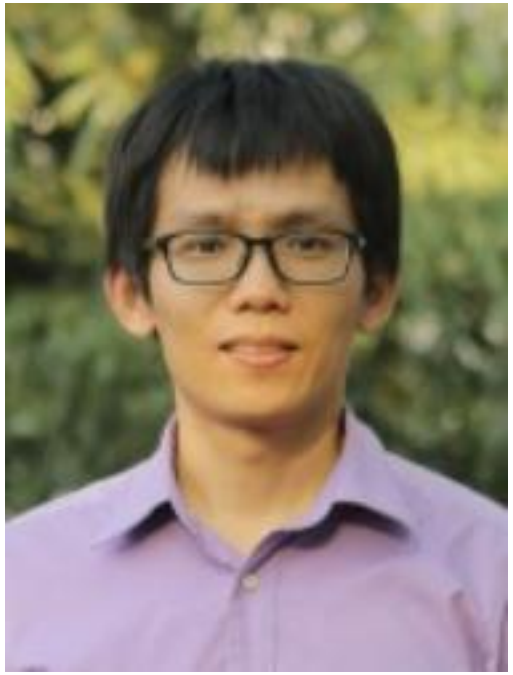
Sequence-to-Sequence Domain Adaptation Network for Robust Text Image Recognition

Yaping Zhang[1,2], Shuai Nie[1], Wenju Liu[1]*, Xing Xu[3,5], Dongxiang Zhang[4,5], Heng Tao Shen[3]

[1]National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences (CASIA)
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences (UCAS)
[3]Center for Future Media & School of Computer Science and Engineering, University of Electronic Science and Technology of China
[4]College of Computer Science and Technology, Zhejiang University, China [5]Afanti AI Lab, China
{yaping.zhang, shuai.nie, lwj}@nlpr.ia.ac.cn, xing.xu@uestc.edu.cn, zhangdongxiang37@gmail.com, shenhengtao@hotmail.com

Authors



张东祥，2012年新加坡国立大学博士毕业，现为浙江大学“百人计划”研究员，博士生导师，CCF数据库专业委员会委员。2016-2018年任电子科技大学特聘教授。目前已发表60余篇高质量论文，其中40余篇为CCFA类论文，1篇ESI高被引论文。

作为在线教育行业第一家吃螃蟹的公司，阿凡题于2016年8月率先成立了人工智能研究院，寻找撬动教育这门生意杠杆的新支点——人工智能(AI)的赋能。过去一年多的时间里，阿凡题已经成长为全球在图像识别、人工智能、大数据领域遥遥领先的教育科技企业。

About domain adaption

▶ Domain adaptation has shown promising advances for alleviating domain shift problem.

▶ It remains challenging to build a robust text recognizer that can handle varying data in new scenarios effectively, due to the inevitable domain shift when the actual data is encountered at “test time”.

▶ To build a robust text recognizer for the shifted target text image, a general solution is to collect large scale annotated text images, but they are high-cost and cannot cover all diversities.

Recent works' disadvantages

▶ However, recent visual domain adaptation works usually focus on non-sequential object recognition with a global coarse alignment.

▶ Inadequate to transfer effective knowledge for sequence like text images with variable-length fine-grained character information.

▶ The most popular domain adaptation methods cannot be directly applied to the sequence prediction, since a global fixed-length representation lacks important fine-grained information at the character level.

Introduction

▶In this paper, we develop a Sequence-to-Sequence Domain Adaptation Network (SSDAN) for robust text image recognition, towards real world applications in various recognition scenarios, including the natural scene text, handwritten text and even mathematical expression recognition.

Figure 1. Examples of different types of domain shift in text image recognition scenarios.

Contributions

- We propose a novel Sequence-to-sequence Domain Adaptation Network dubbed SSDAN for robust text image recognition, which could be generalized to different scenes, such as natural scene text, handwritten text and mathematical expression recognition.

- We introduce a novel GAS unit in SSDAN to bridge the sequence-like text image recognition and domain adaptation, which could adaptively transfer fine-grained character-level knowledge instead of performing domain adaptation by global features.

- The proposed SSDAN is capable of using unsupervised sequence data to reduce domain shift effectively.

Related work

▶ For example, *[Baoguang Shi et al., 2016]* and *[Wei Liu et al. 2016]* introduce a spatial transformer network to rectify the entire text before recognition. Furthermore, CharNet *[Wei Liu et al. 2018]* tried to introduce a character-level spatial transformer to rectify individual characters.

▶ There have been a plethora of recent works in the field of visual domain adaptation addressing the domain shift problem.They generally optimize the global representation via minimizing some measure of domain shift, such as MMD, CORAL, or adversarial loss.

▶ However, they were only designed for spatial affine distortions and hard to generalize to the distortion caused by handwriting styles or various structures in mathematical expressions.

▶ Existing text image recognition methods are usually designed for a specific scenario, and cannot be generalized effectively to different tasks.

▶ Therefore, these methods cannot be directly applied on sequential text images with multiple characters, as the domain shift are locally in the characters rather than the global image.

Related work

▶ Recently, other methods have been proposed to adapt the different font styles for image-to-image translation via adversarial learning *[Samaneh et al. 2018 CVPR]*

▶ Similarly, these methods limitedly translate the font in different style of signal characters on a global image, which are still cannot be extended to text-line images.

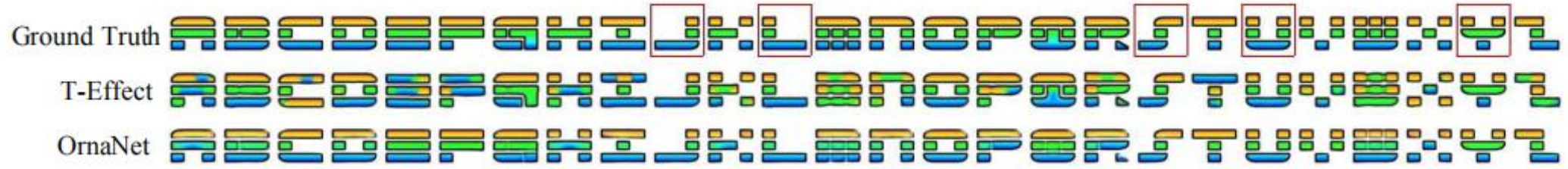


Figure 4: Text Effect Transfer [34] failure example on clean input glyphs.

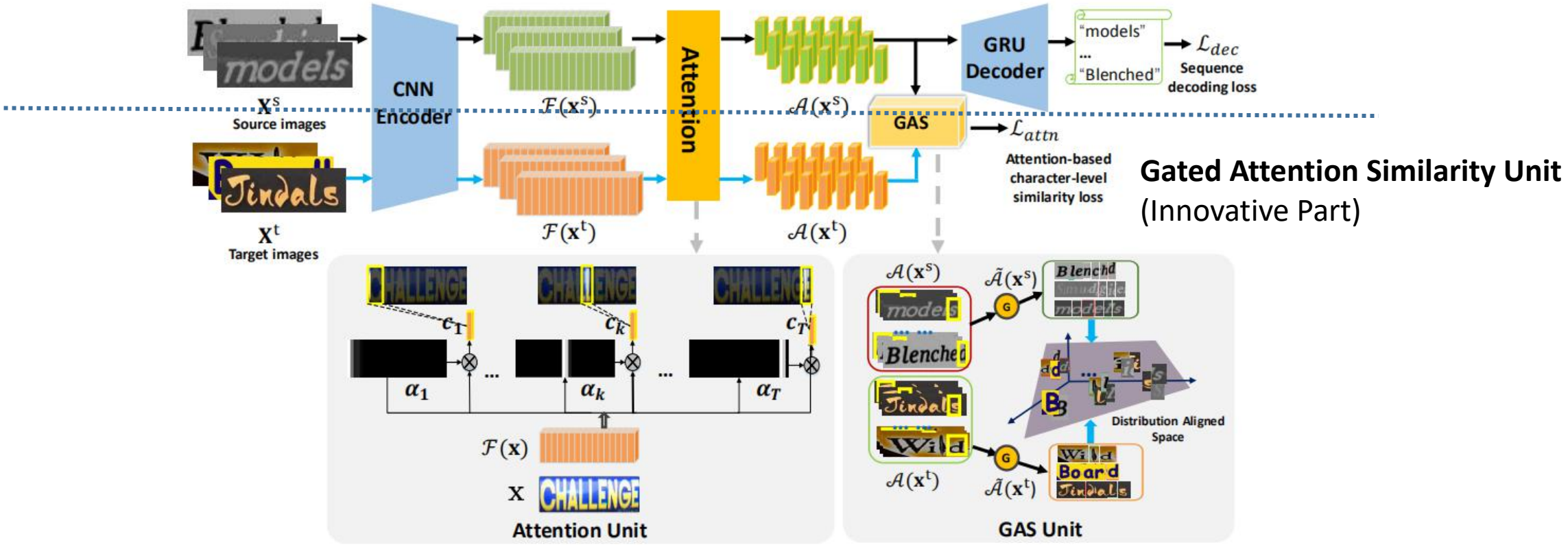
Figure 5: Failure cases on clean input glyphs.

- Source and Target

In this paper, unsupervised sequence-to-sequence domain adaptation is developed for robust text recognition. Specifically, the source domain text images with well-annotated text labels (a sequence of characters or symbols) are available, while we only have an access to unlabeled text images in target domain, which is in a different distribution. More formally, we assume that there are N^s annotated source domain samples $X^s = \{\mathbf{x}_i^s\}_{i=0}^{N^s}$ with the corresponding labels $\mathcal{Y}^s = \{\mathbf{y}_i^s\}_{i=0}^{N^s}$, and N^t unlabeled target-domain samples $X^t = \{\mathbf{x}_i^t\}_{i=0}^{N^t}$ without any available annotated labels in the training time. For $\mathbf{y} \in \mathcal{Y}^s$, $\mathbf{y} = \{y_1, y_2, ..., y_T\}$, where y_k and T denotes a character label and the variable length of text, respectively.

Network Structure(Two parts)

Attentive Text Recognition (Traditional Part)



Gated Attention Similarity Unit (Innovative Part)

Figure 2. The structure of SSDAN consists of: a CNN encoder to map the input images into a sequence of high-level feature vectors , an attention unit between the encoder and decoder to adaptively focus on the location of character, a GRU decoder to convert encoded features into output strings recurrently, and a GAS unit to offer the guidance for model to adaptively find character-level domain-invariant features between the source and target domain. Overall, the unsupervised sequence-to-sequence domain adaptation is achieved by jointly minimizing character-level similarity loss $\mathcal{L}_{attn}$ and source decoding loss $\mathcal{L}_{dec}$.

Attentive Text Recognition(Similar to WAP(*[Jun Du et al. 2017]*))

The attentive text recognition pipeline consists:

▶ a CNN encoder that learns high-level visual representations from an input image.

▶ an attention model between the encoder and the decoder driving the focus of attention of the model towards a specific part of the sequence of encoded features.

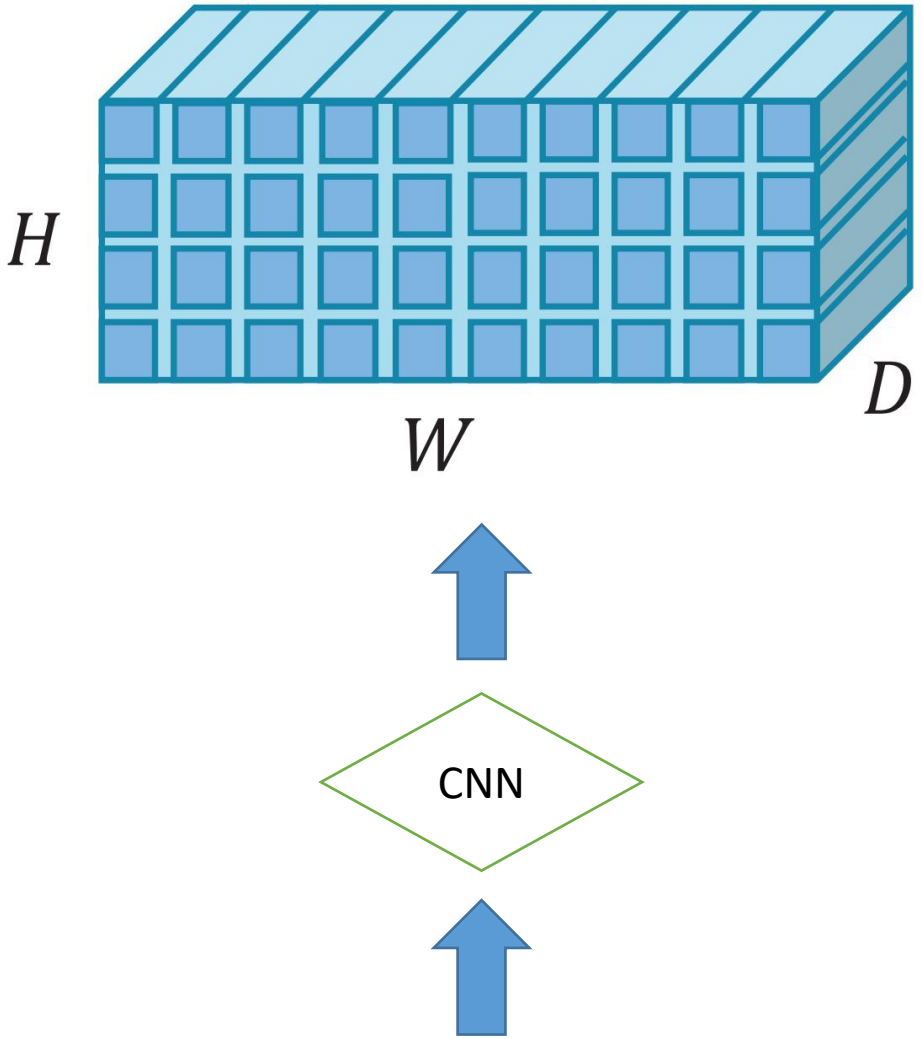
▶ a GRU decoder that generates a sequence of symbols as output, one at every time step.

CNN encoder

CNN Encoder. CNN encoder $\mathcal{F}$ takes the raw input image $\mathbf{x}$ from the source or target domain, and produces a feature grid $\mathcal{F}(\mathbf{x})$ of size $H' \times W' \times D$, where D denotes the number of channels, H' and W' are the resulted feature map height and width, respectively. The encoder output is then reshaped as a grid sequence of L elements, $L = H' \times W'$. Each of these elements is a D-dimensional feature vector that corresponds to a local region of the image through its corresponding receptive field. Hence, the whole encoded image $\mathcal{F}(\mathbf{x})$ could be reformatted as,

$$\mathcal{F}(\mathbf{x}) = [\mathbf{f}_1, ..., \mathbf{f}_L], \mathbf{f}_i \in R^D, \quad (1)$$

where $\mathbf{f}_i$ corresponds to i-th grid of the encoded image $\mathcal{F}(\mathbf{x})$, which preserves specific spatial information of the input image $\mathbf{x}$.



Attention

Attention. Although the CNN encoder keeps the spatial information, we cannot decide the location of a specific character in a text image. Therefore, an attention model is introduced to learn which part of the text image is the most relevant to a decoding character. As shown in Figure 2, the attention is a T-step process, at time-step k, the representation of the most relevant part to character y_k of encoding feature map $\mathcal{F}(\mathbf{x})$ is defined as a context vector $\mathbf{c}_k$:

$$\mathbf{c}_k = \sum_{i=0}^{L} \alpha_{k,i} \mathbf{f}_i, \tag{2}$$

where, the attention weights $\alpha_{k,i}$ is calculated by

$$\alpha_{k,i} = \frac{\exp(\mathbf{s}_{k,i})}{\sum_{j=0}^{L} \exp(\mathbf{s}_{k,j})}, \tag{3}$$

where the attention score $\mathbf{s}_{k,i}$ indicates the probability of that the model attends to the i-th sub-region in the encoded map $\mathcal{F}(\mathbf{x})$ when decoding the k-th character of the text image. Following the past empirical work [7], we defined the attention score as

$$\mathbf{s}_{k,i} = \beta^\top \tanh(\mathbf{W}_h \mathbf{h}_{k-1} + \mathbf{W}_f \mathbf{f}_i), \tag{4}$$

where β, $\mathbf{W}_h$ and $\mathbf{W}_f$ are the parameters to be learnt, $\mathbf{h}_{k-1}$ is the previous decoding state in the decoder.

From *[Bahdanau et al. 2015]* :

In a new model architecture, we define each conditional probability in Eq. (2) as:

$$p(y_i|y_1, \ldots, y_{i-1}, \mathbf{x}) = g(y_{i-1}, s_i, c_i), \tag{4}$$

where s_i is an RNN hidden state for time i, computed by

$$s_i = f(s_{i-1}, y_{i-1}, c_i).$$

The context vector c_i is, then, computed as a weighted sum of these annotations h_i:

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j. \tag{5}$$

The weight α_{ij} of each annotation h_j is computed by

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})},$$

where

$$e_{ij} = a(s_{i-1}, h_j)$$

is an *alignment model* which scores how well the inputs around position j and the output at position i match. The score is based on the RNN hidden state s_{i-1} (just before emitting y_i, Eq. (4)) and the j-th annotation h_j of the input sentence.

▶ From *[Bahdanau et al. 2015]* :

In a new model architecture, we define each conditional probability in Eq. (2) as:

$$p(y_i|y_1,\dots,y_{i-1},\mathbf{x}) = g(y_{i-1},s_i,c_i), \tag{4}$$

where s_i is an RNN hidden state for time i, computed by

$$s_i = f(s_{i-1},y_{i-1},c_i).$$

The context vector c_i is, then, computed as a weighted sum of these annotations h_i:

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j. \tag{5}$$

The weight α_{ij} of each annotation h_j is computed by

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x}\exp(e_{ik})},$$

where

$$e_{ij} = a(s_{i-1},h_j)$$

is an *alignment model* which scores how well the inputs around position j and the output at position i match. The score is based on the RNN hidden state s_{i-1} (just before emitting y_i, Eq. (4)) and the j-th annotation h_j of the input sentence.

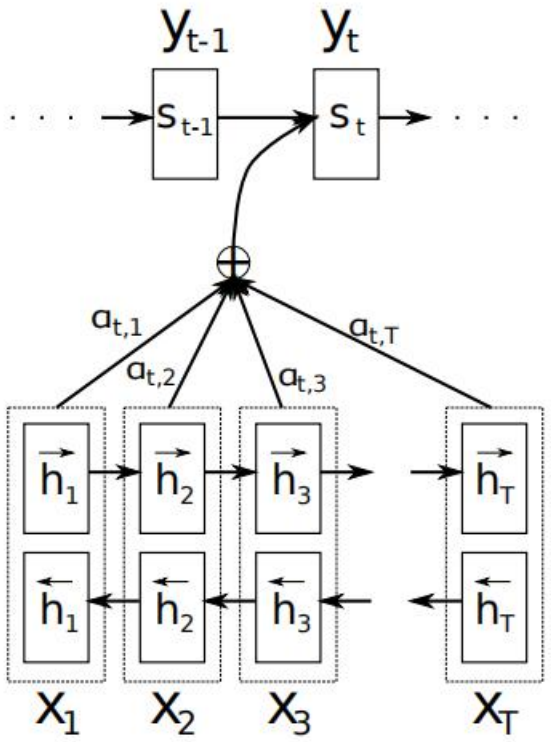


Figure 1: The graphical illustration of the proposed model trying to generate the t-th target word y_t given a source sentence $(x_1, x_2, \dots, x_T)$.

GRU decoder

GRU Decoder. A GRU decoder is employed to predict the string of an input text image recurrently, where we use gated recurrent unit (GRU) neural network. At decoding time step k, the GRU leverages the context vector $\mathbf{c}_k$, previous state $\mathbf{h}_{k-1}$ and previous predicted character y_{k-1} to generate a new hidden state

$$\mathbf{h}_k = GRU(\mathbf{h}_{k-1}, y_{k-1}, \mathbf{c}_k), \tag{5}$$

where, $\mathbf{c}_k$ is generated by the attention mechanism, which focuses on the most relevant region of current decoding character. Then, the probability of current predicted symbol y_k is computed by :

$$p(y_k|y_{k-1}, \mathbf{c}_k) = g\left(\mathbf{W}_o \tanh(\mathbf{E}\tilde{\mathbf{y}}_{k-1} + \mathbf{W}_d\mathbf{h}_k + \mathbf{W}_c\mathbf{c}_k)\right), \tag{6}$$

where g denotes a softmax activation function, $\mathbf{W}_o$, $\mathbf{W}_d$ and $\mathbf{W}_c$ are the mapping matrices, $\mathbf{E}$ is the embedding matrix, and $\tilde{\mathbf{y}}_{k-1}$ is the one-hot vector of character label y_{k-1}.

The probability of the sequential labels $\mathbf{y}$ is finally given by the product of the probability of each label:

$$P(\mathbf{y}|\mathcal{A}(\mathbf{x})) = \prod_{k=1}^{T} p(y_k|y_{k-1}, \mathbf{c}_k), \tag{7}$$

where $\mathcal{A}(\mathbf{x}) = \{\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_T\}$, which could be regarded as a sequence of attended character-level features from an input text image $\mathbf{x}$.

Gated Attention Similarity Unit

▶ Through attention mechanism, an input image x can be adaptively decomposed into a series of character-level feature set $A(x) = \{c_1, c_2, ..., c_T\}$, where c_k presents the feature of k-th character in the text image x.

Hence, we further introduce an adaption gate function $\delta(\mathbf{c}_k)$ to judge if a context vector $\mathbf{c}_k$ is attending to a valid character,

$$\delta(\mathbf{c}_k) = \begin{cases} 1 & \text{if } p\left(y_k | y_{k-1}, \mathbf{c}_k\right) > p_c \\ 0 & \text{if } p\left(y_k | y_{k-1}, \mathbf{c}_k\right) < p_c \end{cases}, \tag{8}$$

where p_c is a confidence threshold. Furthermore, a gate function set $\mathbf{G}$ is adaptively changed according to the specific input image $\mathbf{x}$, which is expressed as:

$$\mathbf{G}(\mathbf{x}) = \{\delta(\mathbf{c}_1), ..., \delta(\mathbf{c}_T)\}, \tag{9}$$

Through the gate function, we can update attention context vector set by adaptation gate function set $\mathbf{G}(\mathbf{x})$,

$$\tilde{A}(\mathbf{x}) = A(\mathbf{x}) \otimes \mathbf{G}(\mathbf{x}), \tag{10}$$

where $\otimes$ denotes element-wise product operator. Specifically, if $\mathbf{c}_k \times \delta(\mathbf{c}_k) = 0$, then current context vector $\mathbf{c}_k$ will not be added in a new attention context vector set.

CORAL Distance

A gated attention similarity loss $\mathcal{L}_{attn}$ is accordingly introduced to measure the distance on the valid attended character-level feature set of source and target domain as

$$\mathcal{L}_{attn} = E_{[\mathbf{x}^s \in \mathbf{X}^s, \mathbf{x}^t \in \mathbf{X}^t]} \left\{ dist\left(\tilde{\mathcal{A}}(\mathbf{x}^s), \tilde{\mathcal{A}}(\mathbf{x}^t)\right) \right\}. \quad (11)$$

$$dist(\mathcal{U}_s, \mathcal{U}_t) = \frac{1}{4d^2} ||cov(\mathcal{U}_s) - cov(\mathcal{U}_t)||_F^2, \quad (12)$$

where $\mathcal{U}_s = \{\mathbf{u}_i^s\}$, $\mathbf{u}^s \in \mathcal{R}^d$, $\mathcal{U}_t = \{\mathbf{u}_i^t\}$, $\mathbf{u}^t \in \mathcal{R}^d$, and $||\cdot||_F^2$ denotes the squared matrix Frobenius norm, $cov(\mathcal{U}_s)$ is the covariance matrix of samples $\mathcal{U}_s$, denoted by

$$cov(\mathcal{U}_s) = \frac{1}{N-1}\left(\mathcal{U}_s{}^\top \mathcal{U}_s - \frac{1}{N}(\mathbf{1}^\top \mathcal{U}_s)^\top (\mathbf{1}^\top \mathcal{U}_s)\right), \quad (13)$$

where $\mathbf{1}$ is a column vector with all elements equal to 1, N is the number of samples $\mathcal{U}_s$, and $\mathcal{U}_s(i,j)$ $(\mathcal{U}_t(i,j))$ indicates the j-th dimension of the i-th source (target) data example.

In our GAS unit, $\mathcal{U}_s$ and $\mathcal{U}_t$ are replaced by the valid attended character-level feature set $\tilde{\mathcal{A}}(\mathbf{x}^s)$ and $\tilde{\mathcal{A}}(\mathbf{x}^t)$, respectively. Note that $\tilde{\mathcal{A}}(\mathbf{x}^s)$ and $\tilde{\mathcal{A}}(\mathbf{x}^t)$ need to be reformatted as a matrix, respectively. Specifically, suppose $\tilde{\mathcal{A}}(\mathbf{x}^s)$ and $\tilde{\mathcal{A}}(\mathbf{x}^t)$ contain T_1 and T_2 elements, respectively. Then $\tilde{\mathcal{A}}(\mathbf{x}^s)$ and $\tilde{\mathcal{A}}(\mathbf{x}^t)$ could be reformatted as matrices with $T_1 \times D$ and $T_2 \times D$ elements, and their covariance matrices are with the same dimension $D \times D$.

There are three optional distance:

▶**MMD**, computing the norm of difference between two domain means.

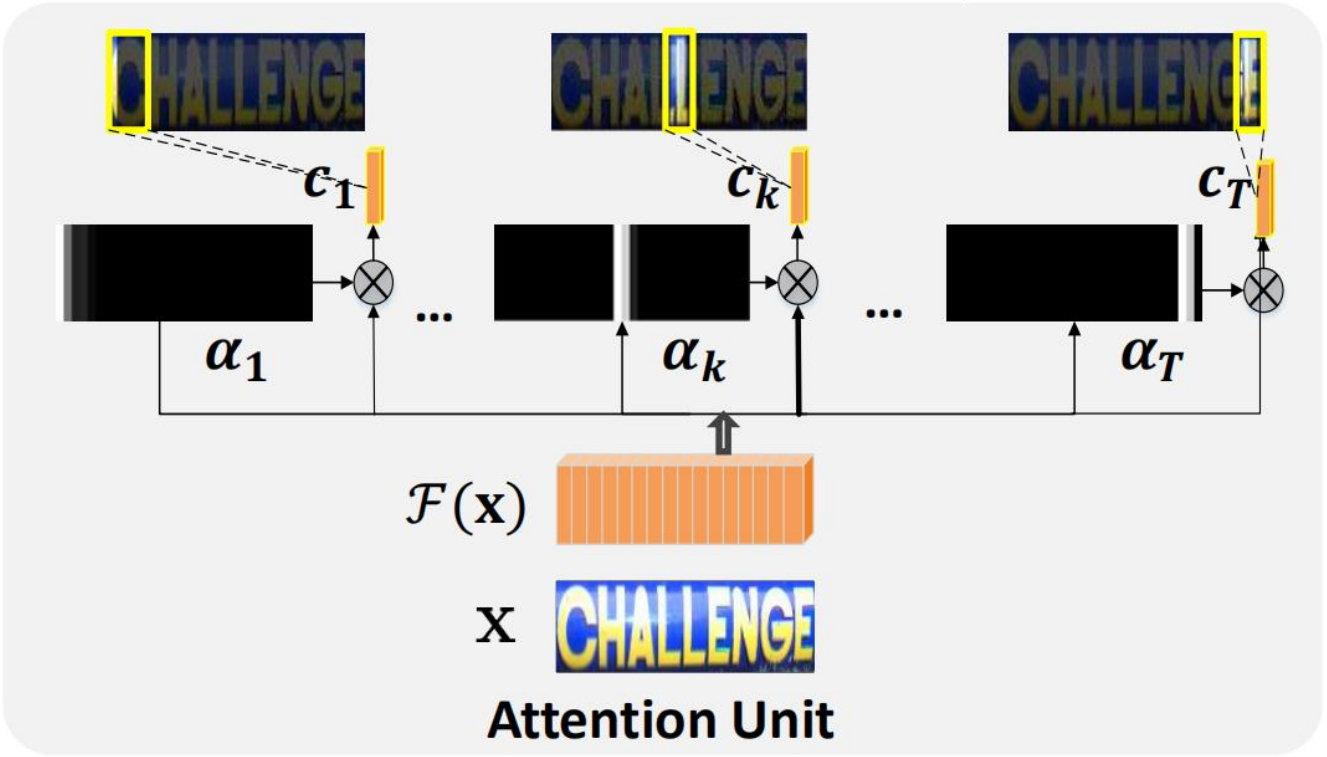
▶**CORAL**, computing the distance of covariance of two domain.

▶**adversarial loss**, minimizing the loss of a domain classifier to learn a representation that is simultaneously discriminative of source labels while not being able to distinguish between domains.

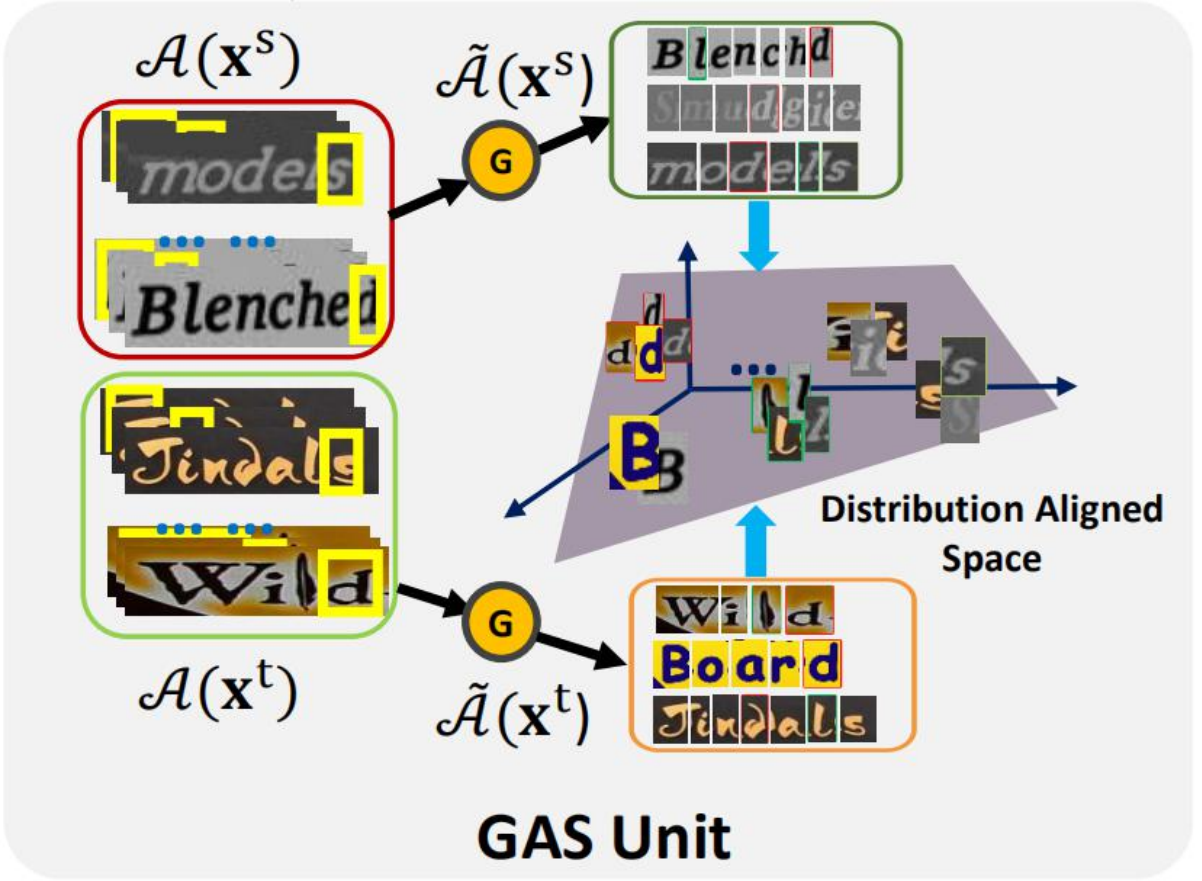
We choose CORAL experimentally.

CORAL Distance

Extract the character



Calculate the similarity



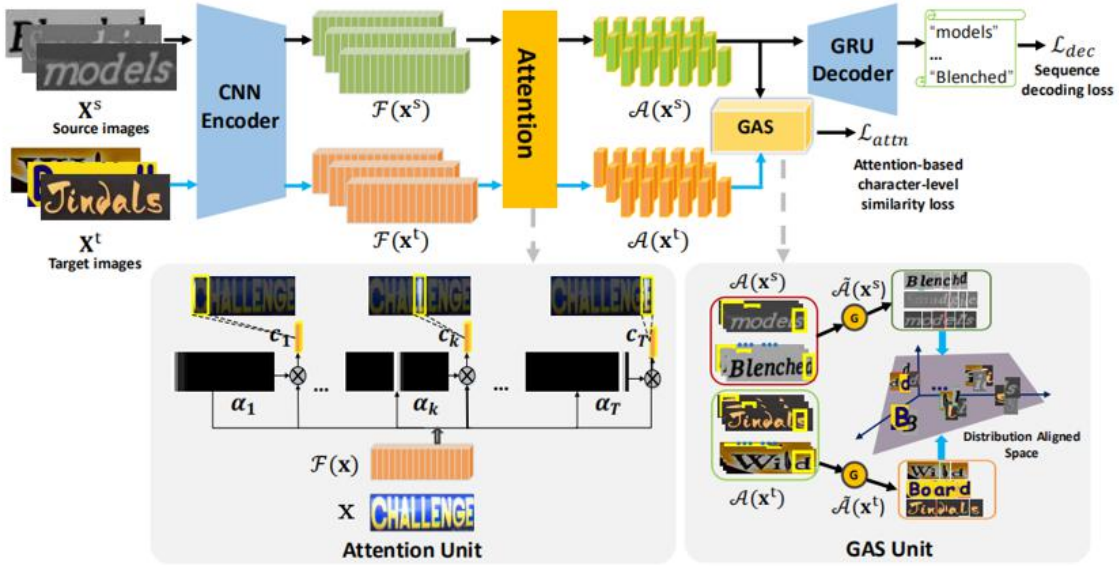
Overall Objective Function

With the well-annotated source-domain data, we could learn an optimized source text image recognizer by minimizing a supervised decoding loss, where we can use the negative log likelihood of sequential probability as the decoding loss $\mathcal{L}_{dec}$ to measure the differences between the predicted and the source labeled character sequences:

$$\mathcal{L}_{dec} = E_{(\mathbf{x}^s, \mathbf{y}^s) \sim (X^s, Y^s)} \{ -\log p(\mathbf{y}^s | \mathcal{A}(\mathbf{x}^s)) \} . \quad (14)$$

$$\mathcal{L}_{SSDAN} = \mathcal{L}_{dec} + \lambda \mathcal{L}_{attn}, \quad (15)$$

where λ is a hyper-parameter to balance two terms. The model parameters can be directly optimized by minimizing the overall objective through stochastic gradient descent optimization algorithms.



Experiments

- **ICDAR-2003 (IC-03)** [25] contains 860 cropped scene text images, following the protocol used in [31].
- **ICDAR-2013 (IC-13)** [18] contains 857 cropped scene images after filtering as did in IC-03.
- **Street View Text (SVT)** [37] consists of 647 test scene word images from Google Street View.
- **IIIT5K-words (IIIT5K)** [27] contains 3,000 cropped test scene text images from the Internet.
- **IAM** [26] is a handwritten English text dataset, written by 657 different writers. It is partitioned into writer-independent training, validation and test partitions of 6161, 976 and 2915 lines, respectively. That contains a total of 46945, 7554 and 20306 correctly segmented words in each partition.
- **CROHME 2014** [28] is a handwritten mathematical expression dataset. It contains 8836 training and 986 test math expressions. There are 101 math symbols. The handwritten expressions or LaTeX notations in the test set never appear in the train set.

Evaluation Metric. For different recognition task, we adopt different evaluation metric as follows:

- **Scene text.** The word prediction accuracy is used to evaluate scene text recognition model, following several benchmark [21, 31].
- **Handwritten text.** Two metrics are used to evaluate the handwritten text recognition model: the Character Error Rate (CER) and the Word Error Rate (WER) [3, 34]. CER is defined as the Levenstein distance between the predicted and real character sequence of the word. WER denotes the percentage of words improperly recognized. For CER and WER, small values indicate better performance.
- **Mathematical expression.** We use a global performance metric expression recognition rate (ExpRate) to denote the percentage of predicted formula sequences matching the real formula sequences [7].

Implementation details

The architecture of the CNN encoder is derived from the DenseNet, where dense blocks are densely concatenation of 1×1 convolution layers and 3×3 convolution layers. while the transition layers are composed of 1×1 convolution and 2×2 average pooling, and the channel 0.5 refers to the compression rate(Dropout).

All convolutions are followed by batch normalization layer and rectified linear unit (Relu) activation function.

We use the pooling layer with kernel size 2 × 1 to reduce feature dimension along the height axis only. As a result, the resolution of feature maps produced by encoder is H/32 × W/4, where the values of H and W are set according to the specific dataset.

After the CNN encoder, we use a bi-directional LSTM to capture more context information for attention, and each LSTM has 256 hidden units. For the decoder, we use a GRU cell with 512 memory blocks.

Results on Scene Text

▶Synthetic dataset MJSYNTH is used as the source training data, and the real scene text data is used as target test data.

▶MJSYNTH contains 8 millions annotated synthetic images, which are generated to simulate natural scene text images.

▶The following table presents the test results on four real scene text datasets.

SSDAN-base : omits the GAS unit to switch off the domain adaption process

Table 1. Scene text recognition accuracies on general scene text recognition benchmarks.

Model	IIIT5K	SVT	IC-03	IC-13
ANN [16]	–	71.7	89.6	81.8
STAR-Net [22]	83.3	83.6	89.9	89.1
R^2AM [20]	78.4	80.7	88.7	90.0
CRNN [31]	81.2	82.7	91.9	89.6
RARE [32]	81.9	81.9	90.1	88.6
Ghosh et al [12]	–	75.1	89.3	–
Gao et al [10]	81.8	82.7	89.2	88.0
ASTER [33]	83.2	**87.6**	**92.4**	89.7
Char-Net [21]	83.6	84.4	91.5	90.8
SSDAN-base	81.1	82.1	91.2	91.0
SSDAN	**83.8**	84.5	92.1	**91.8**

Results on Handwritten Text & Mathematical Expressions

SSDAN-base : omits the GAS unit to switch off the domain adaption process

▶The source and target data are the writer-independent training and test data, respectively.

▶In the experiment, the training expressions and the unseen test expressions are used as source and target data.

Table 2. Results on handwritten text.

Method	WER	CER	Average
bluche2015deep [2]	24.7	**7.3**	16.00
bluche2016joint [3]	24.6	7.9	16.25
sueiras2018offline [34]	23.8	8.8	16.30
SSDAN-base	23.9	9.2	16.55
SSDAN	**22.2**	8.5	**15.35**

Table 3. Results on handwritten mathematical expression.

Method	ExpRate
I [28]	37.2
VI [28]	25.7
VII [28]	26.1
WYCIWYS [8]	28.7
Le et al [19]	35.2
IM2TEX [7]	38.7
SSDAN-base	39.9
SSDAN	**41.6**

Comparison to Standard Domain Adaptation

▶Measuring similarity on CNN outputs directly can be treated as a STandard Domain Adaptation method (STDA), which lacks of fine-grained character-level information in a text image.

▶we have done experiment to compare the STDA with our SSDAN on IAM dataset.

SSDAN-base : omits the GAS unit to switch off the domain adaption process

Table 4. Comparison to standard domain adaptation.

Method	SSDAN-base	SSDAN	STDA
WER	23.9	**22.2**	25.0
CER	9.2	**8.5**	11.1

Experiment Results and Visualization

On IAM dataset:

Table 5. Component Analysis.

Components	Model	V1	V2	V3	V4	V5	V6
Encoder	VGG	✓	✓				
	ResNet			✓	✓		
	DenseNet					✓	✓
Adaptation	GAS		✓		✓		✓
Evaluation	WER	32.8	26.9	29.9	27.9	23.9	**22.2**
	CER	15.9	12.6	14.3	13.1	9.2	**8.5**

On IAM dataset:

Table 6. Effect of Different Domain Shift Measurement.

Method	CORAL	MMD	Adversarial
WER	**22.2**	22.7	24.6
CER	**8.5**	8.8	10.8

On MJSYNTH → SVT dataset:

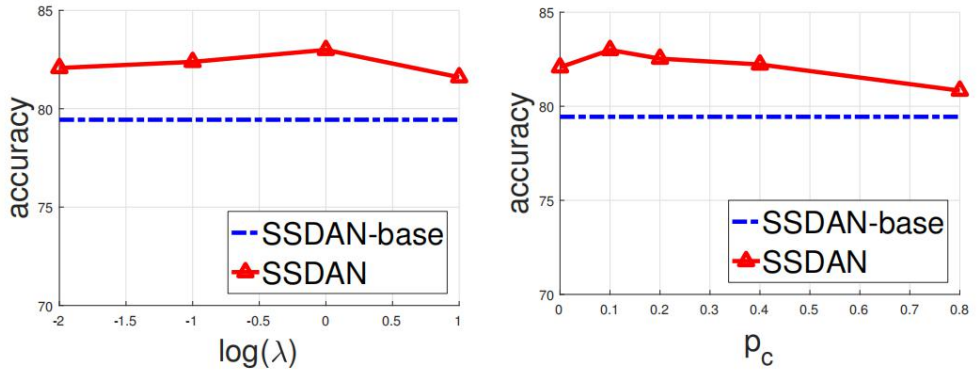


Figure 3. The effect of model parameters λ (left) and p_c (right).

Experiment Results and Visualization

On IAM dataset:

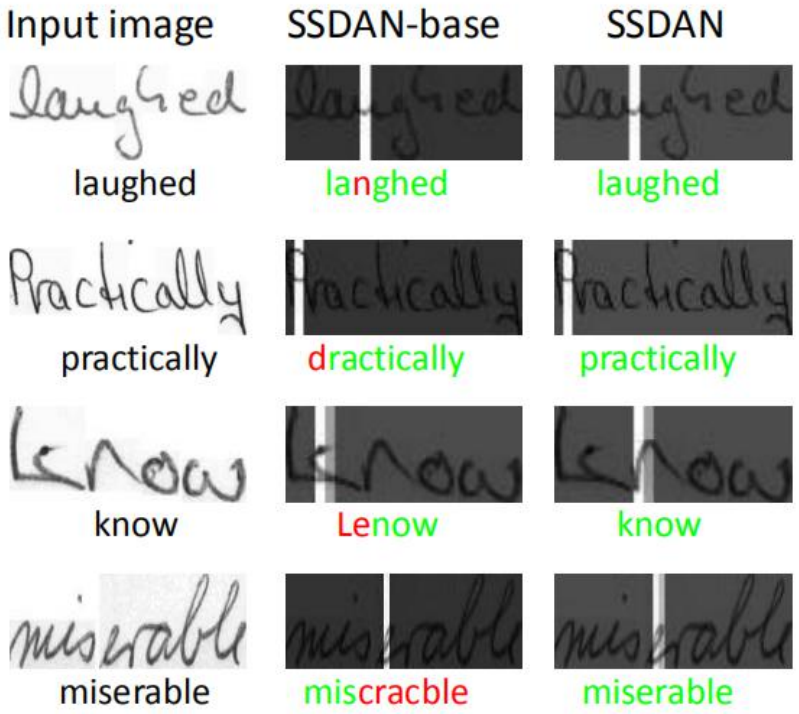


Figure 4. Examples showing the recognition result, the left column is the input image with ground truth, the second column and the last column denote the recognition result without and with domain adaptation, respectively. Each result is shown in the pair of attention visualization and prediction text.

On MJSYNTH dataset(The effect of unsupervised data):

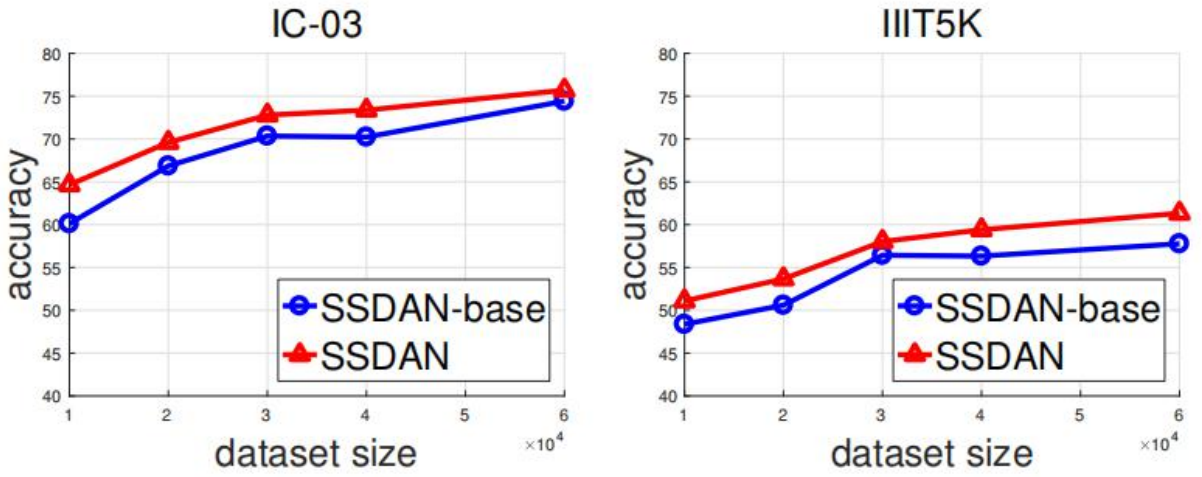


Figure 5. The effect of training dataset size on IC-03 (left) and IIIT5K (right).