Joint Visual Semantic Reasoning: Multi-Stage Decoder for Text Recognition

Ayan Kumar Bhunia\textsuperscript{1}  Aneeshan Sain\textsuperscript{1,2}  Amandeep Kumar*  Shuvozit Ghose*
Pinaki Nath Chowdhury\textsuperscript{1,2}  Yi-Zhe Song\textsuperscript{1,2}
\textsuperscript{1}SketchX, CVSSP, University of Surrey, United Kingdom. 
\textsuperscript{2}iFlyTek-Surrey Joint Research Centre on Artificial Intelligence.

\{a.bhunia, a.sain, p.chowdhury, y.song\}@surrey.ac.uk \{shuvozit.ghose, kumar.amandeep015\}@gmail.com.

Abstract

Although text recognition has significantly evolved over the years, state-of-the-art (SOTA) models still struggle in the wild scenarios due to complex backgrounds, varying fonts, uncontrolled illuminations, distortions and other artifacts. This is because such models solely depend on visual information for text recognition, thus lacking semantic reasoning capabilities. In this paper, we argue that semantic information offers a complimentary role in addition to visual only. More specifically, we additionally utilize semantic information by proposing a multi-stage multi-scale attentional decoder that performs joint visual-semantic reasoning. Our novelty lies in the intuition that for text recognition, prediction should be refined in a stage-wise manner. Therefore our key contribution is in designing a stage-wise unrolling attentional decoder where non-differentiability, invoked by discretely predicted character labels, needs to be bypassed for end-to-end training. While the first stage predicts using visual features, subsequent stages refine on-top of it using joint visual-semantic information. Additionally, we introduce multi-scale 2D attention along with dense and residual connections between different stages to deal with varying scales of character sizes, for better performance and faster convergence during training. Experimental results show our approach to outperform existing SOTA methods by a considerable margin.

1. Introduction

Text recognition has been a popular area of research \cite{1, 33, 53} for decades thanks to its wide range of commercial applications \cite{42}, from translation apps in mixed reality, street signs recognition in autonomous driving to assistive technology for the visually impaired \cite{10}, to name a few. Significant progress in fundamental deep learning components \cite{25, 2} alongside sequence-to-sequence learning frameworks \cite{53, 33, 46}, have boosted unconstrained word recognition accuracy (WRA) in recent times. Despite such developments, state-of-the-art text recognition frameworks \cite{3, 14, 37, 55, 8, 7, 9} still struggle in wild scenarios \cite{1, 42} due to complex backgrounds, varying fonts, uncontrolled illuminations, distortions and other artifacts. While machines struggle with a combination of these challenges, humans recognise them easily via joint visual-semantic reasoning. Therefore, the question in focus is – how to develop a visual-semantic reasoning skill for text recognition?

State-of-the-art text recognition systems \cite{1} mostly rely on extracted visual features to recognize a word image as a machine readable character-sequence. Follow-up efforts have been made towards improving reasoning ability by increasing the depth of convolutional feature extractor \cite{14} having larger receptive fields, or introducing pyramidal pooling \cite{55} and stacking multiple Bi-LSTM layers \cite{37}. Despite all these attempts that merely lead towards a better context modeling \cite{1}, a semantic reasoning potential \cite{13} is largely missing beyond enriching the visual feature. In wild scenarios, a word image might be blurred, distorted, partly

*Interned with SketchX
noisy or have artifacts, making recognition extremely difficult using visual feature alone. In such cases, we humans first try to interpret the easily recognizable characters using visual cues alone. A semantic reasoning skill is then applied to decode the final text by jointly processing the visual and semantic information from previously recognized character sequence. Motivated by this intuition, we propose a novel multi-stage prediction paradigm for text recognition. Here the first stage predicts using visual cues, while subsequent stages refine on top of it using joint visual-semantic information, by iteratively [13, 12] building up the estimates.

Designing this joint visual-semantic reasoning framework for text recognition is non-trivial. One might argue that attentional decoder being a sequence-to-sequence model, encapsulates the character dependency [53, 33, 46] and caters for semantic reasoning. However, due to its auto-regressive nature [2], only those characters predicted previously, could provide semantic context at a given step, thus making the semantic context flow unidirectional during inference. While semantic context becomes negligible towards the initial steps, one wrong prediction here would deal a cumulative adverse impact on the later steps (which stays unrefined due to single stage prediction). Therefore, this single stage attentional decoder fails to model the global semantic context, leaving joint visual-semantic reasoning unaccomplished. To explore the entire global semantic context, we need the completely unrolled prediction from first stage, upon which we can build up the global semantic information. Hence as our first contribution we propose a multi-stage attentional decoder (Figure 1), where we build up global semantic reasoning on the initial estimate of first stage, which is further refined by subsequent stages.

Let us consider the word ‘aeroplane’. For a single stage attentional decoder, if the model predicts ‘n’ instead of ‘r’, ‘aen’ would adversely affect rest of the prediction, without any chance of refinement (being single stage). Also, it holds almost negligible semantic context while predicting the first few characters. Considering we unroll the prediction stage-wise, if a character is predicted wrongly, like ‘aenoplane’, rest of the characters provide significant context as semantic information. This helps in refining ‘n’ to ‘r’ during the later stages coupled with visual information.

Moreover, obtaining the prediction from earlier stages, needs a non-differentiable argmax operation [26] as characters are discrete tokens. This leads to an inefficient modeling of influence of a prior stage on the next predictions. An apparent approach here might be to adapt teacher forcing [31] for the later stages during training. The later stages intend to learn how to refine the initial (might be incorrect) hypothesis towards a correct prediction. This motivation however is defeated on feeding exact ground-truth labels as teacher forcing for subsequent stages. Consequently, we make use of Gumbel-Softmax operation [26] bypassing non-differentiability, and making the network end-to-end trainable even across stages.

In summary our contributions are: First and foremost, we propose a multi-stage character decoding paradigm with stage-wise unrolling. While the first stage predicts using visual features, subsequent stages refine on-the-top of them using joint visual-semantic information. Secondly, we employ a Gumbel-softmax layer to make visual-to-semantic embedding layer differentiable. The model thus learns its refining strategy from initial to final prediction in an end-to-end manner. Thirdly, from the architectural design, we introduce multi-scale 2D attention to deal with varying scales of character size, and empirically found adding dense and residual connection between different stages stabilize training for better performance leading to outperforming other state-of-the-arts significantly on benchmark datasets.

2. Related Works

Text Recognition: While connectionist temporal classification (CTC) layer [18] does not model dependency in the output character space [52], an attention based decoder [53] encases language modeling, weakly supervised character detection and character recognition in a single paradigm. Following some seminal works [53, 32], attention based decoder became state-of-the-art pipeline for text recognition which includes four successive modules: i) a rectification network [53] to simplify irregular text image, ii) convolutional encoder for feature extraction, iii) Bi-LSTM layer for context modeling, and iv) an attentional decoder predicting the characters autoregressively.

Furthermore, the motivation of recent followed-up works can broadly be classified into following directions: (i) improve rectification network by introducing iterative [6] pipeline [63] and modelling geometrical attributes [59] of text image; (ii) four directional feature encoder [15] for better convolutional feature extraction; (iii) improving attention mechanism by extending to 2-D attention [33] and hard character localized annotation [14, 35], to better guide the attention based character alignment process. (iv) Recently, stacking multiple Bi-LSTM layers [37] and pyramidal pooling [55] on convolutional feature were employed towards the goal of better context modeling. These approaches however mainly focus on exploiting visual features, via different architectural modifications [58, 62] on top of Shi et al. [53], but mostly lacking in any semantic reasoning capabilities. Although some works claim to model semantic reasoning by stacking additional Bi-LSTM layers [37, 55], it only helps in modelling better contextual information without having actual reasoning potential. In this context, word-embeddings [46] from pre-trained language model were used to initialize the hidden state of attentional decoder, however we are skeptical towards this. For e.g. two related words “Chair” and “Table” may lie close in word-
We adopt ResNet from [53] as a backbone convolutional network to extract visual features from input image. To deal with characters of varying scales, we extend to multi-scale architecture for text recognition, with the help of Feature Pyramid Networks [36] which makes every resolution level semantically strong using lateral connections. Let a feature-map from particular scale be represented as $B_l \in \mathbb{R}^{H_l \times W_l \times D}$, where $l = L$ denotes deepest residual block having lowest resolution but highest level semantics. $H_l$ and $W_l$ are the height and width of the feature map from respective scales which depend on the accumulated strides of successive pooling layers, with all scales having $D$ channels uniformly [36]. To balance between computational ease and performance gain, we consider $l = \{L, L-1, L-2\}$ through empirical validation. Visual features have two components, (i) multi-scale feature-maps $\{B_L, B_{L-1}, B_{L-2}\}$ which acts as context for 2D attention in the later decoding process. (ii) The holistic feature $I_{BL}$, used to initialize the initial state of first stage decoder. This $I_{BL}$ is recognised as the final hidden state of a 2-layer Bi-LSTM which takes in a sequential feature ($W_L \times D$), obtained from column-wise max-pooling on feature-map $B_L$ from deepest residual block (ensuring height stays unity), followed by reshaping.

3. Methodology

Overview: Given an input word image $I$, we intend to predict the character sequence $Y = \{y_1, y_2, ..., y_T\}$, where $T$ denotes the variable length of text. Our framework is two-fold: (i) a visual feature extractor extracts context-rich holistic feature and multi-scale feature maps. (ii) Following that, a multi-stage attentional decoder builds up the character sequence estimates, in a stage-wise successive manner. While dealing with irregular/curved word images [59], [15], image rectification based approaches [59] often fall short [15, 35]. To do away with the burden of adding a separate sophisticated rectification network entirely, we follow a 2D attention mechanism [33] that helps to localize individual character in a weakly-supervised manner during decoding.

3.1. Visual Feature Extraction

We adopt ResNet from [53] as a backbone convolutional network to extract visual features from input image. To deal with characters of varying scales, we extend to multi-scale architecture for text recognition, with the help of Feature Pyramid Networks [36] which makes every resolution level semantically strong using lateral connections. Let a feature-map from particular scale be represented as $B_l \in \mathbb{R}^{H_l \times W_l \times D}$, where $l = L$ denotes deepest residual block having lowest resolution but highest level semantics. $H_l$ and $W_l$ are the height and width of the feature map from respective scales which depend on the accumulated strides of successive pooling layers, with all scales having $D$ channels uniformly [36]. To balance between computational ease and performance gain, we consider $l = \{L, L-1, L-2\}$ through empirical validation. Visual features have two components, (i) multi-scale feature-maps $\{B_L, B_{L-1}, B_{L-2}\}$ which acts as context for 2D attention in the later decoding process. (ii) The holistic feature $I_{BL}$, used to initialize the initial state of first stage decoder. This $I_{BL}$ is recognised as the final hidden state of a 2-layer Bi-LSTM which takes in a sequential feature ($W_L \times D$), obtained from column-wise max-pooling on feature-map $B_L$ from deepest residual block (ensuring height stays unity), followed by reshaping.

3.2. Joint Visual-Semantic Reasoning Decoder

Overview: Let the prediction from $s^{th}$ stage decoder be denoted as $Y^s = \{y^s_1, y^s_2, ..., y^s_T\}$. Specifically, the first-stage decoder relies only on the extracted feature. Subsequent stages additionally use global semantic information that is built on top of the initial estimate, in a stage-wise decoding paradigm. For completeness, we first describe basic attentional decoder in a generalized fashion (ignoring stage notation). Later on, we particularly illustrate the design for first stage ($s = 0$) vs. later stages ($s \geq 1$).

3.2.1 Attentional Decoder Background

Text recognition framework aims to model conditional distribution $P(Y|I)$, which can be factorized as $P(Y|I) = \prod_{t=0}^{T} P(y_t | V, y_{<t})$ where each character output $y_t$ is modelled via conditional distribution over extracted visual information $V$, and the history of previously predicted characters $y_{<t}$ till then. The basic attentional decoder [53] models this factored conditional distribution using an autoregressive Recurrent Neural Network (RNN) as follows:

$$P(Y|I) = \prod_{t=0}^{T} P(y_t | g_t, H_{t-1}, y_{t-1})$$

(1)

Every time-step prediction $y_t$ is conditioned on three factors: (i) $H_{t-1} \in \mathbb{R}^{d_{h_{n}}}$: the previous hidden state of RNN that captures the history knowledge of previously predicted characters $y_{<t} = \{y_0, \cdots, y_{t-1}\}$. (ii) The apparent influence of previously predicted character $y_{t-1}$ upon predicting $y_t$, following character modelling protocol. (iii) The glimpse vector $g_t$, that learns to encode the visual information by attending a smaller specific part of visual feature, which is maximally relevant to predict the character $y_t$. 

Figure 2: With the extracted context-rich holistic feature \( (h_L) \) and multi-scale feature maps \((B_L, B_{L-1}, B_{L-2})\), a multi-stage attentional decoder predicts the character sequences in consecutive stages. Once the previous stage’s decoder completely unrolls itself across time, the current one begins prediction using the global-semantic information from previous stage’s predicted character sequence, coupled with visual features refined via joint visual-semantic reasoning. 

Technically, \( g_L = \psi(B, Q_t) \), where \( B \) is a spatial feature-map, encoding visual information from previous convolutional network, and \( Q_t \) acts as a query to locate the attentive regions for predicting \( y_t \). Mathematically put,

\[
\begin{aligned}
J &= \tanh(W_B \odot B + W_H \odot Q_t) \\
\alpha_{i,j} &= \text{softmax}(W_{\text{attn}} \odot J_{i,j}) \\
y_t &= \sum_{i,j} \alpha_{i,j} \cdot B_{i,j} \quad i = [1...H], \quad j = [1...W]
\end{aligned}
\]  

(2)

Here, “\( \odot \)” and “\( \odot \)” denote convolution and matrix multiplication respectively. \( W_B, W_H, W_{\text{attn}} \) are the learnable parameters. Usually, \( Q_t = H_{t-1} \) containing history of prediction information is used as a query to locate \( y_t \). Moreover, query vector enriched in global semantic information (e.g. as in \( s \geq 1 \)) could also be used instead, for better performance. While calculating the attention weight \( \alpha_{i,j} \) at every spatial position \((i, j)\), we employ a convolution operation with \( 3 \times 3 \) kernel \( W_{\text{attn}} \), to consider the neighborhood information in 2D attention mechanism.

The current hidden state \( H_t \) is updated by:

\[
H_t = f_{\text{rnn}}(H_{t-1}; [E(y_{t-1}), g_t])
\]

where \( E(\cdot) \) is character embedding layer with embedding dimension \( \mathbb{R}^{128} \), and \([\cdot]\) signifies a concatenation operation. Finally, we apply a final linear classification layer having learnable weights \((W_v, b_v)\) and giving logits \( l_t = F_{\text{cls}}(H_t) \); \( l_t \in \mathbb{R}^{|V|} \) where \(|V|\) denotes the character vocabulary size. The current step character is obtained as:

\[
P(y_t) = \text{softmax}(l_t).
\]

3.2.2 Decoder Stage \( s = 0 \)

Henceforth, we affix notation for specific decoder stage keeping earlier mathematical notation intact. For the first stage decoder RNN \( f_{\text{rnn}}^0 \), the initial hidden state is initialized from holistic visual feature:

\[
H_0^0 = \tanh(W_v \odot h_L + b_v), \quad \text{with} \quad W_v, b_v \text{ being trainable parameters. This enriches } f_{\text{rnn}}^0 \text{ with holistic visual information, while } g_0^0 = \psi(B, H_{t-1}^0) \text{ augments with localized character specific information. At every } t\text{-th step, we obtain the distribution over the output character space as } P(y_t^0) = \text{softmax}(F_{\text{cls}}^0(H_t^0)) \text{ and } H_t^0 = f_{\text{rnn}}^0(H_{t-1}^0; [E(y_{t-1}^0), g_t^0]). \text{ The decoding process stops once the ‘end-token’ is predicted. Sequences having variable length in batches are handled by zero-padding.}
\]

3.2.3 Decoder Stage \( s \geq 1 \)

Incorrect instances might exist in the prediction of preceding stage, which is why the later stages should work towards refining erroneous predictions while keeping the correct ones intact. While this seems similar to Language Model (LM) based post-processing [50] or Error Correction Network [50], our proposed stage-wise decoders are all coupled in an end-to-end trainable deep architecture. Here, gradients can backpropagate across stages during training, thus leading to learning better data driven refining strategy re-utilising the visual feature. The later stage decoders \( s \geq 1 \) are modelled as follows:

\[
P(Y^s | I) = \prod_{t=0} P(y_t^s | y_{t-1}^s, H_{t-1}^s, \mu_t^s)
\]

(3)

Fundamentally, there are three differences compared to basic attentional decoder (Eqn. 1):

(i) \( y_t^s \) is conditioned on joint visual-semantic information \( \mu_t^s = [\phi_t^s, \chi_t^s] \), where visual-part comes from \( \phi_t = \phi_t(H^{s-1}) \) and global semantic part comes from \( \chi_t^s = \omega_t(E(Y^s-1)) \). Here, \( \phi(\cdot) \) and \( \omega(\cdot) \) are reasoning modules working on previous stage’s character aligned visual feature \( H^{s-1} \in \mathbb{R}^{T \times d_{\text{fin}}} \) and semantic characters.
the output logits are differentiable. As our motivation lies in coupling multi-stage vocabulary extraction. To take advantage from multiple stages, we add a dense connection [23] between computed current $g_t^s$ and preceding $g_{<s}^s = \{g_{t-1}^s, \ldots, g_0^s\}$ as: $g_t^s = \text{Gumbel-Softmax}(l_t^s)$. Next, we calculate joint visual-semantic feature $\mu_t^s$ and successively $g_t^s$ is computed. Glimpse vector $g_t^s$ for $s \geq 1$ is enhanced by dense connection to give $g_t^s$. Now we update the hidden state of current stage decoder RNN by: $H_t^s = \text{RNN}(H_{t-1}^s; [E(y_{t-1}^s), g_t^s, \mu_t^s])$. Excluding the final stage, we directly apply linear classifier to get: $P(y_t^s) = \text{softmax}(F_{\text{cls}}^s(H_t^s))$.

For the final stage, $s = S$, we add a residual connection [22] between initial $H_0^S$ and final $H^S$ using LayerNorm [54] as follows: $H_t^S = \text{LayerNorm}(H_t^S + H_0^S)$. The motivation aligns with original residual convolutional architecture [22], but here we integrate it to train deeper model with multiple attention decoder stages for text recognition. The final prediction is obtained as $P(y_t^S) = \text{softmax}(F_{\text{cls}}^S(H_t^S))$. See Figure 2 for more clarity.

3.3. Learning Objective

We accumulate cross-entropy loss from all stages of attentional decoder to train our text-recognition model.

$$L_C = -\sum_{s=0}^{S-1} \sum_{t=1}^{T} \hat{y}_t \cdot \log P(y_t^s|H_t^s)$$

where $\hat{Y} = \{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_T\}$ is the ground-truth label. Furthermore, we use additional auxiliary linear classifier over character aligned individual visual and semantic features $\vartheta_t^s$ and $\chi_t^s$ respectively, that are processed through reasoning modules. The next two losses could be thought of as an auxiliary loss driving towards better convergence that enrich individual character aligned features with better visual and semantic information. This is given by: $L_V = -\sum_{s=1}^{S} \sum_{t=1}^{T} \hat{y}_t \cdot \log P(y_t^s|\vartheta_t^s)$ and $L_S = -\sum_{s=1}^{S} \sum_{t=1}^{T} \hat{y}_t \cdot \log P(y_t^s|\chi_t^s)$. The network is thus trained using: $L_{\text{Total}} = \lambda_1 \cdot L_C + \lambda_2 \cdot L_V + \lambda_3 \cdot L_S$, where $\lambda_1, \lambda_2, \lambda_3$ are weights decided empirically.
4. Experiments

Datasets: Following the similar approach described in [63, 59, 1, 15, 53, 43], we train our model on synthetic datasets (without any further fine-tuning) such as SynthText [20] and Synth90k [24], which holds 6 and 8 million images respectively. The evaluation is performed without fine-tuning on datasets containing real images like: Street View Text (SVT), ICDAR 2013 (IC13), ICDAR 2015 (IC15), CUTE80, SVT-Perspective (SVT-P), IIIT5K-Words. Street View Text dataset [56] consists of 647 images, most of which are blurred, noisy or have low resolution. While ICDAR 2013 [27] has 1015 words, ICDAR 2015 contains a total of 2077 images of which 200 images are irregular. CUTE80 [49] offers 288 cropped high-quality curved text images. SVT-Perspective [47] presents 645 samples from side-view angle snapshots containing perspective distortion. IIIT5K-Words [45] distinguishes itself by presenting randomly picked 3000 cropped word images.

Implementation Details: We use ResNet architecture from [53] with FPN heads having 256 channels in each multi-scale feature-maps. The kernel size of intermediate pooling layers is so adjusted that $B_{L}, B_{L-1}, B_{L-2}$ have spatial size of $4 \times 25$, $8 \times 25$, and $16 \times 50$ respectively. The hidden state size of two-layer encoder BLSTM and each decoder LSTM is kept at 256. Semantic ($\omega$) and visual ($\phi$) reasoning blocks consist of 2 stacked transformer units [54] with 4 heads and hidden state size 256. The hidden units in attention block is of size 128. A total of 37 classes are taken including alphabets, numbers and end-tokens; with the maximum sequence length (N) set to 25. We use ADADELTA optimizer [1] with learning rate 1.0 and batch size 32. We resize the image to 32x100 and train our model in a 11 GB NVIDIA RTX-2080-Ti GPU using PyTorch. We first warm-up using single stage attentional decoder for 50K iterations, and then train our proposed three-stage ($S = \{0, 1, 2\}$) attentional decoder (ablation on optimal stages in Sec. 4.2) framework end-to-end, for 600K iterations with $\lambda_1, \lambda_2, \lambda_3$ set to 1, 0.1, 0.1 respectively. Please note that the first stage is fed with one-time step shifted ground truth label to accommodate teacher forcing in sequence modeling, however, later stages are fed with model’s prediction from previous stage in order to learn the data driven refining strategy.

4.1. Result Analysis and Discussion

Table 1 shows our proposed method to surpass SOTA methods by a reasonable margin. Every method’s salient contributions are briefly mentioned there as well. In this section, we first describe the limitations of the existing or alternative (naive) designs and then illustrate (using IC15) how and why all our design components/choices contribute towards superiority over others.

Figure 3: Examples showing how joint visual-semantic information could help to recognise through refining over stages ($s = 0 \to s = 1 \to s = 2$), shown left to right.

[i] Limitation of previous attentional decoders: Existing methods relying on unidirectional auto-regressive attentional decoders exhibit a bottleneck, and its drawback becomes evident from the following scenario: An easily recognizable character present towards the end of a word would fail to provide any contextual semantic information towards recognizing some noisy character present earlier. We on the contrary let the first stage completely unroll itself. Thereafter the prediction of previous stage (even if certain time-step’s character is incorrect) could be rectified in the subsequent stages using joint visual-semantic information. Although SCATTER [37] stacks multiple BLSTM layers on the top of baseline design from ASTER [53], both methods lack semantic reasoning as they barely enrich visual feature encoding. Examples from our stage-wise decoder are shown in Figure 3.

[ii] Significance of Differentiable Semantic Space: Improving semantic reasoning for better text recognition was only considered by [61] and [46] among all SOTA methods. Although Qiao et al. [46] proposed to use word embedding, such technique relies on semantic meaning of a word instead of the required character sequence. For example, the word “table” and “chair”, although semantically related have character combinations that are way-apart. Therefore, we emphasise on modelling character sequences instead, to help recognize a noisy character based on two-way information passing. Even though Yu et al. [61] took this direction to some extent, their non-differentiable semantic-reasoning block imposes a significant limitation. We alleviate that with the help of gumbel-softmax [26] to develop a differentiable semantic space and allow learning of multi-stage semantic reasoning. While the use of teacher forcing for later stages by feeding ground-truth label for training multi-stage decoder might seem an alternative, empirical evidence suggests otherwise. The third stage decoder obtains 74.4% accuracy as compared to 74.5% accuracy (on IC15) in first stage – no practical gains. Another straightforward way is to use straight-through estimator [5], which simply copies gradients from argmax output to the next in-
| Methods       | Year | IIIT-5K | SVT  | IC13 | IC15 | SVT-P | CUTE80 | Remarks                                      |
|---------------|------|---------|------|------|------|-------|--------|----------------------------------------------|
| Shi et al. [31]| 2015 | 81.2    | 82.7 | 89.6 | -    | 66.8  | 54.9   | • End-to-end trainable CNN + RNN + CTC.     |
| Lee et al. [32]| 2016 | 78.4    | 80.7 | 90.8 | -    | 42.7  | -      | • Recursive CNN + RNN + Atten. decoder.     |
| Shi et al. [32]| 2016 | 81.9    | 81.9 | 88.6 | -    | -     | -      | • Introduce rectification network for irregular images. |
| Cheng et al. [33]| 2017 | 87.4    | 85.9 | 93.3 | 70.6 | 71.5  | 63.9   | • Learning to focus on character centre, but needs char. location label. |
| Liu et al. [39]| 2018 | 83.6    | 84.4 | -    | 60.0 | 73.5  | -      | • Rectify the distortion at individual character level. |
| Bai et al. [3] | 2018 | 88.3    | 87.5 | 94.4 | 73.9 | -     | -      | • Edit distance based seq. dissimilarity modeled to handle noisy characters. |
| Liu et al. [40]| 2018 | 89.4    | 87.1 | 94.0 | -    | 73.9  | 62.5   | • Leverage rendering parameters of synth. word image generation for training. |
| Shi et al. [33]| 2018 | 93.4    | 93.6 | 91.8 | 76.1 | 78.5  | 79.5   | • Improved rectification network by Thin-Plate Spline. |
| Cheng et al. [33]| 2018 | 87.0    | 82.8 | -    | 68.2 | 73.0  | 76.8   | • Four directional convolutional feature extraction for irregular images. |
| Liao et al. [35]| 2019 | 91.9    | 86.4 | 91.5 | 79.9 | -     | -      | • Segment individual character + discrete char. recogn. and word formation. |
| Yang et al. [59]| 2019 | 94.4    | 88.9 | 93.9 | 78.7 | 80.8  | 87.5   | • Models geometrical attributes of text for better images rectification. |
| Li et al. [33] | 2019 | 95.0    | 91.2 | 94.0 | 78.8 | 86.4  | 89.6   | • Introduce 2D-attention to deal with irregular images. |
| Baek et al. [1] | 2019 | 87.9    | 87.5 | 92.3 | 71.8 | 79.2  | 74.0   | • Comparative study of different methods and insightful analysis. |
| Zhan et al. [63]| 2019 | 93.3    | 90.2 | 91.3 | 76.9 | 79.6  | 83.3   | • Iterative image rectification. |
| Our Baseline (Stage-0) | -   | 88.0    | 84.9 | 90.4 | 74.5 | 75.3  | 82.6   | • Joint visual-semantic reasoning through multi-stage decoding using multi-scale feature maps and differential semantic space. |
| Our Baseline (Stage-1) | -   | 92.6    | 89.5 | 93.9 | 80.3 | 81.5  | 87.2   | • Faster parallel decoding + semantic reasoning block (non-differentiable). |
| Proposed (Stage-2)    | -   | 95.2    | 92.2 | 95.5 | 84.0 | 85.7  | 89.7   | • Stacking more Bi-LSTM layers + gated fusion of visual-contextual feature. |
| Our Baseline (Stage-3) | -   | 95.2    | 92.1 | 95.5 | 83.6 | 85.5  | 89.6   | • Tries to predict the word-embedding vector to initialise the state of decoder. |

**Put.** However, this results in significant instability where later stage performance drops by 3.9% to 80.1% due to discrepancies between forward and backward passes resulting in much higher variance than gumbel-softmax [26].

**[iii] Why not directly use logits instead of gumbel-softmax for semantic reasoning:** Feeding logits (probability distribution over character vocabulary prior to argmax) from a previous stage to the next, is a reasonable argument that would make everything differentiable and eliminate the need for gumbel-softmax. However, it is important to remember that characters are discrete tokens [1]. Using logits requires one to replace character embedding layer $E(\cdot)$ by a simple FC layer. Unlike $E(\cdot)$ that picks up a particular row of a trainable matrix based on discrete one-hot vector, a FC layer will give varying representations for the same character sequence based on the confidence of predictions. We confirm this hypothesis of sub-optimality empirically, as results drop from 84.0% to 82.8%.

**[iv] Why use top-down attentional decoder:** While low resolution and semantically strong features are good for classification, tasks requiring focus in local regions, such as object detection and semantic segmentation, benefit even further when combined with high-resolution semantically weak features found in shallower regions of a feature extractor [11]. Although our first stage is similar to a basic attentional decoder focusing on feature map of the last layer to benefit from rich semantic information, that is more invariant to distortion, later stages (refining stages) combine higher resolution feature-map from preceding layers. This not only handles varying character size, but also verifies prior prediction by exploiting joint information between high resolution projection and previous predictions to guide the refining process. This hypothesis is verified by contradiction, high-resolution semantically weak feature $B_{L-2}$ in $s = 0$ and lower resolution semantically strong features in later stages $s > 1$. We observe performance collapses to 72.1% in IC15 dataset due to inability of high resolution semantically weak features to output the initial estimates.

**[v] Significance of self-attention based Joint Visual-Semantic Reasoning:** To emulate human-like inference, self-attention based reasoning functions allow two way information passing across visual and semantic spaces to obtain a joint visual-semantic context. Its significance could be empirically understood by removing the visual reasoning block and modifying the architecture accordingly, which drops result by 2.9%. A similar drop of 4.8% was observed when the semantic reasoning block was removed. On removing both we observe 77.1% accuracy – a significant drop of 6.9% from our method (Table 2).

**[vii] Do multi-scale (resolution) feature maps help?** We empirically validate this by excluding multi-scale feature maps and use $B_L$, instead of $B_{L-s}$, to calculate $g_s^t$ at every stage $s$. Such modification drops performance by 2.7% (against ours), to 81.3%, which highlights the contribution of multi-scale feature maps in our method.

**[vii] Comparison with alternative multi-scale attentional decoder designs:** In text recognition, the only other work realising importance of multi-scale information is by Wan et al. [55], where pyramid pooling was used. Here visual feature maps from different spatial resolutions were concatenated, which eventually harmed downstream tasks owing to the large semantic gaps between such feature maps. Consequently, we introduce lateral connections following Feature Pyramid Networks [36], semantically strengthening high-resolution levels for superior performance. Simply employing pyramid pooling for all stages $s = \{0,1,2\}$ however, drops performance by 2.1% (against ours) to 81.9%.

**[viii] Significance of Dense and Residual Connections:**
Table 2: (Left) Effect of multiple constraints on IC15. (Right) Varying training data size. \( L_t^C \): Last stage only, \( L_t^C \): Multi-Stage, GAP: WRA margin against final performance.

| \( L_t^C \), \( L_t^C \), \( L_t^C \) | Method | Syn | Syn | Syn | 10K | 50K | 100K | 1M |
|-----------------|---------|-----|-----|-----|-----|-----|-----|-----|
| ✔ - - - | 73.1 | 6.9 | 52.1 | 67.4 |
| ✔ - ✔ | 79.2 | 4.8 |
| - ✔ - | 81.1 | 2.9 |

Beside improving visual information flow in the forward pass, the residual connection between initial \( H_t^0 \) and final \( H_t^S \) ensures efficient gradient flow in visual feature networks, accelerating convergence of the whole network. Furthermore, the dense connection is used to adaptively learn a more discriminative glimpse vector by combining its features from preceding stages with the current one, thus stabilising the training of multi-stage multi-scale attentional decoder. Removing dense connection \( g_t \) calculation decreases the performance by 1.6\%, and removing residual connection decreases it by 1.3\%. On removing both we get an even larger drop of 1.9\%. Faster training is observed while using both dense and residual connections.

**[ix] Significance of Multiple Constraints:** We design experimental setups (Table 2) that reveal the following observations: (a) imposing loss \( L_C \) only in the last stage harms the model, resulting in 73.1\% accuracy. We attribute this to the poor gradient flow across stages. (b) Adding multi-stage \( L_C \) loss results in 77.1\% accuracy, performing closer to the proposed method. (c) Adding visual-semantic constraints \( L_V \) and \( L_S \) finally gives the best performance of 84.0\%. This shows multi-stage constraint is vital for training and convergence. The intuition behind multiple constraints sources from multi-task learning, which ensures better convergence, thus enriching individual character aligned feature, with better visual-semantic information.

**[ix] Varying training data size:** Following [44], we also vary the training size and evaluate our proposed framework compared to single stage baseline and Yu et al. [61] in Table 2. Significant overhead at low data regime brings the superiority to our proposed method over others.

### 4.2. Further Analysis and Insights

**[i] Design of Visual-Semantic Reasoning Module:** One can capture two-way visual semantic information using (a) Bi-LSTM (b) Transformer [54] with multi-headed self-attention mechanism. Table 3 shows Transformer to outperform LSTM by 1.3\%. Furthermore, pre-training global semantic reasoning module \( \omega(\cdot) \) using BERT [16] like training topology, scores 0.9\% higher accuracy than without it.

**[ii] Weight sharing across stages:** The stage-wise attentional decoder has five trainable modules, \( F_{\text{cls}}(\cdot), E(\cdot), \Phi(\cdot), \omega(\cdot) \) and \( \psi(\cdot) \), whose weights can either be shared across stages or have a separate module for each stage. Using separate weights achieves 82.5\% accuracy on IC15, whereas sharing across stages results in 82.3\%. Interestingly using a separate \( F_{\text{cls}}, \omega \), and shared \( E(\cdot), \Phi, \omega \) gives 84.0\%, a 1.7\% rise, in contrast to sharing all weights – probably because sharing parameters which are not stage dependent reduce model complexity and has better optimization.

**[iii] Computational Analysis:** Each stage needs to unroll itself completely, before the next starts processing. Hence, the performance gain comes at a cost of extra computational expenses (analysis in Table 4), which is reasonable given the superior performance over strong baselines. Even so, we experimented with ResNet-101 as a backbone feature extractor, having similar number of parameters and flops to ours. This naive stacking of multiple-layers lags by 8.9\%, which accredits our gain to our novel design choice.

**[iv] Comparison with SOTA Language Model:** We compare our framework with state-of-the-art Language Modeling (LM) based post-processing techniques based on librispeech text-corpus. Based on [19] we adopt two techniques: (a) Shallow Fusion that results in 74.3\% and (b) Deep Fusion giving 75.9\% accuracy on IC15 (Table 3).

**[v] Optimum Stages:** The optimal value for the number of stages \( s \) is found empirically on IC15. For \( s = 1 \) we have 80.3\% accuracy that improves at \( s = 2 \) to give 84.0\%, but saturates at \( s = 3 \) giving 83.6\%. Hence we consider \( s = 2 \) to be optimal. This performance saturation could be attributed to vanishing gradient problem which is addressed via residual/dense connection, but still persists to some extent. Also, for \( s > 2 \), the joint visual-semantic information might reach its optimum, where the result saturates. Please refer to supplementary material as well.

### 5. Conclusion

We propose a novel joint visual-semantic reasoning based multi-stage multi-scale attentional decoding paradigm. The first stage predicts from visual features, followed by refinement using joint visual-semantic information. We further exploit Gumbel-softmax operation to make visual-to-semantic embedding layer differentiable. This enables backpropagation across stages to learn the refining strategy using joint visual-semantic information. Experimental results indicate superior performance of our model.
References

[1] Jeonghun Baek, Geewook Kim, Junyeop Lee, Sungrae Park, Dongyoon Han, Sangdoo Yun, Seong Joon Oh, and Hwalsuk Lee. What is wrong with scene text recognition model comparisons? dataset and model analysis. In ICCV, 2019. 1, 6, 7

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In ICLR, 2015. 1, 2

[3] Fan Bai, Zhanzhan Cheng, Yi Niu, Shiliang Pu, and Shuigeng Zhou. Edit probability for scene text recognition. In CVPR, 2018. 1, 7

[4] Sean Bell, C Lawrence Zitnick, Kavita Bala, and Ross Girshick. Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. In CVPR, 2016. 3

[5] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. arXiv preprint arXiv:1308.3432, 2013. 6

[6] Ayan Kumar Bhunia, Pinaki Nath Chowdhury, Aneeshaan Sain, and Yi-Zhe Song. Towards the unseen: Iterative text recognition by distilling from errors. In ICCV, 2021. 2

[7] Ayan Kumar Bhunia, Abhirup Das, Ankan Kumar Bhunia, Perla Sai Raj Kishore, and Partha Pratim Roy. Handwriting recognition in low-resource scripts using adversarial learning. In CVPR, 2019. 1

[8] Ayan Kumar Bhunia, Shuvozit Ghose, Amaandeep Kumar, Pinaki Nath Chowdhury, Aneeshaan Sain, and Yi-Zhe Song. Metahtr: Towards writer-adaptive handwritten text recognition. In CVPR, 2021. 1

[9] Ayan Kumar Bhunia, Aneeshaan Sain, Pinaki Nath Chowdhury, and Yi-Zhe Song. Text is text, no matter what: Unifying text recognition using knowledge distillation. In ICCV, 2021. 1

[10] Ali Furkan Biten, Ruben Tito, Andres Mafla, Luis Gomez, Marçal Rusiñol, Ernest Valveny, CV Jawahar, and Dimosthenis Karatzas. Scene text visual question answering. In CVPR, 2019. 1

[11] Zhaowei Cai, Quanfu Fan, Rogerio S Feris, and Nuno Vasconcelos. A unified multi-scale deep convolutional neural network for fast object detection. In ECCV, 2016. 3, 7

[12] João Carreira, Pulkit Agrawal, Katerina Fragkiadaki, and Jitendra Malik. Human pose estimation with iterative error feedback. In CVPR, 2016. 2

[13] Xinlei Chen, Li-Jia Li, Li Fei-Fei, and Abhinav Gupta. Iterative visual reasoning beyond convolutions. In CVPR, 2018. 1, 2, 3

[14] Zhanzhan Cheng, Fan Bai, Yunlu Xu, Gang Zheng, Shiliang Pu, and Shuigeng Zhou. Focusing attention: Towards accurate text recognition in natural images. In ICCV, 2017. 1, 2, 7

[15] Zhanzhan Cheng, Yangliu Xu, Fan Bai, Yi Niu, Shiliang Pu, and Shuigeng Zhou. Aon: Towards arbitrarily-oriented text recognition. In CVPR, 2018. 2, 3, 6, 7

[16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In NAACL, 2019. 5, 8

[17] Yazan Abu Farha and Jurgen Gall. Ms-ten: Multi-stage temporal convolutional network for action segmentation. In CVPR, 2019. 3

[18] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In ICML, 2006. 2

[19] Caglar Gulcehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loïc Barrault, Hieu-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. On using monolingual corpora in neural machine translation. In ACL, 2015. 8

[20] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localisation in natural images. In CVPR, 2016. 6

[21] Bharath Hariharan, Pablo Arbelaez, Ross Girshick, and Jitendra Malik. Hypercolumns for object segmentation and fine-grained localization. In CVPR, 2015. 3

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In CVPR, 2016. 5

[23] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In CVPR, 2017. 5

[24] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Synthetic data and artificial neural networks for natural scene text recognition. arXiv preprint arXiv:1406.2227, 2014. 6

[25] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and koray kavukcuoglu. Spatial transformer networks. In NeurIPS, 2015. 1

[26] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In ICLR, 2017. 1, 2, 5, 6, 7

[27] Dimosthenis Karatzas, Faisal Shafait, Seiichi Uchida, Masakazu Iwamura, Luis Gomez i Bigorda, Sergi Robles Mestre, Joan Mas, David Fernandez Mota, Jon Almazan Almazan, and Luis Pere De Las Heras. Icdar 2013 robust reading competition. In ICDAR, 2013. 6

[28] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. ICLR, 2017. 3

[29] Tiao Kong, Anbang Yao, Yurong Chen, and Fuchun Sun. Hypernet: Towards accurate region proposal generation and joint object detection. In CVPR, 2016. 3

[30] Ranjay Krishna, Yuek Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. IJCV, 2017. 3

[31] Alex M Lamb, Anirudh Goyal Alias Parth Goyal, Ying Zhang, Saizheng Zhang, Aaron C Courville, and Yoshua Bengio. Professor forcing: A new algorithm for training recurrent networks. In NeurIPS, 2016. 2
