

2020

A Machine Learning-based Approach to Vietnamese Handwritten Medical Record Recognition

Tuan Minh Phung

Oxford University Clinical Research Unit, tuanpm@oucru.org

Minh Ngoc Dinh

Royal Melbourne Institute of Technology, minh.dinh4@rmit.edu.vn

Duy Pham Thien Dang

RMIT University, duy.dangphamthien@rmit.edu.vn

Hoang Minh Tu Van

Oxford University Clinical Research Unit, vanhmt@oucru.org

C. Louise Thwaites

Oxford University Clinical Research Unit, lthwaites@oucru.org

Follow this and additional works at: <https://aisel.aisnet.org/acis2020>

Recommended Citation

Phung, Tuan Minh; Dinh, Minh Ngoc; Dang, Duy Pham Thien; Van, Hoang Minh Tu; and Thwaites, C. Louise, "A Machine Learning-based Approach to Vietnamese Handwritten Medical Record Recognition" (2020). *ACIS 2020 Proceedings*. 22.

<https://aisel.aisnet.org/acis2020/22>

This material is brought to you by the Australasian (ACIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in ACIS 2020 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

A Machine Learning-based Approach to Vietnamese Handwritten Medical Record Recognition

Completed research paper

Tuan Minh Phung

Emerging Infection Group
Oxford University Clinical Research Unit
Ho Chi Minh City, Vietnam
Email: tuanpm@oucru.org

Minh Ngoc Dinh

School of Science and Technology
Royal Melbourne Institute of Technology
Ho Chi Minh City, Vietnam
Email: minh.dinh4@rmit.edu.vn

Duy-Dang Pham

School of Science and Technology
Royal Melbourne Institute of Technology
Ho Chi Minh City, Vietnam
Email: duy.dangphamthien@rmit.edu.vn

Hoang Minh Tu Van

Emerging Infection Group
Oxford University Clinical Research Unit
Ho Chi Minh City, Vietnam
Email: vanhmt@oucru.org

C. Louise Thwaites

Emerging Infection Group
Oxford University Clinical Research Unit
London, United Kingdom
Email: lthwaites@oucru.org

Abstract

Handwritten text recognition has been an active research topic within computer vision division. Existing deep-learning solutions are practical; however, recognizing Vietnamese handwriting has shown to be a challenge with the presence of extra six distinctive tonal symbols and extra vowels. Vietnam is a developing country with a population of approximately 100 million, but has only focused on digitalization transforms in recent years, and so Vietnam has a significant number of physical documents, that need to be digitized. This digitalization transform is urgent when considering the public health sector, in which medical records are mostly still in hand-written form and still are growing rapidly in number. Digitization would not only help current public health management but also allow preparation and management in future public health emergencies. Enabling the digitalization of old physical records will allow efficient and precise care, especially in emergency units. We proposed a solution to Vietnamese text recognition that is combined into an end-to-end document-digitalization system. We do so by performing segmentation to word-level and then leveraging an artificial neural network consisting of both convolutional neural network (CNN) and a long short-term memory recurrent neural network (LSTM) to propagate the sequence information. From the experiment with the records written by 12 doctors, we have obtained encouraging results of 6.47% and 19.14% of CER and WER respectively.

Keywords: Vietnamese handwriting recognition, Document layout analysis, Convolutional neural network, Recurrent neural network, Medical record recognition.

1 Introduction

Medical record is a document that stores a patient's manifestation of disease, development, examination, diagnosis, and treatment. Medical records play an essential role not only in medical treatment, but also in education and training, research, and development. Nowadays, thanks to the advancement of information and communication technologies (ICT), electronic health record systems are becoming a new standard for medical record management (Samadbeik et al. 2020). However, there are already millions of paper-based records in existence, which present many challenges. First, it is very challenging to maintain the paper-based records properly since such records usually have a single copy. Once they are degenerated or lost, they could no longer be recoverable. Second, indexing a specific paper-based record takes lots of effort (Cuk et al. 2017; Marutha and Ngoepe 2017). Those medical records are grouped up using ad-hoc naming conventions (e.g. hospital department name followed by the date they were created). Thus, tracking a patient's health history manually is almost impossible (Marutha and Ngoepe 2017). This is particularly concerning when treating complex patients with many illnesses, when knowledge of prior history, treatment and investigations is essential to deliver high quality patient care. Finally, it is a waste of important and valuable data because they are often not used for other purposes like training and analysis. In the age of Big Data, such a wealth of data is extremely useful for early diagnosis, prevention and treatment planning (Andreu-Perez, 2015).

Nevertheless, the resource required to digitalize the enormous amount of physical, paper-based records into machine processable forms is excessive, in terms of manpower, time and money (Cuk et al. 2017). Commercial solutions such as Google Cloud Vision and Microsoft Computer Vision do not work well in Vietnam because they do not have tailor-made models for Vietnamese. In fact, to the best of our knowledge, there is little-to-no research that applied state-of-the-art techniques in the Vietnamese handwritten text recognition domain. Such negligence emphasizes the urgency of this work because Vietnam is a middle-income country with a population of over 100 million people.

In this work, we proposed an end-to-end solution which takes an image of a hard-copy medical document and transcribes it into an electronic version. The approach performed segmentation to the word level and use a deep learning architecture consists of ResNet – BiLSTM - CTC to convert the image into text. The output sequence is finally constraint to a lexicon to further boost the accuracy.

The proposed work is important because a Vietnamese handwritten text recognition system can:

- Accelerate the digitalize process of medical health record system. With the help of the machine in processing all the records, health facilities could gradually shift to an electronic system without any sudden change in protocol. A system would also allow remote medical centres or field health workers with limited computer access to continue with paper systems which could then be digitalized easily.
- Improve the health care quality. Patient records could easily be shared between departments, thus reduces unnecessary tests and optimize treatment.
- Generate a digital medical-note dataset for a variety of potential medical machine-learning solutions. In fact, our healthcare collaborators (OUCRU and HTD) are planning to use the generated data to develop expert diagnostic systems, to improve the treatment process and to minimize errors in healthcare practices.

The paper is organized as follows. Section 2 provides an overview of related works in the field of text recognition. The pipeline's architecture is discussed in Section 3. Experiments and performance results are discussed in Section 4. Finally, we conclude the paper with plan for future works in Section 5.

2 Background

The automatic transcription of these handwritten documents is often referred as Handwritten Text Recognition (HTR), which is a subset of Optical Character Recognition (OCR) domain. Traditional handwritten text recognition methods were based on hand-crafted features engineering. Since 2015, Deep Learning, as a successful supervised learning framework for classification and recognition problems, has become the mainstream solution for large-scale OCR problems.

A common pipeline for character recognition (shown in Figure 1), given a document image as input, often consists of four tasks: *pre-processing*, *text localization* and *text segmentation*, and *text recognition*. In the following sections, we discussed these tasks to present the foundation for our proposed solution.

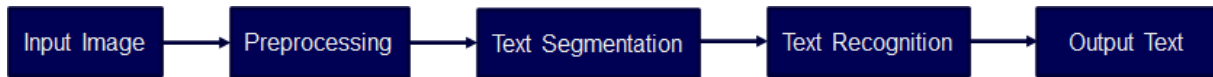


Figure 1. A common pipeline for handwritten text recognition

2.1 Pre-processing

Scanned documents, such as medical records that need to be digitized in this work, come with typical problems such as possible graphics, image, page orientation, noise, and skew problems. Binarization is often the first application one needs to do in the pre-processing stage. Binarization converts a given image into a binary/grayscale image. The method first generates a histogram of the pixel intensity values and then computes one or more thresholds for the background value of the image. Threshold value(s) is used to determine if a pixel belongs to the background using the formulae below.

$$I_b(x, y) = \begin{cases} \text{black} & \text{if } I_f(x, y) \leq Thr(x, y) \\ \text{white} & \text{if } I_f > Thr(x, y) \end{cases}$$

2.2 Text Localization and Segmentation

After pre-processing, the next step is to localize the text and segment it. Localization means identifying units of text (e.g. a single line, a single word, or a single character) and creating a collection of boundary boxes around them. These regions of interest (ROIs) are later used by the segmentation module.

There have been various attempts to develop efficient deep neural networks to detect units of text from an image. Zhou et al. (2017) presented a fast and effective framework for localizing multi-oriented text. The system utilized a fully convolutional network to generate word or line level prediction. Nevertheless, the model may miss the text within long regions or incorrectly transcribe some text due to its lack of training data. Baek et al. (2019) introduced a model named CRAFT (Character Region Awareness for Text) that describes inter-character relationships using a heat map and connects the characters using a connected component labelling (CCL) algorithm. The model is robust to text with different scale variance and achieves good generalization capability.

2.3 Text Recognition

Once units of text were identified and segmented, text recognition task can be conducted. Common text recognition methods can be divided into three categories: character classification based, word classification based, and sequence-based methods.

2.3.1 Character Classification Based Methods

Bissacco et al. (2013) picked Histogram of Orient Gradient features to train a deep neural network for character classification. The model also adapted a two-level language model to provide a soft-scoring signal, not limited to dictionary words. Jaderberg et al. (2015) proposed a convolutional architecture engaging a conditional random field (CRF). The model involved two convolutional neural networks (CNNs), one is responsible for predicting a character at each position of the output and the other one detects the presence of the n-grams model. This group of approaches locates characters individually in images and recognizes them subsequently. Complicated language models or heuristic rules are required to join characters into words because of potential absent characters or redundant characters.

2.3.2 Word Classification Based Methods

Methods in this group treat the text recognition as a multi-class classification task where each word presented in the dictionary is a class label. Yang et al. (Yang et al. 2017) proposed an adaptive ensemble of deep neural networks (AdaDNNs), which selects and combines network components at different iterations. Compared to baseline DNNs, AdaDNNs increases the accuracy by 8.06% for the ICDAR 2015 dataset. Kang et al. (2016) proposed a context-aware convolutional recurrent neural network for word recognition. The end-to-end pipeline makes use of the metadata in a social network posts such as the title, tags, comments to further improve the recognition rate. The authors showed that using contextual data could boost the recognition accuracy by 3.1%. The above methods perform well if all the words were known in advance. In other words, they can only recognize words that are well-defined in the dictionary. However, the accuracy of the models suffered when the input image is too big.

2.3.3 Sequence Based Methods

Character segmentation is considered a difficult task due to the complex of personal handwriting styles and paper conditions. Thus, to prevent the low-quality outputs of the segmentation phase from affecting the accuracy of the whole system, recent work treats the handwritten text recognition as a sequence recognition problem. In particular, the approaches in this group employ the following network architecture CNN – BiLSTM – Transcription Layer (Shi et al., 2017).

Currently, there are two transcription techniques that dominate this group of approaches: Connectionist Temporal Classification technique and Attention Encoder – Decoder technique. On the one hand, Connectionist Temporal Classification (CTC) transforms the network output $Y = y_1, y_2, \dots, y_T$ into a conditional probability distribution over label sequences $P(L|Y)$. The first application of CTC in handwritten text recognition system was implemented by Graves et al. (2006). But it was only until Shi et al. (2017) presented a novel network architecture for image-based sequence detection that CTC became the standard solution in text recognition.

On the other hand, the Attention-based mechanism initially aims to improve the performance of machine translation system, and later became popular in the deep learning field including text recognition. To a certain degree, the text recognition process is similar to the process of recognizing continuous speech. Therefore, attention-based mechanism such as the Attention Encoder – Decoder technique could be used in HTR to make the machine aware of the region the text coming from.

Lee et al. (2016) introduced recursive RNNs with attention modelling for dictionary-free text recognition. The model uses convolutional layers to extract the features from the input image, and then decodes them into predefined character classes. Soft-attention mechanism enables the model to select features flexibly for end-to-end training.

Table 1 below summarizes the pros and cons of the three text recognition methods. We consider word-based classification the most suitable for dealing with Vietnamese text because many Vietnamese characters come with tonal symbols (e.g., ‘ă’, ‘â’ etc.). In this work, we implement a word-base classification technique to recognize individual Vietnamese, and further utilizing a sequence-based method to improve the text recognition accuracy. The details are presented in section 3.

	Principle	Advantage	Disadvantage
Character-based	Segment individual character and subsequently recognizes every character	Insensitive to font variation	Complex language model is necessary Heavily depend on the character separation
Word-based	Treat the problem as a multi-class classifier and label the word	Effectively recognize words with massive number of labels	Dictionary is required Long arbitrary sequence degeneracy
Sequence-based	Recognize characters based on surrounding connections	Does not require segmentation at the character-level. Could process long strings	High complexity involved in tuning network

Table 1. Comparison of three different frameworks of text recognition method

2.4 Post-processing

The objective of post-processing in an OCR pipeline is to detect and correct linguistic errors from the recognized output of the previous module. Similar to other languages, Vietnamese spell checking involves two steps: error detection and generating suggestion candidates. Nguyen et al. (2008) introduced an approach for Vietnamese spell checking. The authors used a modified Edit Distance and SoundEx algorithms to calculate the score of the suggestion candidates. The advantage of the solution is that the weight function is derived from various factors and the adaptability with various type of document. On the flip side, the approach required high computing computer, large memory and a sufficiently large corpus.

3 Proposed Solution

We develop a modular pipeline (as shown in Figure 2) so that different implementations of specific components can be flexibly swapped and tested. A modular design also makes our image processing pipeline more scalable and robust for future implementations.



Figure 2. Pipeline for the medical record processing

3.1 Pre-processing

For the binarization process, we implement the local thresholding method by Sauvola et al. (1997) given by the following formula:

$$T_{sauvola} = m * (1 - k * (1 - S/R))$$

Where m is the mean of pixels under the window area, S is the dynamic range of variance, and k with value between $[0 - 1]$. We pick this method because the contrast between the text and background in the document is apparent.

3.2 Text Localisation

3.2.1 Identify the Writing Area

To extract the writing area of the medical record, we need to be aware of the document layout. In Figure 3, the blue area indicated the region of interest, which we extract using a greedy algorithm. Given a list of contours C_1, C_2, \dots, C_n from image X , we used a greedy approach to detect the writing area. The writing area is obtained by accumulating adjacent contours until the predefined threshold is reached. The algorithm makes use of the knowledge about the layout of the medical note and it works well even if the doctor drew a separate line across the frame by themselves. The only downside of this method is that if the doctor's handwriting is out of the designated text-area, the algorithm fails to catch the that part of the text. We plan to address this limitation in future work.

Algorithm 1: Pseudocode of extracting writing area algorithm

Data: Input image

Output: A list of candidate contours for writing areas

```

contours = get_contours_from_image()
contours.sort_by_area()
candidate_contours = [ ]
for contour in contours do
    if len(candidate_contours) then
        candidate_contours.append(contour)
        continue;
    if is_adjacent_contour(candidate_contours) then
        candidate_contours.append(contour)
    if get_percentage_area(candidate_contours) >= THRESHOLD then
        Break
  
```

3.2.2 Skewness Correction

Figure 3 shows a case where the alignment of text in the image is skewed. Therefore, a perspective warp is necessary here to obtain the writing area in a correct dimension. Such operation is carried by first extracting the 4 corner points of the rectangular writing area, then computing the homography and finally perform appropriate transformation to get the correct perspective.

3.3 Text Segmentation

3.3.1 Line Segmentation

After successfully extract writing areas, we perform line segmentation. Because the paper-based medical records have dotted lines, we projected the pixels from the image horizontally to transform a dotted line into a completely connected line. Before the denoising phase, every line is detected and normalized.

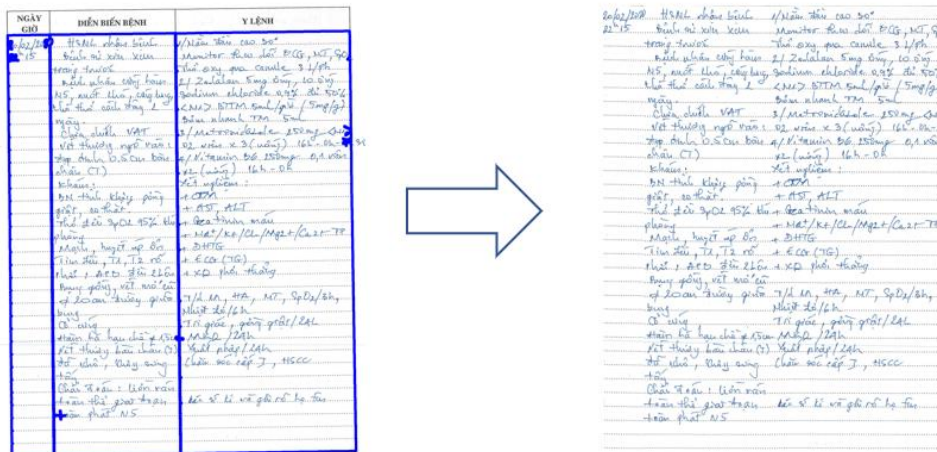


Figure 3. An illustration of perspective warping operation

3.3.2 Line denoising

Figure 4 depicts two types of noise that could be found in a detected line: (1) dots appears in the handwriting and (2) written text overlapped with the line below. To improve the overall accuracy of the image processing pipeline, noises like those need to be cleaned. The process consists of the three steps:

- Normalizing the line image
- Locating the contour location of the noise
- Masking the noises using a bitwise operation.

As for the dots, we look for the contours that has perimeter and area matched the dot's values. For the overlapped text, we look for the contour that starts from the beginning of the image and with the perimeter smaller than a defined threshold.

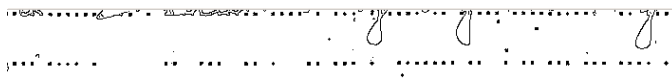


Figure 4. A mask of the noises detected from the image

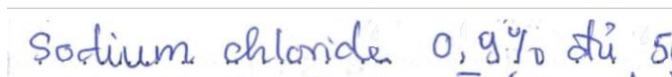


Figure 5. The output image of the denoising operation

3.3.3 Word segmentation

Our word segmentation utilized an implementation of the scale-space segmentation technique proposed in Manmatha (1999). The scale-space mentioned that the size of the object is dependent on the distance to the camera. Therefore, a visual system should be able to spot the object at every possible scale. By applying an anisotropic filter kernel to the image, we generate the blobs corresponding to individual words. After thresholding the blob-image, connected components representing each word are extracted. The method is independent of the writing style and does not require prior training. However, the method also requires that the word-spacing to be clear.

3.4 Text Recognition

3.4.1 Network Architecture

The text recognition module is a combination of ResNet-LSTM-CTC. We discuss each component of this architecture in the subsections below. In brief, the network operates as follow.

- The input image needs to be padded-and-normalized.
- The processed image is fed into a convolutional network for feature extraction.
- The extracted features are mapped into a feature sequence.

- Finally, the feature sequence is sent to a transcription layer whose mission is to determine the label with the highest probability for the map.

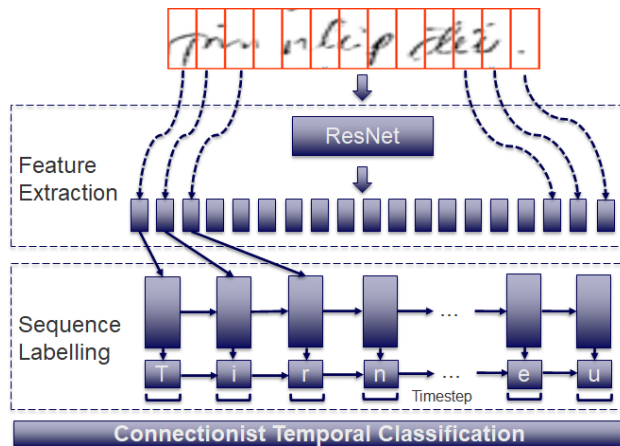


Figure 6. The Text Recognition Network Architecture

3.4.2 Network Input

The network takes a 240x32 grayscale image as the input. Any input image with different scales will be normalized by an operation called padding-normalized. First, the image aspect ratio is calculated by dividing the width over the height. Then the result is compared to the acceptance ratio δ (width/height). Based on this ratio, we decide whether we should pad vertically or horizontally. The reason for taking such action is that we want to keep the aspect ratio of the image, which persists the writer's writing style.

3.4.3 Feature Extraction

In this stage, a CNN extracts a feature map $M = \{m_0, m_1, \dots, m_i\}$ with i corresponding to the number of character class. Each column in the resulting feature mapped by a feature extractor has a corresponding distinguishable receptive field along the horizontal line of the input image. These features are used to estimate the character for each receptive field (Baek et al. 2019).

Training a deep neural network could be challenging because the accuracy could get saturated and worsen rapidly. To tackle such challenge, He et al. presented a deep residual learning network (ResNet) (He et al. 2016), whose building block is expressed as:

$$y = F(X, W_i) + x$$

Where x and y are the input and output vector of the layer considered, and $F(X, W_i)$ is the residual mapping to be learned. To ease the difficulty in training deeper CNNs, we used a ResNet architecture (Cheng et al. 2017) as the backbone for feature extraction.

3.4.4 Sequence modelling

The extracted feature map M from the previous stage is then rearranged into a sequence of features V . Every column presented in the map M is considered as a sequence frame. However, this frame usually is affected by missing contextual information; thus, we use a LSTM network tackle the problem. Because a typical LSTM only considers past events, while context information from both directions are necessary in our approach, we need to stack 2 LSTM networks so we could cover both directions.

3.4.5 Transcription module

The transcription module uses a method called Connectionist Temporal Classification to take the per-frame predictions to a string label. Here, Best Path Decoding is used to decode the feature sequence into the ground truth. The algorithm takes the most probable character per time-step forming the 'best path' and then perform a collapsing function to the path. From there, a predict label is generated.

3.5 Post-processing

The predicted text from the previous steps is processed by a spell corrector module. This module has the responsibility to eliminate non-word errors and reduce the amount of grammar errors. For example, an operation like elevating the head of the patient to 30 degrees could be expressed fully in Vietnamese as:

Nâng^{Elevate} *đầu*_{của} *bệnh nhân*^{head of the patient} *cao*^{high} *30*_{độ}^{thirty degrees}.

Most of the doctors would remove the entity and shorten, results in the following sentence:

Nằm^{Rest} *đầu*^{head} *cao*^{high} *30*[°]_{thirty degrees}.

As sentences in the record only follows Vietnamese grammar partially, it would be more accurate to create a new post-processing method particularly for the medical records.

3.5.1 Corpus and tokenization

Corpus is a large and structured set of texts. It is used to validate linguistic rules on a specific language. Our corpus contains approximately 3000 Vietnamese words and was built manually based on the Official Diagnosis and Treatment Guidelines for Infectious Diseases (Vietnamese Ministry of Health, 2005). With the introduction of tonal symbols, the meaning of the word could evolve in very different ways. Given the word ‘cao’ (means “high”), with tonal symbols, we could obtain 4 more variations: cáo (fox), cạo (shave), cào (scratch), cảo (dumpling). There are also some grammatically incorrect variations such as “cào”, “cảo”, however, different Vietnamese word has different correct variations to it.

The input text line is broken into words and normalized at this post-processing step: all word units are converted to lower case. Since our recognition module is performed at word level, this makes the process of tokenization more straightforward.

3.5.2 Detection of errors

The step computes the modified version of Levenshtein distance (Levenshtein, 1965) using the vocabulary from the above corpus. A Vietnamese word could be broken into 5 parts for collation:

Initial consonant + Vowel + Optional end consonant + Tone

Although there is a large amount of possible combinations, only a portion of them are grammatically correct. Consider the word ‘luat’. The only possible correct combination of the mentioned word is ‘luật’, even though for the letter ‘a’, there are potentially three variations (‘a’, ‘ă’, ‘â’). By exploiting such Vietnamese language rules, we developed a set of heuristics by applying a custom Weighted Edit Distance to generate potential correct combinations. Figure 7 depicts scheme used in this work. The output of this stage is a set of possible combinations of initial clause.

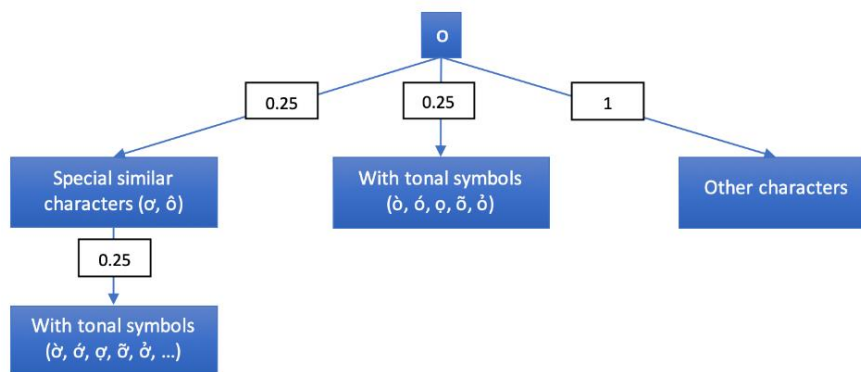


Figure 7. Custom Levenshtein Distance

3.5.3 Bigrams Language Model

 → **Ground Truth:** “1/ *Nằm*^{Raised} *đầu*^{head} *cao*_{to (in term of height)} *30*[°]”
Recognized : “1/ *Nằm*_{Five} *đầu*_{hurt} *cao*_{to(in term of height)} *30*[°]”

Figure 8. Recognized Text Output

After obtaining the output from the CRNN model, we proceed to tokenize the recognized sequence. The above recognized output could be tokenized as:

< Number > *Nằm*_{Five} *đầu*_{hurt} *cao*_{to(in term of height)} < Number >

From there, we applied our custom Edit Distance and generate a matrix of candidates with their edit distance against the recognized word.

Word	Probable candidates with Weight Distance
Năm	Năm (0.25), Năm (0.25), Nam (0.25), Năm (0.5)
đau	đau (0.25), đau (0.25), đau (0.5), đau (0.5)
cao	cao (0.25), cáo (0.25), cào (0.25), cáo (0.25)

Table 2. Possible candidates for every recognized word

For every candidate generated at the former step, the probabilities of the possible correction with the left word (if the correction is not at the first position) and the right word (if the correction is not at the last position) are calculated. The combination with highest probabilities is introduced into the new output text. In this case, the most probable combination of the words is: “Năm đau cao”.

We acknowledge that the tonal symbols of the Vietnamese language can lead to erroneous detection of the handwritten text, for example na’ m versus nam. Especially, inputs containing incorrect grammar could lead to such an error, since our solution predicts tonal symbols based on the grammar structure of the sentence. Despite being simple, this algorithm proved to be an effective mean to reduce the false corrections.

4 Experiments and Results

4.1 Dataset

The system was trained and evaluated with the dataset provided by the Oxford University Clinical Research Unit (OUCRU - <http://www.oucr.org/>). The dataset included handwritten ICU records by 12 doctors plus a set of clinical words commonly used at the Hospital of Tropical Diseases (<http://benhnhietdoi.vn/>). Note that no clinical data or patient records were used for this project. The dataset was split into a train set (validation included) and a test set with a ratio of 80:20. To optimize the training process, all images and labels are packaged into a Lightning Memory-Mapped Database (LMDB) (Chu, 2011) to utilize the disk transfer speed.

4.2 Training Configurations

The text recognition module is implemented using the Torch framework (Paszke, 2019). Our implementation bases on CUDA and CUDNN technologies to take advantage of GPU acceleration. Our experiments were executed on the same environment: Intel I5 7600k CPU, GTX 1060 6GB, and 32GB of RAM under the Python 3.7 environment.

Training Configurations for experiments	
Optimizer	Adadelata
Iteration count	3500
Validation interval	100
Batch size	32
Decay rate	0.95
Gradient clipping	5
Feature extraction output	512
BiLSTM hidden size	512
Character class count	76

Table 3. Model training configuration

4.3 Results

4.3.1 Training results analysis

In Figure 10, we observe the validation loss graph oscillated until the 2300th iteration during training, even though the CER and WER rate gradually dropped (Figure 9). After the 2300th iteration, the model already picked up the relevant features for recognizing Vietnamese words provided denoised lines on from the training images. The loss function flats out in the last 300 iterations which suggests the termination for the training phase.

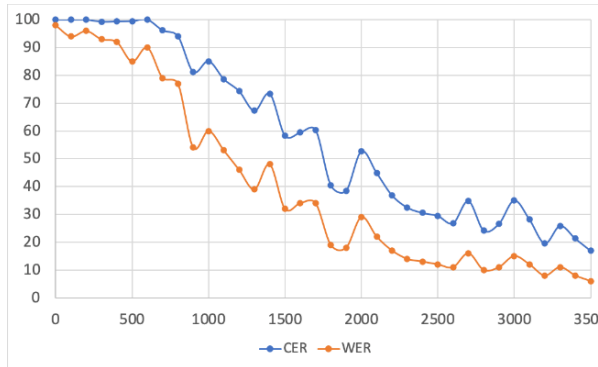


Figure 9. CER and WER over iteration graph



Figure 10. Loss over iteration graph

4.3.2 Baseline RCNN versus ResNet

We replaced the feature extraction module with a RCNN of an OCR model (Lee et al. 2016) in order to introduce a baseline to compare our text recognition model with. Following that, we trained the whole pipeline using the same configuration as the ResNet's (Cheng et al. 2017) and reported the results below:

	Validation		Test		Trainable parameters	Time
	CER	WER	CER	WER		
CRNN	4.37%	12.01%	8.47%	22.85%	11,354,700	3.3 hours
ResNet	6.16%	16.95%	6.47%	19.14%	53,758,060	5 hours

Table 4. The comparison of above training results

As we can see from Table 4, the CRNN network is much faster to train compare to the ResNet one. With the same training configuration, the CRNN managed to reach a WER of 12% on the validation set, which is 4.94% higher than the ResNet. However, the accuracy on the test set of baseline architecture is slightly lower compared to the residual network. In other words, the proposed ResNet architecture generalized better than baseline architecture. Because ResNet architecture has 5 times more trainable parameters than RCNN one, the total training time is much longer. It is also expected that the ResNet model would take a longer time to infer.

4.3.3 With-language model versus without-language model

As a spell corrector is applied to perform grammar correction on the output text, the accuracy of both pipeline with and without the post-processing module is captured. The barebone pipeline without the post-process module achieved a CER and line accuracy of 14.84% and 45.61% respectively. With the use of the module, the CER is lessened by 0.25%. Also, it boosted the line accuracy significantly to 53.55%.

4.3.4 Performance speed

To assess the performance of our proposed pipeline, we measure the overheads (in seconds) of each individual component in our pipeline and report their average results in Table 5 below. Overall, the text recognition part is the most time-consuming part of the pipeline in which ~76% of the total time is spent here (3.78 seconds out of 4.94 seconds). This is expected as the text recognition process also involves the pdf conversion and the disk output tasks. However, we also note that the line denoising tasks more time than other extraction and segmentation tasks. We consider improving the denoising task, as described in section 3.3.2, using some DNN based techniques such as denoising autoencoders in future work. Nevertheless, with the average total processing time of 4.94 seconds per hand-writing record which contains up to 300 Vietnamese words, the pipeline's performance is acceptable.

	Time
Writing area extraction	0.128 second
Line segmentation	0.125 second
Line denoising	0.5188 second
Word segmentation	0.3845 second
Text recognition	3.7839 seconds
Total	4.94 seconds

Table 5. The execution time of the proposed pipeline

5 Conclusion

There has been great advance on handwritten text recognition. However, most existing methods are developed for English language and there is little-to-none dedicated for Vietnamese. Vietnamese handwritten text recognition is fundamentally more challenging than that of English due to the presence of more character classes, complex vocals and tonal symbols. In this paper, we introduced an end-to-end pipeline to perform text recognition on scanned Vietnamese medical records. We tackle those challenges by enhancing various tasks in the text recognition pipeline. For example, we apply denoising process, perform text segmentation to the word level, and apply the Bigram language model to improve the probabilities of the possible correction given the neighbouring words. Most importantly, we combine and implement a deep learning architecture which consists of a ResNet for feature extraction, a BiLSTM network for text sequence modelling, and CTC for final transcription task.

Our research's contributions are twofold. First, by delivering an optical character recognition (OCR) working pipeline, our work contributes to the surging literature on machine learning applications, especially for processing scanned medical records. Our proposed pipeline demonstrates promising results, especially in the context where empirical research on Vietnamese handwritten medical records remains scarce. Second, digitalizing hard copy medical records can lead to more secure preservation of important and confidential patient information. Furthermore, it facilitates the digital transformation of medical centers and hospitals, especially those in developing countries such as Vietnam, and improves their readiness for adopting modern EHR management systems. This is why our work is significant.

We acknowledge a limitation in our current pipeline where the text written outside of the detected writing area could be omitted. The issue is when the text overflows to the next column, the pipeline assumes that it belongs to the next block. As future work, widening the text search area could address this problem.

6 References

- Samadbeik, M., Fatehi, F., Braunstein, M., Barry, B., Saremlan, M., Kalhor, F., and Edirippulige, S. 2020. "Education and Training on Electronic Medical Records (EMRs) for Health Care Professionals and Students: A Scoping Review," *International Journal of Medical Informatics* (142), p. 104238.
- Cuk, S., Wimmer, H., and Powell, L. M. 2017. "Problems Associated with Patient Care Reports and Transferring Data between Ambulance and Hospitals from the Perspective of Emergency Medical Technicians," *Issues in Information Systems* (18:4).
- Marutha, N. S., and Ngoepe, M. 2017. "The Role of Medical Records in the Provision of Public Healthcare Services in the Limpopo Province of South Africa," *South African Journal of Information Management* (19:1), pp. 1-8.
- Andreu-Perez, J., Poon, C. C., Merrifield, R. D., Wong, S. T., and Yang, G. Z. 2015. "Big data for health," *IEEE Journal of Biomedical and Health Informatics* (19:4), pp. 1193-1208.
- Levenshtein, V. I. 1965. "Binary Codes Capable of Correcting Deletions, Insertions, and Reversals," *Soviet physics. Doklady* (10), pp. 707-710.
- Zhou, X., Yao, C., Wen, H., Wang, Y., Zhou, S., He, W., and Liang, J. 2017. "East: An Efficient and Accurate Scene Text Detector," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2642-2651.
- Baek, Y., Lee, B., Han, D., Yun, S., and Lee, H. 2019. "Character Region Awareness for Text Detection," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9357-9366.
- Bissacco, A., Cummins, M. J., Netzer, Y., and Neven, H. 2013. "Photoocr: Reading Text in Uncontrolled Conditions," *2013 IEEE International Conference on Computer Vision*, pp. 785-792.
- Jaderberg, M., Simonyan, K., Vedaldi, A., and Zisserman, A. 2015. "Deep Structured Output Learning for Unconstrained Text Recognition." *International Conference on Learning Representations*, pp. 1-10.
- Yang, C., Yin, X.-C., Li, Z., Wu, J., Guo, C., Wang, H., and Xiao, L. 2017. "Adadnns: Adaptive Ensemble of Deep Neural Networks for Scene Text Recognition," *ArXiv (abs/1710.03425)*.
- Kang C, Kim G, Yoo SI. 2017. "Detection and recognition of text embedded in online images via neural context models," *Proceedings of association for the advancement of artificial intelligence*, pp 4103-4110.

- 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 Transactions on Pattern Analysis and Machine Intelligence* (39), pp. 2298-2304.
- Graves A, Gomez F. 2006. "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," *International conference on machine learning*, pp 369–376.
- Lee, C.-Y., and Osindero, S. 2016. "Recursive Recurrent Nets with Attention Modeling for Ocr in the Wild," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2231-2239.
- Nguyen, P., Ngo, T., Phan, D. A., Dinh, T., and Huynh, T. H. 2008. "Vietnamese Spelling Detection and Correction Using Bi-Gram, Minimum Edit Distance, Soundex Algorithms with Some Additional Heuristics," *2008 IEEE International Conference on Research, Innovation and Vision for the Future in Computing and Communication Technologies*, pp. 96-102.
- Sauvola, J. J., Seppänen, T., Haapakoski, S., and Pietikäinen, M. 1997. "Adaptive Document Binarization," *Proceedings of the Fourth International Conference on Document Analysis and Recognition* (1), pp. 147-152 vol.141.
- Manmatha, R., and Srimal, N. 1999. "Scale Space Technique for Word Segmentation in Handwritten Manuscripts."
- Baek, J., Kim, G., Lee, J., Park, S., Han, D., Yun, S., Oh, S. J., and Lee, H. 2019. "What Is Wrong with Scene Text Recognition Model Comparisons? Dataset and Model Analysis," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4714-4722.
- He, K., Zhang, X., Ren, S., and Sun, J. 2016. "Deep Residual Learning for Image Recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770-778.
- Cheng, Z., Bai, F., Xu, Y., Zheng, G., Pu, S., and Zhou, S. 2017. "Focusing Attention: Towards Accurate Text Recognition in Natural Images," *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 5086-5094.
- Vietnamese Ministry of Health. 2015. "Official Diagnosis and Treatment Guidelines". Retrieved November 2, 2020, from <http://kcb.vn/vanban/huong-dan>.
- Paszke, A., et al. 2019. "PyTorch: An Imperative Style, High-Performance Deep Learning Library." *ArXiv* abs/1912.01703.

Copyright

Copyright © 2020 authors. This is an open-access article licensed under a [Creative Commons Attribution-NonCommercial 3.0 New Zealand](https://creativecommons.org/licenses/by-nc/3.0/), which permits non-commercial use, distribution, and reproduction in any medium, provided the original author and ACIS are credited.