Vietnamese Handwritten Text Recognition

Nguyễn Mạnh Dương
Trường Công nghệ Thông tin và Truyền thông
Đại học Bách Khoa Hà Nội

Nguyễn Quang Tri
Trường Công nghệ Thông tin và Truyền thông
Đại học Bách Khoa Hà Nội

Nguyễn Thanh Bình
Trường Công nghệ Thông tin và Truyền thông
Đại học Bách Khoa Hà Nội

Trần Đức Tuấn Kiên
Trường Công nghệ Thông tin và Truyền thông
Đại học Bách Khoa Hà Nội

Phạm Tuấn Kiệt
Trường Công nghệ Thông tin và Truyền thông
Đại học Bách Khoa Hà Nội

Abstract

This report presents a comprehensive exploration and implementation of a Vietnamese Handwritten Text Recognition (VHTR) system using advanced deep learning techniques. Handwritten text recognition remains a challenging task due to the inherent variability in individual writing styles and script complexity. In this project, we address these challenges by leveraging state-of-the-art deep learning architectures, including convolutional neural networks (CNNs) and recurrent neural networks (RNNs), to develop a robust VHTR model.

The project involves the preprocessing of handwritten text images, including data augmentation and normalization, to enhance the model's ability to generalize across diverse writing styles. The proposed VHTR model is trained on a large dataset of handwritten Vietnamese text samples, with an emphasis on optimizing hyperparameters to achieve superior performance.

1 Introduction

In an era where digital transformation is reshaping every facet of our lives, the need for efficient and accurate text recognition technologies has become increasingly vital. Handwritten Text Recognition (HTR) stands at the forefront of this technological revolution, bridging the gap between traditional written communication and the digital realm. This report delves into the intricacies of a groundbreaking project focused on Vietnamese Handwritten Text Recognition, aiming to revolutionize the way we interact with and digitize handwritten documents in the Vietnamese language.

Vietnamese, a language with a rich cultural heritage, presents unique challenges in the field of Handwritten Text Recognition due to its intricate characters and diverse writing styles. Recognizing the significance of preserving and digitizing valuable handwritten content in Vietnamese, our project endeavors to develop state-of-the-art recognition models that transcend the limitations of existing technologies.

This report not only outlines the motivations and objectives behind the Vietnamese Handwritten Text Recognition project but also provides a comprehensive overview of the methodologies employed, the datasets curated, and the technical intricacies encountered during the development process. Furthermore, it explores the potential applications and implications of the project, underscoring its broader impact on digitizing historical archives, enhancing accessibility, and facilitating seamless integration into various sectors.

Preprint. Under review.

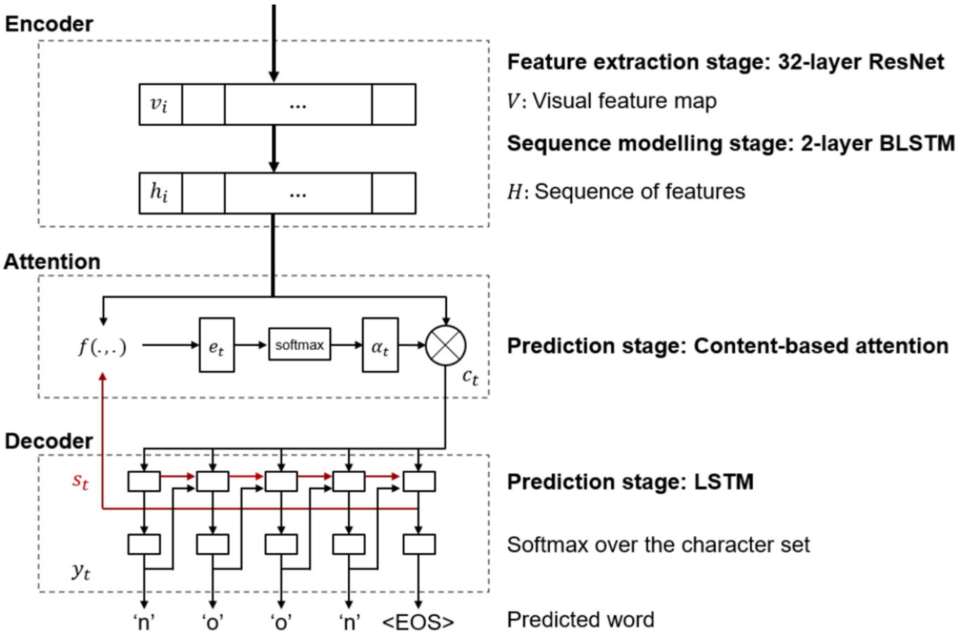

Encoder
v_i
...
h_i
...
Feature extraction stage: 32-layer ResNet
V: Visual feature map
Sequence modelling stage: 2-layer BLSTM
H: Sequence of features
Attention
$f(.,.)$
e_t
softmax
α_t
c_t
Prediction stage: Content-based attention
Decoder
s_t
y_t
Prediction stage: LSTM
Softmax over the character set
'n' 'o' 'o' 'n' <EOS>
Predicted word

Figure 1: Our proposed solution - contains 3 main stages. The image is passed through feature extractor, and then modeled as sequence of feature before aggregated via attention based. At the prediction stage, to follow sequential property of texts, LSTM is applied before softmax to get the probability over character set

As we navigate through the chapters of this report, we will unravel the complexities of Vietnamese Handwritten Text Recognition, shedding light on the innovative solutions and advancements that have been achieved. The ultimate goal is to contribute to the ever-evolving landscape of optical character recognition, fostering a deeper connection between the analog past and the digital future in the context of the Vietnamese language.

2 Methodology

2.1 TrBA Model

The TRBA model integrates the transformer model's ability to capture long-range dependencies with bidirectional feature aggregation for scene text recognition. This model improves upon previous scene text recognition approaches by offering enhanced accuracy and efficiency.

2.1.1 Architecture

The TRBA model consists of three main components: Transformation, Feature Extraction, and Sequence Modeling followed by Prediction. Each component is designed to process the input text image sequentially to produce accurate text recognition. This layer will be removed in our pipeline for efficiency reason

Transformation (Trans.) The Transformation stage normalizes the input text image to make it more conducive for the feature extraction stage. It typically uses a Spatial Transformer Network (STN) to handle various text orientations and scales in the image.

- Spatial Transformer Network (STN) for handling text image distortions.

Feature Extraction (Feat.) In the Feature Extraction stage, a Convolutional Neural Network (CNN) is used to extract meaningful features from the normalized text images.

- CNN variants like VGG, RCNN, or ResNet to create rich feature representations.

Sequence Modeling (Seq.) This stage captures the contextual information within the sequence of characters using a bidirectional aggregation mechanism.

- Bidirectional LSTM or other RNN variants to model the sequence of features.

Prediction (Pred.) The final stage involves predicting the sequence of characters from the features identified by the model.

- Typically involves an attention-based decoder for sequence prediction.

2.2 Spatial Transformer Networks

Spatial Transformer Networks (STNs) introduce a novel and learnable module to the conventional neural network architecture, enabling explicit spatial manipulation of data within the network. This innovative component is a differentiable module that can be seamlessly integrated into existing convolutional architectures, thereby granting neural networks the capability to actively transform feature maps in a spatial manner.

The main intuition of STN is to automatically align images, which is useful for scene text recognition task. However, since this setting is not similar to our dataset with noise and plain background, we choose not to use this special layer as preprocess stage. This leads to the significant reduction in parameters number without much loss of performance

2.3 ResNet extractor

ResNet (Residual Network) introduces a novel architecture with "skip connections" or "shortcut connections" to enable training of very deep networks. The fundamental building block of a ResNet is the residual block.

2.3.1 Architecture

Residual Block Structure A typical residual block has the following form:

$$\text{Output} = \mathcal{F}(\text{Input}) + \text{Input}$$

Where:

- $\mathcal{F}(\text{Input})$: The residual function to be learned.
- Input: The input feature map.
- Output: The output feature map of the residual block.

The function $\mathcal{F}$ represents the residual mapping to be learned by the stacked layers (typically two or three convolutional layers) in the block.

Skip Connections The key component of ResNet is the skip connection that skips one or more layers:

$$\text{Output} = \mathcal{F}(\text{Input}) + \text{Input}$$

Skip connections do the following:

- Help in addressing the vanishing gradient problem by allowing this alternate shortcut path for gradient to flow through.
- Enable the network to learn identity functions effectively, ensuring that the added layers can only improve performance.

Deep ResNet Architecture Deep ResNet architectures stack these residual blocks, forming very deep networks. A typical deep ResNet might have 50, 101, or even more layers.

ResNet significantly improves the performance of deep neural networks through the use of residual blocks and skip connections, enabling the training of networks that are much deeper than was previously feasible.

2.4 Long Short-Term Memory (LSTM) Networks

Long Short-Term Memory networks – usually just called "LSTMs" – are a special kind of RNN, capable of learning long-term dependencies. They were introduced by Hochreiter Schmidhuber (1997), and were refined and popularized by many in the following years. LSTMs are explicitly designed to avoid the long-term dependency problem.

2.4.1 Architecture

An LSTM cell has three gates and a cell state vector to control the flow of information. The gates are:

- Forget Gate: It decides what information should be thrown away or kept.
- Input Gate: It updates the cell state with new information.
- Output Gate: It decides what the next hidden state should be.

The key equations governing the gates and state updates are:

$$
\begin{aligned}
f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\
i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\
o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\
\tilde{C}_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \\
C_t &= f_t * C_{t-1} + i_t * \tilde{C}_t \\
h_t &= o_t * \tanh(C_t)
\end{aligned}
$$

Where:

- f_t, i_t, and o_t are the forget, input, and output gates' activations at time step t, respectively.
- C_t is the cell state at time step t.
- h_t is the hidden state at time step t.
- x_t is the input at time step t.
- W and b terms represent weights and biases for different gates.
- σ denotes the sigmoid function, and $*$ denotes the Hadamard product (element-wise multiplication).

LSTMs are powerful for modeling sequential data because they maintain a cell state and utilize gates to regulate the flow of information, effectively allowing them to capture long-term dependencies and disregard non-relevant data.

2.4.2 Bidirectional LSTM

One of LSTM limitations is that it can only model the sequence of feature in one direction. However, in the image-to-text tasks, convolutional image features are usually transformed into sequential feature in order to be compatible with text feature space. Intuitively, images need to be reviewed from top to bottom, left to right (one direction) and opposite. In order to apply this human-like approach in our pipeline, Bidirectional LSTM is used for transform image features into corresponding vector in textual space

Bidirectional LSTM is actually composed of 2 LSTM layers with opposite directions. The intuition here is in some cases that the forward direction is insufficient for efficient inference, we need to fully understand the contextual information via both direction forward.

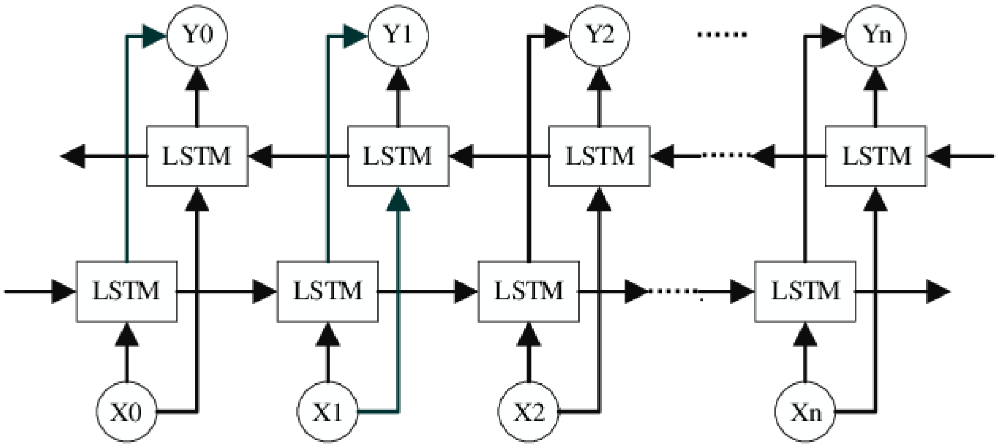


Figure 2: Bidirectional LSTM architecture

2.5 Attention Mechanism

The attention mechanism is a technique that allows neural networks to focus on different parts of the input sequence for each step of the output sequence, improving the quality of the results in tasks like translation, image captioning, and more. It was introduced to improve the performance of sequence-to-sequence models in handling long-range dependencies.

2.5.1 Idea

In its simplest form, the attention mechanism can be described as a mapping of a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key.

2.5.2 Basic Formulation

Given:

- A query q,
- A set of keys $K = \{k_1, k_2, \ldots, k_n\}$,
- A set of values $V = \{v_1, v_2, \ldots, v_n\}$,

The attention weights a and output o are computed as:

$$\begin{aligned} e_i &= \text{compatibility}(q, k_i) \\ a &= \text{softmax}(e) \\ o &= \sum_i a_i v_i \end{aligned}$$

Where:

- e_i is a score that represents how much focus to put on each part of the input sequence.
- The softmax function is used to normalize the scores to a probability distribution.
- o is the final output which is the weighted sum of the values.

2.5.3 Scaled Dot-Product Attention

One popular attention function is the scaled dot-product attention. The input consists of queries and keys of dimension d_k, and values of dimension d_v. The dot products of the query with all keys are computed, divided by $\sqrt{d_k}$, and a softmax function is applied to obtain the weights on the values.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Attention mechanisms have become an integral part of models that need to capture various dependencies in input data. They are especially prevalent in sequence modeling tasks and have led to significant improvements in machine translation, question answering, and other areas of natural language processing. This method is used in our pipeline as convention

3 Experiments and Results

3.1 Experimental Setup

3.1.1 Dataset

Our method's performance is assessed via the public dataset in Vietnamese Handwritten Text Recognition - OCR track in SOICT Hackathon 2023 competition. The dataset contains over 100k images in total, which is split as 80K for training, 20K for testing.

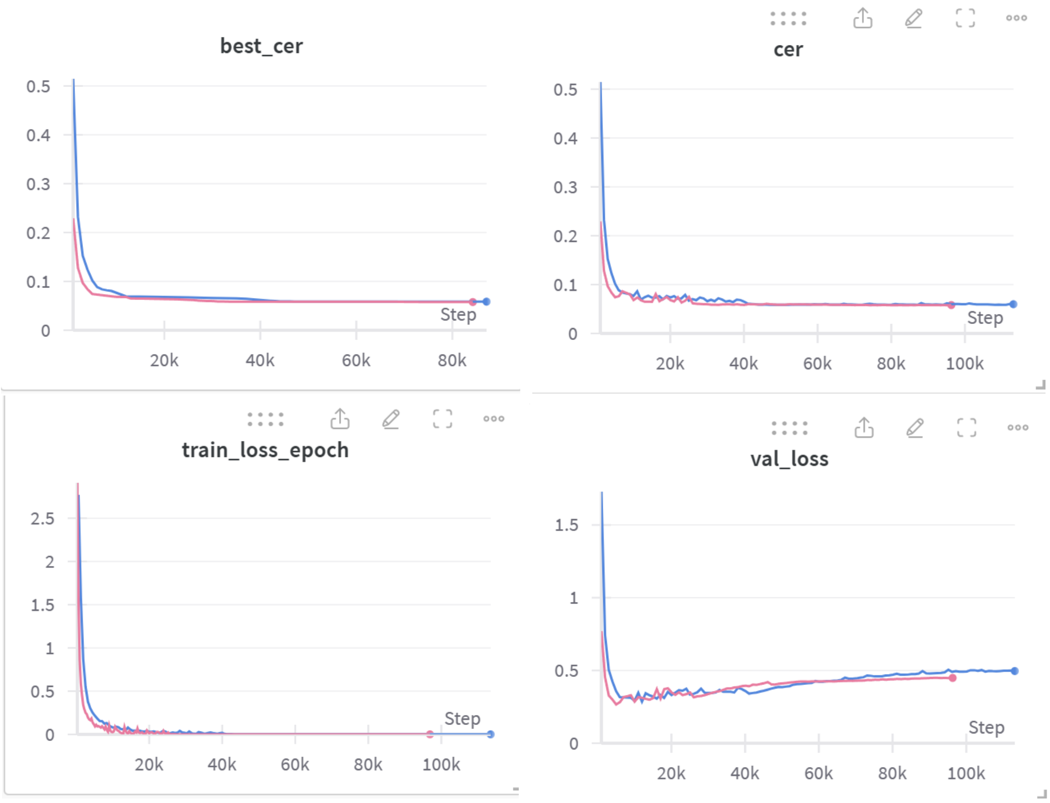
3.1.2 Network Setup

We choose ResNet-based CNN with 32 layers for feature extractor in order to increase compatibility with the next Bi-LSTM layer. Both original LSTM and Bi-LSTM are applied with the original setup, following PyTorch. At the last stage, scaled dot-product attention is used as convention.

3.2 Result Analysis

As it can be seen from cer and $best\_cer$, during the training steps, the CER of pink line (resize + padding) almost always lower than that of the blue line(resize + padding + normalize). Moreover, when the training steps become larger, CER of pink line is more stable, from which we can obtain that using only resize + padding brings more effectiveness compared to using resize + padding + normalize despite the same performance of the two choices of preprocessing. We raise an intuition that most of images in the dataset is plain and simple background, mostly black and white, which leads to insignificant effect of normalize stage.

Overfitting can be seen from the validation loss after about 10k - 20k steps. The CER metric is around 0.058, which is significant but not reach SOTA performance. The reason might be insufficient data since all the training images can not cover every character combination that might exist in real scenarios.


best_cer

cer

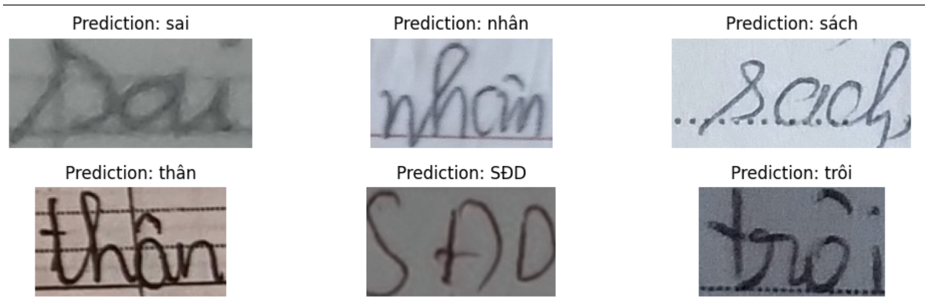
train_loss_epoch

val_loss

Figure 3: Experimental results

4 Conclusion

In conclusion, the Vietnamese Handwritten Text Recognition (HTR) project has proven to be a significant stride towards bridging the gap between traditional documentation and digital platforms in Vietnam. Through meticulous research, advanced machine learning algorithms, and the collaborative efforts of our team, we have successfully developed a robust system capable of accurately deciphering handwritten Vietnamese text. The project not only showcases the potential of cutting-edge technology in linguistic applications but also underscores its relevance in preserving and digitizing cultural heritage. As we move forward, the impact of this HTR system extends beyond mere technological advancement; it paves the way for enhanced accessibility to historical documents, improved information retrieval, and contributes to the broader global effort in digitizing diverse languages. This endeavor stands as a testament to the fusion of technology and cultural preservation, fostering a future where handwritten texts, regardless of language, can seamlessly integrate into the digital landscape.


Prediction: sai

Prediction: nhân

Prediction: sách

Prediction: thân

Prediction: SĐD

Prediction: trôi

Figure 4: Visualized inference result

5 Future Work

Multilingual Support: To enhance the versatility of the Vietnamese Handwritten Text Recognition (VHTR) system, future work could focus on extending language support to include other languages commonly used in Vietnam. This expansion would broaden the applicability of the system, making it more beneficial for a diverse user base.

Improved Accuracy and Robustness: Continuous efforts should be invested in refining the VHTR model's accuracy and robustness. Fine-tuning the model with larger and more diverse datasets, particularly focusing on challenging handwritten samples, can contribute to better performance in real-world scenarios.

Online Recognition Capability: The current VHTR system is primarily designed for offline handwritten text recognition. Incorporating online recognition capabilities, where the system processes text in real-time as it is being written, would be a valuable extension. This feature could find applications in digital pens, tablets, and interactive writing surfaces.

References

[1] R. Atienza. Data augmentation for scene text recognition, 2021.

[2] R. Atienza. Vision transformer for fast and efficient scene text recognition, 2021.

[3] J. Baek, G. Kim, J. Lee, S. Park, D. Han, S. Yun, S. J. Oh, and H. Lee. What is wrong with scene text recognition model comparisons? dataset and model analysis, 2019.

[4] D. Bautista and R. Atienza. Scene text recognition with permuted autoregressive sequence models, 2022.

[5] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

[6] Y. Du, Z. Chen, C. Jia, X. Yin, C. Li, Y. Du, and Y.-G. Jiang. Context perception parallel decoder for scene text recognition, 2023.

[7] Y. Du, Z. Chen, C. Jia, X. Yin, T. Zheng, C. Li, Y. Du, and Y.-G. Jiang. Svtr: Scene text recognition with a single visual model, 2022.

[8] H. N. Duc. Vietnamese word list: Ho ngoc duc's word list. `http://www.informatik.unileipzig.de/~duc/software/misc/wordlist.html`, 2004.

[9] S. Fang, H. Xie, Y. Wang, Z. Mao, and Y. Zhang. Read like humans: Autonomous, bidirectional and iterative language modeling for scene text recognition, 2021.

[10] S. Fogel, H. Averbuch-Elor, S. Cohen, S. Mazor, and R. Litman. Scrabblegan: Semi-supervised varying length handwritten text generation, 2020.

[11] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial transformer networks, 2016.

[12] J. Lee, S. Park, J. Baek, S. J. Oh, S. Kim, and H. Lee. On recognizing texts of arbitrary shapes with 2d self-attention, 2019.

[13] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge, 2015.

[14] D. Sang and N. Thuan. An efficientnet-like feature extractor and focal ctc loss for image-base sequence recognition. pages 326–331, 11 2020.

[15] L. N. Smith. A disciplined approach to neural network hyper-parameters: Part 1 – learning rate, batch size, momentum, and weight decay, 2018.

[16] X. Xie, L. Fu, Z. Zhang, Z. Wang, and X. Bai. Toward understanding wordart: Corner-guided transformer for scene text recognition, 2022.

[17] D. Yu, X. Li, C. Zhang, J. Han, J. Liu, and E. Ding. Towards accurate scene text recognition with semantic reasoning networks, 2020.