

Đại học Quốc gia Thành phố Hồ Chí Minh
Trường Đại học Công nghệ Thông tin
Khoa Khoa học máy tính



Báo cáo cuối kỳ

VIETNAMESE RECEIPT EXTRACTION

Giảng viên hướng dẫn

ThS. Đỗ Văn Tiến

Sinh viên thực hiện

Nguyễn Khả Tiến - 19522337

Nguyễn Lê Anh Quân - 19522081

Nguyễn Thành Vinh

Lớp

CS532.M21.KHCL

Ho Chi Minh City - 2022

Mục lục

1	Tổng quan đề tài:	2
1.1	Sơ lược về đề án:	2
1.2	Mục tiêu của đề án:	2
1.3	Ngữ cảnh bài toán:	2
1.4	Thách thức bài toán	3
2	Xây dựng bộ dữ liệu	3
3	Các bước thực hiện chính và quá trình huấn luyện mô hình	4
3.1	Phát hiện hóa đơn, cắt phần hóa đơn	4
3.1.1	Phát hiện hóa đơn	5
3.1.2	Xoay và cắt hóa đơn	6
3.2	Phát hiện văn bản (Paddle):	6
3.3	VietOCR	7
3.3.1	CNN Của Mô Hình OCR:	7
3.3.2	Laguage model: AttentionOCR	8
3.3.3	Huấn luyện mô hình VietOCR:	10
3.4	Model PICK (KIE)	11
4	Kết quả huấn luyện và đánh giá mô hình:	12
5	Thiết kế phần mềm:	14
6	Kết luận	15
	Tài liệu tham khảo	16

1 Tổng quan đề tài:

1.1 Sơ lược về đề án:

Đề án này hướng đến tìm hiểu quy trình và thực hiện một bài toán Information Extraction (Rút trích thông tin) nói chung và các bài toán con trong đó nói riêng. Cụ thể hơn là đề án này sẽ tìm hiểu và thực hiện quy trình nhận diện và trích xuất thông tin văn bản từ ảnh chụp hóa đơn tiếng Việt trong cuộc sống bằng thiết bị di động.

1.2 Mục tiêu của đề án:

Để thực hiện được đề án, chúng tôi đã đặt ra những mục tiêu như sau:

- Hiểu và nắm rõ ý tưởng cũng như vấn đề cần giải quyết của các bài toán con
- Huấn luyện và đánh giá được một số mô hình con trong pipeline. Từ đó rút ra được kết luận và nguyên nhân cho kết quả đó để phục vụ cho công việc cải tiến sau này.
- Xây dựng được ứng dụng demo Information Extraction trong thực tế.


1.3 Ngữ cảnh bài toán:

Các hóa đơn thường mang lại các thông tin cần thiết cho các công ty, và đa số chúng thuộc dạng giấy hoặc là các văn bản dạng điện tử như tệp tin PDF hoặc dạng ảnh. Để quản lý những thông tin này một cách hiệu quả, các công ty trích xuất và lưu trữ chúng vào cơ sở dữ liệu của họ. Thêm vào đó các hóa đơn cũng thường mang lại thông tin rất cần thiết trong việc quản lý chi tiêu của một cá nhân, và đa số người dùng thường nhận được hóa đơn ở dạng giấy hoặc dạng ảnh. Để quản lý thông tin chi tiêu thông qua hóa đơn một cách hiệu quả, OCR (optical character recognition) là một công nghệ phục vụ nhận diện chữ viết trong ảnh một cách hiệu quả và tự động.

Việc trích xuất được thông tin từ hóa đơn 1 cách tự động có thể giúp người dùng chỉ cần chụp hình 1 hóa đơn và thông tin cần thiết sẽ được tự động lưu lại và đưa vào cơ sở dữ liệu của người dùng đó. Nhờ vậy mà người dùng khỏi phải lặn lội việc nhập hóa đơn 1 cách thủ công nữa.

Input/Output bài toán

- Input: Ảnh có chứa hóa đơn Tiếng Việt.
- Output: Các trường thông tin như trong hình.
 - SELLER: tên của cửa hàng
 - ADDRESS: địa chỉ của cửa hàng
 - TIMESTAMP: thời gian xuất hoá đơn
 - TOTAL_COST: tổng chi phí có trong hoá đơn

Input	Output (1 line, 4 fields, separated by).
	SCTC CỔ THỎ 104 TRẦN PHÚ – CẨM PHẢ 104 Trần Phú – phường Cẩm Tây – Thành phố Cẩm Phả Ngày 09/08/2020 (7:44 CH 7:44CH) Tổng thanh toán 115.000đ

Hình 1: Input và output bài toán

1.4 Thách thức bài toán

- Bài toán còn gặp nhiều vấn đề nan giải bởi các tác động bên thứ 3 như: ảnh hưởng độ sáng, méo mó, nghiêng nặng. . .
- Các tờ hóa đơn không nằm ở một vị trí nhất định trong ảnh và không phải lúc nào cũng thẳng, đều và dễ đọc, kể cả việc mỗi hóa đơn có kích thước khác nhau.
- Bài toán xử lý trên dữ liệu tiếng Việt nên có thể bị ‘bỏ sót’ các dấu thanh (‘sắc’, ‘huyền’, ‘hỏi’, ‘ngã’, ‘nặng’) vì chúng có kích thước nhỏ hơn so với chữ.

Mặc dù những khó khăn này, các thuật toán deep learning ngày nay hoạt động tương đối tốt và càng phát triển.

2 Xây dựng bộ dữ liệu

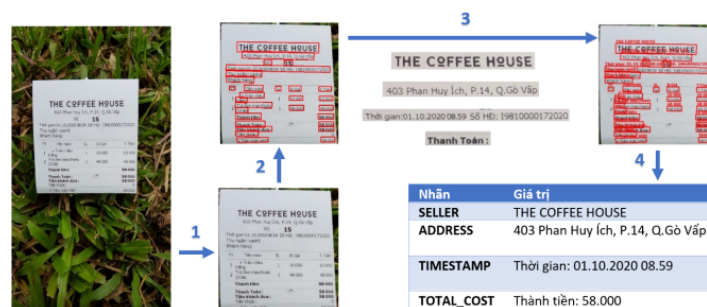
Sử dụng bộ dữ liệu MCOCR-2021 [1] để phục vụ huấn luyện), từ cuộc thi RIVF2021 của Việt Nam bao gồm: 1000 ảnh huấn và 400 ảnh kiểm thử. Bộ dữ liệu bao gồm những tấm ảnh chứa hoá đơn bằng Tiếng Việt được chụp từ điện thoại. Mỗi hóa đơn sẽ được gán nhãn 4 trường thông tin như đã đề cập ở mục 1.3



Hình 2: Dữ liệu mẫu từ ban tổ chức

3 Các bước thực hiện chính và quá trình huấn luyện mô hình

Pipeline của nhóm sẽ có 4 bước chính như hình vẽ dưới đây:



Hình 3: Pipeline quá trình trích xuất thông tin hóa đơn

1. Phát hiện hóa đơn, cắt phần hóa đơn.
2. Phát hiện văn bản (Paddle)
3. Nhận diện văn bản (VietOCR)
4. Trích xuất thông tin (PICK)

3.1 Phát hiện hóa đơn, cắt phần hóa đơn

Vì đầu vào của bài toán là ảnh chụp hóa đơn với bất kỳ điều kiện về ánh sáng, góc độ hoặc ngoại cảnh. Những yếu tố này có thể gây ra những nhầm lẫn không đáng có khi thực hiện việc phát hiện văn bản (ví dụ những background phức tạp, có nhiều nhiễu). Ngoài ra góc độ cũng là một trong những yếu tố ảnh hưởng đến hệ thống. Vì vậy, bước phát hiện hóa đơn và cắt phần hóa đơn đóng

một vai trò quan trọng trong việc xử lý các trường hợp khó, cũng như tránh gặp phải các nhầm lẫn không mong muốn.

Với các bước thực hiện chính như sau:

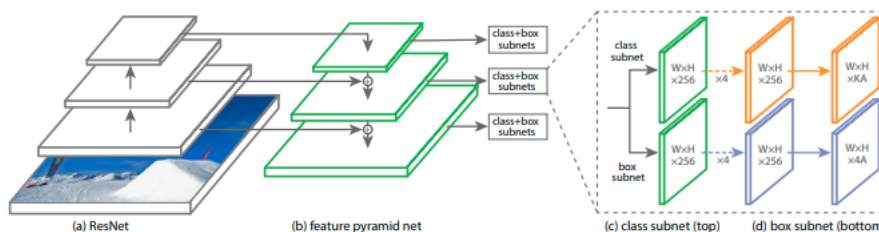
- Bước 1: Xác định vị trí của hóa đơn trong ảnh, sử dụng một object detection để thực việc này.
- Bước 2: Xác định được góc nghiêng của hóa đơn. Nhóm tiến hành sử dụng thư viện Tesseract [2] để xác định thành phần này. Trong đó, theo quan sát của bộ dữ liệu chúng ta có 4 góc nghiêng là:
 1. Góc bình thường
 2. Nghiêng 90 độ theo chiều kim đồng hồ.
 3. Nghiêng 90 độ theo chiều ngược kim đồng hồ.
 4. Nghiêng 180 độ.



Hình 4: Các bước xử lý trong bài toán phát hiện hóa đơn.

3.1.1 Phát hiện hóa đơn

Trong bài toán này, nhóm thực hiện việc nhận diện vị trí hóa đơn bằng kiến trúc RetinaNet [3]. Đây là kiến trúc thiết kế theo hướng One-Stage nên có thể đảm bảo được về độ chính xác cũng như là thời gian xử lý cho mỗi ảnh nhanh với tốc độ cao.



Hình 5: Kiến trúc của Retinanet

3.1.2 Xoay và cắt hóa đơn

Sử dụng thư viện Tesseract [2], nhóm thực hiện xác định và xoay hóa đơn thông qua API có sẵn của thư viện này đó là *image_to_osd*. Ví dụ như sau:

```
imPath='path_to_image'
im = cv2.imread(str(imPath), cv2.IMREAD_COLOR)
newdata=pytesseract.image_to_osd(im)
re.search('(?!<Rotate: )\d+', newdata).group(0)
```

```
----
Output:
Page number: 0
Orientation in degrees: 270
Rotate: 90
Orientation confidence: 21.27
Script: Latin
Script confidence: 4.14
----
```

3.2 Phát hiện văn bản (Paddle):

Sau khi nhận được kết quả output từ bước trước. Bước cần làm tiếp theo đó chính là **phát hiện được vị trí các vùng chữ có trong hóa đơn**. Ở bước này nhóm sử dụng lại module có sẵn text detection của Paddle [4].

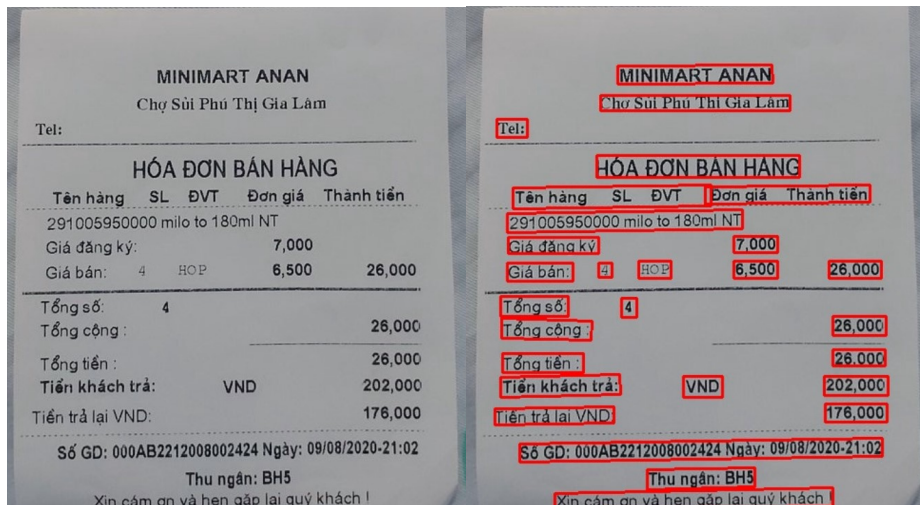


Hình 6: Paddle OCR

Mô hình sẽ nhận dữ liệu đầu vào là một ảnh chứa hóa đơn đã được crop và có kết quả đầu ra như dưới đây.

PaddleOCR là một mô hình OCR, nó hỗ trợ cả hai task là detect và recognize văn bản có trong hình ảnh. Mô hình này đã có nhiều thành tựu về độ hiệu quả trong bài toán OCR. Những ưu điểm của mô hình này:

- Dung lượng khá nhẹ (3.5-4 Mb): Được xem là mô hình phát triển thiết bị đầu cuối nhẹ nhất.
- Độ linh hoạt: Có thể áp dụng trên nhiều ứng dụng (vé tàu, biển số xe, hóa đơn,...).
- Độ chính xác: Độ đo F1 sẽ lớn hơn 0.5 (F1 score > 0.5), gấp 2 lần độ đo những mô hình OCR khác.



Hình 7: Input

Hình 8: Output

- Độ đa dạng: Có thể detect được hơn 27 ngôn ngữ kể cả tiếng Việt.

Tuy nhiên nhóm chúng tôi chỉ sử dụng phần Text detection của module này để phát hiện văn bản trong hóa đơn.

3.3 VietOCR

Sau khi phát hiện được các vùng chứa văn bản, công đoạn tiếp theo sẽ là nhận diện văn bản. Ở đây [5] - một framework nhận dạng văn bản tiếng Việt hoạt động ổn định và hiệu quả.

VietOCR là mô hình được tạo ra nhằm giải quyết các bài toán liên quan đến OCR, đặc biệt là nó rất hiệu quả đối với những bài toán nhận diện chữ tiếng Việt và do người Việt làm ra.

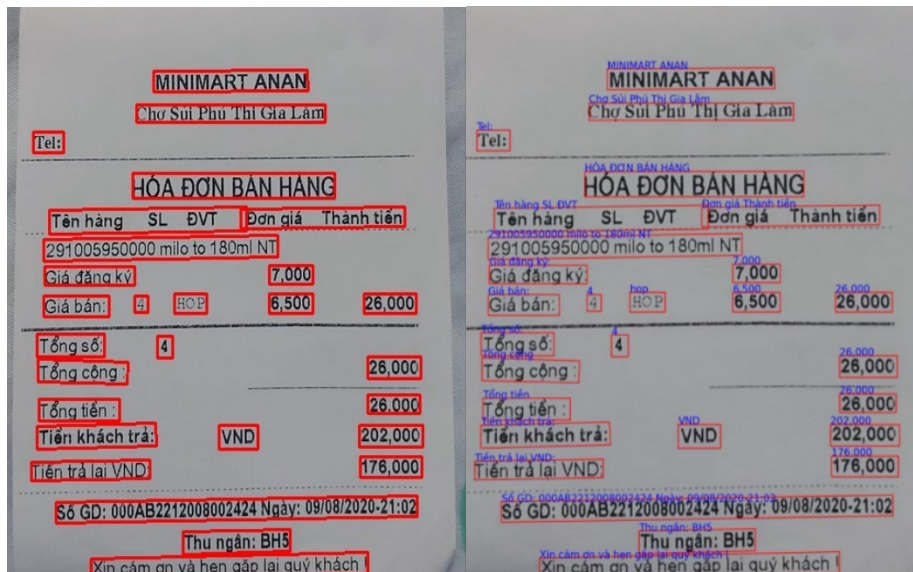
Mô hình VietOCR là sự kết hợp giữa mô hình CNN (VGG-19) và Language Model, sẽ có 2 mô hình OCR để giải quyết:

- Attention-OCR.
- Transformer-OCR.

Mô hình sẽ nhận dữ liệu đầu vào là một ảnh chứa hóa đơn đã được detect vùng chữ và kết quả đầu ra của mô hình được thể hiện như hình dưới đây.

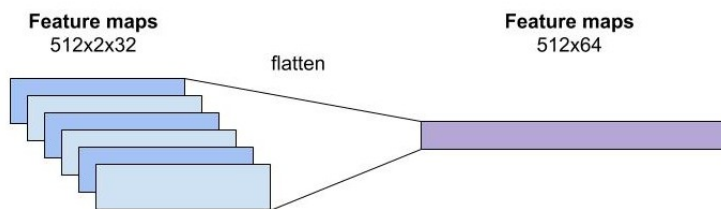
3.3.1 CNN Của Mô Hình OCR:

Mô hình CNN dùng trong bài toán OCR nhận đầu vào là một ảnh, thông thường có kích thước với chiều dài lớn hơn nhiều so với chiều rộng, do đó việc điều chỉnh tham số stride size của tầng pooling là cực kì quan trọng. Thường thì kích thước stride size của các lớp pooling cuối cùng là $w \times h = 2 \times 1$ trong mô hình OCR. Không thay đổi stride size phù hợp với kích thước ảnh thì sẽ dẫn đến kết quả nhận dạng của mô hình sẽ tệ. Đầu ra của VGG sẽ làm đầu vào cho mô hình **Language model** phía sau.



Hình 9: Input

Hình 10: Output



Hình 11: Đặc trưng sau khi được trích xuất từ VGG

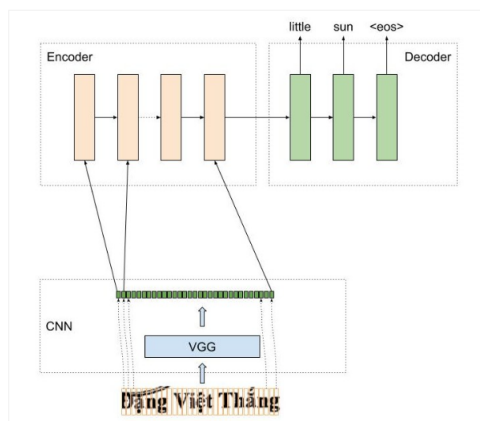
3.3.2 Language model: AttentionOCR

AttentionOCR là sự kết hợp giữa mô hình CNN và mô hình Attention Seq2Seq. Cách hoạt động của mô hình này tương tự như kiến trúc của mô hình seq2seq trong bài toán dịch máy. Với bài toán dịch máy từ tiếng việt sang anh, chúng ta cần encode một chuỗi tiếng việt thành một vector đặc trưng, còn trong mô hình AttentionOCR, thì dữ liệu đầu vào này là một ảnh.

Một ảnh qua mô hình CNN, sẽ cho một feature maps có kích thước $channel \times height \times width$, feature maps này sẽ trở thành đầu vào cho mô hình LSTM, tuy nhiên, mô hình LSTM chỉ nhận chỉ nhận đầu vào có kích thước là $hidden \times timestep$. Một cách đơn giản và hợp lý là 2 chiều cuối cùng $height \times width$ của feature maps sẽ được duỗi thẳng. Feature maps lúc này sẽ có kích thước phù hợp với yêu cầu của mô hình LSTM. Feature maps của mô hình CNN sau khi được flatten thì được truyền vào làm input của mô hình LSTM, tại mỗi thời điểm, mô hình LSTM cần dự đoán từ

tiếp theo trong ảnh là gì [5].

subsubsectionLanguage model: TransformerOCR Trước khi đi đi vào mô hình



Hình 12: VietOCR với language model AttentionOCR

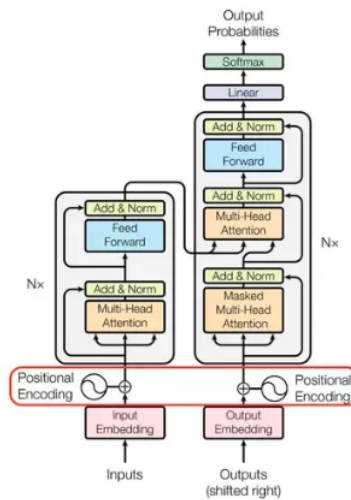
TransformerOCR, chúng tôi thực hiện phân tích một số nhược điểm của các mô hình RNN (LSTM) như sau:

- Thời gian huấn luyện lâu: Khi xử lý một câu văn bằng RNN thì mô hình xử lý câu văn một cách tuần tự theo từng timestep do đó hidden state sau phải phụ thuộc vào hidden state trước thực hiện xong mới tính toán được. Điều này khiến mô hình không thể thực hiện tính toán song song, không tận dụng được khả năng tính toán của GPU khiến thời gian training lâu hơn nhiều so với cấu trúc như CNN.
- Khả năng ghi nhớ kém : Đây là vấn đề muôn thuở đối với mạng có kiến trúc tuần tự như RNN. Nói đơn giản là mô hình sẽ chỉ học được các từ ở đầu câu, càng về sau những đặc trưng học được càng ít do gradient biến mất (vanishing gradient).

Transformer giải quyết được nhược điểm của mô hình tuần tự truyền thống nhờ chủ yếu vào hai cấu trúc là Multi-head attention - Positional encoding.

Transformer cũng đang là mô hình SOTA hiện tại trong hầu hết các bài toán từ thị giác cho đến xử lý ngôn ngữ. Mô hình này giúp việc trích xuất văn bản từ ảnh một cách hiệu quả. Kiến trúc transformer cũng giống với các mô hình sequence-to-sequence bao gồm hai phần encoder và decoder.

- Phần Encoder, Gồm N block, mỗi block bao gồm hai sub-layer: Multi-Head Attention và Feed forward network. Tác giả dùng một residual connection ở mỗi sub-layer này. Theo sau mỗi sub-layer đó là một lớp Layer Norm có ý nghĩa tương tự như lớp Batch Norm trong CNN. Residual connection cũng góp phần giúp mô hình có thể sâu hơn , deep hơn nhờ giảm tác động của vanishing gradient.
- Phần Decoder cũng tương tự như encoder gồm N block, mỗi block gồm 2 sub-layer. Tuy nhiên, nó có một lớp Masked Multi-Head Attention. Lớp



Hình 13: Kiến trúc Transformer

này chính là lớp Multi-Head Attention. Nó có chức năng chú ý đến toàn bộ những decoder hidden state trước. Lý do mà nó lại được đặt tên như vậy là khi huấn luyện Transformer, ta đưa toàn bộ câu vào cùng một lúc nên nếu ta đưa toàn bộ target sentence cho decoder trước thì mô hình sẽ chẳng học được gì cả. Do đó phải che (mask) bớt một phần token ở decoder hidden state sau trong quá trình decode.

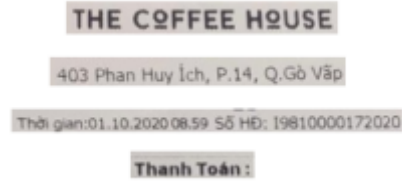
3.3.3 Huấn luyện mô hình VietOCR:

Ở trong đề án lần này, chúng tôi sử dụng pretrained từ repo VietOCR của tác giả Phạm Quốc. Pretrained weight đã được huấn luyện với 10 triệu ảnh, một số lượng rất lớn. Tuy nhiên, theo tác giả thì mô hình **khá nhạy cảm với sự thay đổi nhỏ của ảnh đầu vào** khi sử dụng pretrained model trên tập dữ liệu mới chưa được huấn luyện. Do đó chúng tôi sẽ dùng pretrained này nhưng custom lại với dữ liệu của chúng tôi tạo ra.

Về phần dữ liệu cho VietOCR. Do là để nhận diện tốt trên domain hóa đơn, nên chúng tôi sẽ khai thác từ những ảnh hóa đơn của ban tổ chức. Chúng tôi dùng model có sẵn là Paddle như đã giới thiệu ở mục ?? để xác định được vị trí vùng chữ trong các ảnh hóa đơn. Sau đó tiến hành cắt ảnh chữ vùng chữ dựa trên thông tin tọa độ như hình dưới đây:

Sau khi có các ảnh vùng chữ, chúng tôi tiến hành label khoảng 7000 tấm để phục vụ cho việc huấn luyện mô hình VietOCR. Có khoảng 5000 ảnh huấn luyện và 2000 ảnh kiểm thử.

Như đã đề cập ở mục [VietOCR](#), chúng ta có 2 loại mô hình OCR để giải quyết là Attention-OCR và Transformer-OCR. Chúng tôi huấn luyện cả 2 mô hình này và kết hợp với pretrained của tác giả để ra được những mô hình cuối cùng. Với model pretrained weight đã được huấn luyện với 10 triệu ảnh từ tác giả. Chúng tôi tiếp tục train trên bộ dữ liệu chúng tôi tự tạo thêm 20000 iters lần lượt với cả Attention-OCR(seq2seq) và Transformer-OCR config. Thông số cụ



Hình 14: Minh họa data huấn luyện cho VietOCR

thể để train mô hình VietOCR như sau:

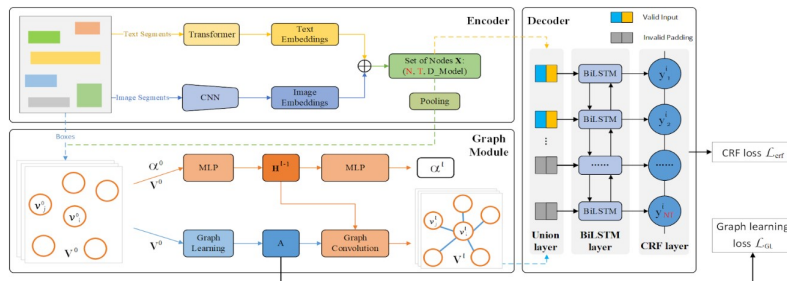
- 'print every':100,
- 'valid every':500,
- 'iters':20000,
- 'batch size': 16

3.4 Model PICK (KIE)

Sau khi nhận dạng được nội dung của các vùng chữ, nhóm sẽ thực hiện việc trích xuất các trường thông tin cần thiết sử dụng [6].

PICK là một framework hiệu quả và mạnh mẽ trong việc xử lý các bộ dữ liệu phức tạp để trích xuất thông tin (Key Information Extraction- KIE) bằng cách kết hợp Graph Learning với Graph Convolution, mang lại một biểu diễn ngữ nghĩa phong phú hơn chứa các tính năng văn bản, hình ảnh và bố cục chung một cách rõ ràng. Bên cạnh đó, việc PICK sử dụng rất nhiều features của văn bản bao gồm text, image và position features làm tăng khả năng biểu diễn của chúng. Việc biểu diễn các features từ rất nhiều thông tin làm cho PICK đạt hiệu năng tốt hơn các mô hình còn lại.

Hướng tiếp cận đơn giản nhất đó là sử dụng Text Classification để phân loại ra những thông tin nào thuộc lớp nào, cách giải quyết này có thể đơn giản và tốt trên những dạng văn bản có sự dạng thấp và phân biệt rõ rệt giữa các trường thông tin, và đặc biệt là cấu trúc văn bản đó đơn giản.



Hình 15: Trích xuất thông tin với model PICK

- **Encoder:** Sử dụng mô hình Transformer để trích xuất đặc trưng từ văn bản và sử dụng một mạng CNN để trích xuất đặc trưng từ ảnh. Sau đó kết hợp text embeddings và image embeddings lại thành vector biểu diễn X thể hiện khả năng biểu diễn text và image chứa text đó. X được đưa xem là đầu vào của Graph module.
Set of Note X: X là ma trận sau khi kết hợp 2 ma trận thu được từ Transformer vs CNN và được biểu diễn dưới dạng các Node của Graph và X được dùng làm input của Graph Module => Có thể nói toàn bộ quá trình nói trên đều thuộc hàm Encoder và các set Node chính là các bounding box thu được sau OCR.
- **Graph module:** Sử dụng một mạng GCN (Graph Convolutional Network) để làm giàu khả năng biểu diễn giữa các node với nhau. Việc các thông tin cần trích xuất có vị trí và nội dung khác nhau, nó không cục bộ và không theo thứ tự nên việc sử dụng Graph giúp mô hình có thể học được khả năng biểu diễn mối tương quan giữa chúng về khoảng cách và vị trí trong văn bản.
Graph Modeling: cách thức xây dựng graph dựa trên các bounding box text đã được OCR.
- **Decoder:** Sau khi kết hợp mô-đun Graph để làm giàu thông tin thì mô hình kết hợp thông tin đó và cả thông tin do mô-đun Encoder sinh ra để đưa vào mô-đun Decoder để nhận dạng và phân loại chúng. Ở mô-đun này mô hình PICK sử dụng BiLSTM layer + CRF layer.
Nhận đầu vào là sự kết hợp giữa đầu ra của Encoder và Graph Module. Cho qua một mạng BiLSTM layer và CRF layer để phân loại chúng. Cuối cùng mô hình sử dụng 2 hàm loss để tối ưu đồng thời chúng, đó là loss của Graph learning và CRF loss.

Mô hình sẽ nhận dữ liệu đầu vào là một ảnh chứa hóa đơn đã được recognize vùng chữ và kết quả đầu ra của mô hình được thể hiện như ở hình dưới đây.

4 Kết quả huấn luyện và đánh giá mô hình:

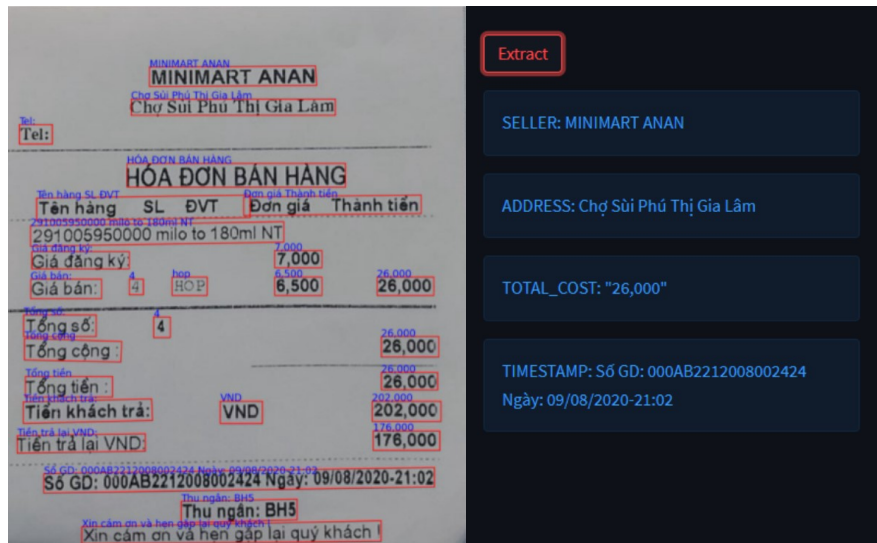
Sau khi huấn luyện qua 20.000 iterations trên GPU RTX 2080TI thì nhóm đã thu được kết quả như sau

Model	Acc full seq	Acc per char	GPU Time
vgg_transformer	0.7666	0.9228	20.90
vgg_seq2seq	0.7646	0.9201	9.56

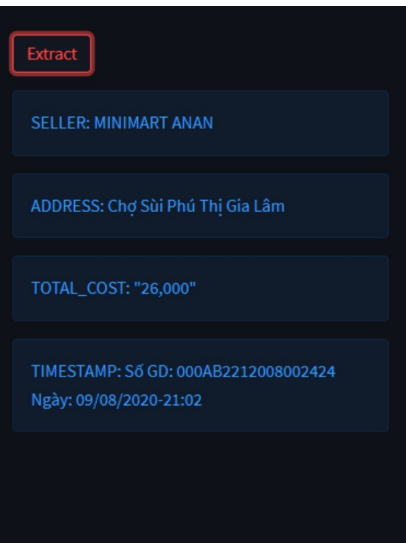
Kết quả huấn luyện cho ra được 4 trường thông tin cần trích xuất:

- **SELLER:** Tên của cửa hàng
- **ADDRESS:** địa chỉ của cửa hàng
- **TIMESTAMP:** thời gian xuất hoá đơn
- **TOTAL COST:** tổng chi phí có trong hoá đơn

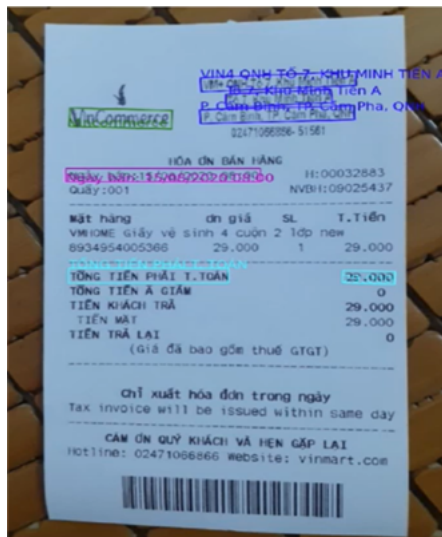
Đánh giá tốc độ chạy full pipeline của mô hình được thể hiện ở bảng dưới đây



Hình 16: Input



Hình 17: Output



Hình 18: Kết quả thông tin trích xuất

Stage	Avg processing/ image (s)	GPU Usage (MB VRAM)
Receipt Detection	0.981	803
Text Detection (PANet)	0.504	1099
Text Recognition (VietOCR)	0.960	1029
Key Information Extraction (PICK)	0.705	1891
Full pipeline	3.15	4822

Bảng 1: Kết quả đánh giá về tổ độ xử lý và dung lượng VRAM cho toàn bộ pipeline

5 Thiết kế phần mềm:

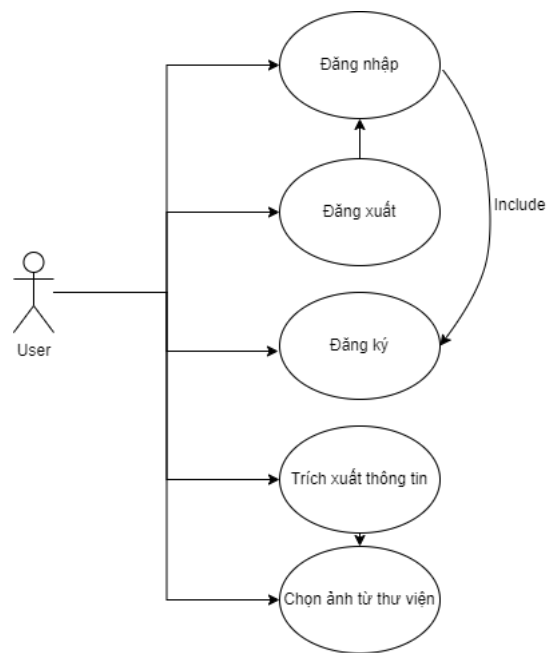
Yêu cầu của người dùng cần chúng ta làm một web-app có thể upload hình ảnh hóa đơn lên web và khách hàng nhận được thông tin được trích xuất ra từ hình ảnh hóa đơn mà người dùng đã upload. Khách hàng có thể xem lại các ảnh đã được trích xuất. Từ những yêu cầu của khách hàng, chúng ta sẽ làm đặc tả yêu cầu như sau:

- **Chức năng đăng nhập:** Giúp người dùng có thể đăng nhập vào web. Chức năng này dành cho người dùng.
- **Chức năng đăng xuất:** Giúp người dùng có thể đăng xuất khỏi web. Chức năng này dành cho người dùng.
- **Chức năng đăng ký:** Giúp người dùng có thể đăng kí một tài khoản và thông tin của khách hàng sẽ được lưu vào database. Chức năng này dành cho người dùng.
- **Chức năng trích xuất thông tin từ hóa đơn:** Chức năng này gửi ảnh của người dùng upload lên website lên server thông qua API để lấy thông tin mà model đã predict ra được trả lại cho người dùng. Chức năng này dành cho người dùng.
- **Chức năng chọn ảnh từ thư viện ảnh:** Khi người dùng upload lên trên web thì chúng ta sẽ lưu ảnh và thông tin đã trích xuất, khi người dùng đăng nhập vào thì có thể thấy được những ảnh đã được upload lên web cũng như thông tin đã được trích xuất ra của những bức ảnh đó. Chức năng này dành cho người dùng.

Từ đặc tả yêu cầu, chúng ta cần sử dụng những công nghệ sau:

- **Database:** NoSQL giúp ta xây dựng database một cách đơn giản. NoSQL cung cấp một cơ chế để lưu trữ và truy xuất dữ liệu được mô hình hóa khác với các quan hệ bảng được sử dụng trong các cơ sở dữ liệu kiểu quan hệ.
- **Code:** Chúng ta xài Visual Studio Code vì nó là text editor nhẹ nhưng nhiều extension trong đó hỗ trợ rất tốt ngôn ngữ lập trình mà chúng ta đang xài đó là python.
- **Testing:** Phần testing thì tỷ lệ thành công của tất cả các chức năng là ổn trừ chức năng trích xuất thông tin có những test case model predict không ra nên sẽ báo lỗi. Chúng ta cần handle những lỗi này bằng việc sử dụng try catch.

Chức năng	Tỷ lệ thành công
Đăng nhập	100%
Đăng kí	100%
Đăng xuất	100%
Trích xuất thông tin từ hóa đơn	90%
Chọn ảnh từ thư viện ảnh	100%



Hình 19: Use case

6 Kết luận

Chúng ta đã xây dựng được hệ thống trích xuất thông tin hình ảnh hóa đơn và hoạt động khá là tốt. Tuy nhiên vẫn có khá nhiều hóa đơn vẫn chưa thể trích xuất được và trả về lỗi và về mặt UI vẫn chưa được hoàn thiện chỉnh chu và nhóm sẽ cố gắng phát triển thêm sau này.

Tài liệu tham khảo

- [1] CodaLab - Competition. URL: <https://aihub.vn/competitions/1>.
- [2] Anthony Kay. "Tesseract: An Open-Source Optical Character Recognition Engine". In: *Linux J*. 2007.159 (July 2007), p. 2. ISSN: 1075-3583.
- [3] Tsung-Yi Lin et al. *Focal Loss for Dense Object Detection*. 2017. DOI: [10.48550/ARXIV.1708.02002](https://doi.org/10.48550/ARXIV.1708.02002). URL: <https://arxiv.org/abs/1708.02002>.
- [4] PaddlePaddle/PaddleOCR. original-date: 2020-05-08T10:38:16Z. July 1, 2022. URL: <https://github.com/PaddlePaddle/PaddleOCR>.
- [5] VietOCR - Nhận Dạng Tiếng Việt Sử Dụng Mô Hình Transformer và AttentionOCR – Quoc Pham – Data Scientist at Overspace. URL: <https://pbcquoc.github.io/vietocr/> (visited on 07/01/2022).
- [6] Wenwen Yu. *PICK-PyTorch*. original-date: 2020-07-15T15:17:46Z. June 29, 2022. URL: <https://github.com/wenwenyu/PICK-pytorch> (visited on 07/01/2022).