu, Z.; Hopcroft, J. E.; and Weinberger, K. Q. 2017. Snapshot ensembles: Train 1, get M for free. In *Proceedings of International Conference on Learning Representations (ICLR'17)*.

[Huang, Liu, and Weinberger 2017] Huang, G.; Liu, Z.; and Weinberger, K. Q. 2017. Densely connected convolutional networks. In *Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'17)*, 4700–4708.

[Jaderberg et al. 2014] Jaderberg, M.; Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2014. Synthetic data and artificial neural networks for natural scene text recognition. *CoRR* abs/1406.2227.

[Jaderberg et al. 2016] Jaderberg, M.; Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2016. Reading text in the wild with convolutional neural networks. *International Journal of Computer Vision* 116:1 – 20.

[Jaderberg, Vedaldi, and Zisserman 2014] Jaderberg, M.; Vedaldi, A.; and Zisserman, A. 2014. Deep features for text spotting. In *Proceedings of the 13th European Conference on Computer Vision (ECCV'14)*, 512–528.

[Karatzas et al. 2013] Karatzas, D.; Shafait, F.; Uchida, S.; Iwamura, M.; i Bigorda, L. G.; Mestre, S. R.; Mas, J.; Mota, D. F.; Almazán, J.; and de las Heras, L. 2013. ICDAR 2013 robust reading competition. In *Proceedings of 12th International Conference on Document Analysis and Recognition (ICDAR'13)*, 1484–1493.

[Karatzas et al. 2015] Karatzas, D.; Gomez-Bigorda, L.; Nicolaou, A.; Ghosh, S.; Bagdanov, A.; Iwamura, M.; Matas, J.; Neumann, L.; Chandrasekhar, V. R.; Lu, S.; Shafait, F.; Uchida, S.; and Valveny, E. 2015. ICDAR 2015 competition on robust reading. In *Proceedings of 13th International Conference on Document Analysis and Recognition (ICDAR'15)*, 1156–1160.

[Kawaguchi 2016] Kawaguchi, K. 2016. Deep learning without poor local minima. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016 (NIPS'16)*, 586–594.

[Keskar et al. 2017] Keskar, N. S.; Mudigere, D.; Nocedal, J.; Smelyanskiy, M.; and Tang, P. T. P. 2017. On large-batch training for deep learning: Generalization gap and sharp minima. In *Proceedings of International Conference on Learning Representations (ICLR'17)*.

[Létourneau et al. 2003] Létourneau, D.; Michaud, F.; Valin, J.-M.; and Proulx, C. 2003. Textual message read by a mobile robot. In *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'03)*, volume 3, 2724–2729.

[Manmatha, Han, and Riseman 1996] Manmatha, R.; Han, C.; and Riseman, E. M. 1996. Word spotting: A new approach to indexing handwriting. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR'96)*, 631–637.

[Minetto et al. 2011] Minetto, R.; Thome, N.; Cord, M.; Leite, N. J.; and Stolfi, J. 2011. Snoopertrack: Text detection and tracking for outdoor videos. In *Proceedings of the 18th IEEE International Conference on Image Processing (ICIP'11)*, 505–508.

[Mishra, Alahari, and Jawahar 2012] Mishra, A.; Alahari, K.; and Jawahar, C. V. 2012. Top-down and bottom-up cues for scene text recognition. In *Proceedings of 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'12)*, 2687–2694.

[Rodriguez, Kuncheva, and Alonso 2006] Rodriguez, J. J.; Kuncheva, L. I.; and Alonso, C. J. 2006. Rotation Forest: A new classifier ensemble method. *IEEE Trans. Pattern Analysis Machine Intelligence* 28(10):1619–1630.

[Sanketi, Shen, and Coughlan 2011] Sanketi, P.; Shen, H.; and Coughlan, J. M. 2011. Localizing blurry and low-resolution text in natural images. In *Proceedings of 2011 IEEE Workshop on Applications of Computer Vision (WACV'11)*, 503–510.

[Shahab, Shafait, and Dengel 2011] Shahab, A.; Shafait, F.; and Dengel, A. 2011. ICDAR 2011 robust reading competition challenge 2: Reading text in scene images. In *Proceedings of International Conference on Document Analysis and Recognition (ICDAR'11)*, 1491–1496.

[Shi and Xu 2005] Shi, X., and Xu, Y. 2005. A wearable translation robot. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA'05)*, 4400–4405.

[Shi, Bai, and Yao 2017] Shi, B.; Bai, X.; and Yao, C. 2017. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*. published online.

[Shi et al. 2013] Shi, C.; Wang, C.; Xiao, B.; Zhang, Y.; Gao, S.; and Zhang, Z. 2013. Scene text recognition using part-based tree-structured character detection. In *Proceedings of 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'13)*, 2961–2968.

[Shi et al. 2016] Shi, B.; Wang, X.; Lyu, P.; Yao, C.; and Bai, X. 2016. Robust scene text recognition with automatic rectification. In *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'16)*, 4168–4176.

[Tian et al. 2017] Tian, S.; Yin, X.-C.; Su, Y.; and Hao, H.-W. 2017. A unified framework for tracking based text detection and recognition from web videos. *IEEE Trans. Pattern Analysis and Machine Intelligence*. published online.

[Veit et al. 2016] Veit, A.; Matera, T.; Neumann, L.; Matas, J.; and Belongie, S. J. 2016. Coco-text: Dataset and benchmark for text detection and recognition in natural images. *CoRR* abs/1601.07140.

[Wang and Belongie 2010] Wang, K., and Belongie, S. 2010. Word spotting in the wild. In *Proceedings of European Conference on Computer Vision (ECCV'10)*, 591–604.

[Weinman et al. 2014] Weinman, J. J.; Butler, Z.; Knoll, D.; and Feild, J. 2014. Toward integrated scene text reading. *IEEE Trans. Pattern Analysis and Machine Intelligence* 36(2):375–387.

[Wu, Chen, and Yang 2005] Wu, W.; Chen, X.; and Yang, J. 2005. Detection of text on road signs from video. *IEEE Trans. Intelligent Transportation Systems* 6(4):378–390.

[Ye and Doermann 2015] Ye, Q., and Doermann, D. 2015. Text detection and recognition in imagery: A survey. *IEEE Trans. Pattern Analysis and Machine Intelligence* 37(7):1480–1500.

[Yin et al. 2014] Yin, X.-C.; Huang, K.; Yang, C.; and Hao, H.-W. 2014. Convex ensemble learning with sparsity and diversity. *Information Fusion* 20:49–59.

[Yin et al. 2016] Yin, X.-C.; Zuo, Z.-Y.; Tian, S.; and Liu, C.-L. 2016. Text detection, tracking and recognition in video: A comprehensive survey. *IEEE Trans. Image Processing* 25(6):2752–2773.

[Zhou 2012] Zhou, Z.-H. 2012. *Ensemble Methods: Foundations and Algorithms*. Boca Raton, FL: Chamman & Hall/CRC.