ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
ĐẠI HỌC CÔNG NGHỆ THÔNG TIN


# BÁO CÁO ĐỒ ÁN

## MÔN HỌC: THỊ GIÁC MÁY TÍNH NÂNG CAO

## ĐỀ TÀI

# VIETNAMESE HANDWRITTEN RECOGNITION

**Giảng viên hướng dẫn:** Mai Tiến Dũng

**Sinh viên thực hiện:** Tô Thanh Hiền - 19521490
Trần Vĩ Hào - 19521482
Lê Đặng Đăng Huy - 19521612

**Lớp:** CS331.M22.KHCL

Thành phố Hồ Chí Minh, ngày 24 tháng 06 năm 2022

# MỤC LỤC

I. TÓM TẮT....2
II. GIỚI THIỆU CHUNG....2
III. CÁC NGHIÊN CỨU LIÊN QUAN....3
1. CNN....3
1.1. Convolutional Layer....3
1.2. Sub-sampling và lớp Pooling....4
1.3. Fully-connected Layer (FC Layer)....4
2. RNN....4
3. CTC....6
4. Transformers....7
5. Vision transformers....7
6. Data efficient image transformers....8
7. Bert....8
7.1. Pretraining BERT....9
a. Masked LM (MLM)....9
b. Next Sentence Prediction (NSP)....9
7.2. Fine-Tuning BERT....9
8. RoBerta....10
IV. PHƯƠNG PHÁP TIẾP CẬN....11
1. CRNN+CTC....11
1.1. Feature Sequence Extraction....12
1.2. Sequence Labeling....12
1.3. Transcription....13
2. VietOCR....13
3. TrOCR....14
V. THỰC NGHIỆM....15
1. Thang đo....15
2. Bộ dữ liệu....15
3. Kết quả thực nghiệm....16
4. Kết luận....16
VI. TÀI LIỆU THAM KHẢO....17

# I. TÓM TẮT

Nhận dạng ký tự quang học tuy là một trong những bài toán lâu đời của lĩnh vực thị giác máy tính nhưng vẫn chưa có phương pháp giải quyết triệt để bài toán. Bài toán OCR thông thường sẽ gồm 2 tác vụ **text detection** và **text recognition**. Ở đây nhóm chúng tôi tập trung vào nghiên cứu tác vụ nhỏ của OCR là text recognition. Các phương pháp hiện có cho text recognition được xây dựng dựa trên CNN để hiểu hình ảnh và RNN hay transformers để tạo văn bản cấp ký tự. Với kết quả tiên tiến nhất của các mô hình image transformers đã có nhóm chúng tôi dự định thay vì sử dụng CNN thì sẽ áp dụng các mô hình image transformer cho việc trích xuất đặc trưng của mô hình text recognition. Ở đây chúng tôi sẽ thử nghiệm các mô hình sử dụng cả transformer và CNN làm backbone để giải quyết Vietnamese handwritten recognition.

# II. GIỚI THIỆU CHUNG

Nhận dạng ký tự quang học là bài toán chuyển đổi các văn bản, ký tự xuất hiện trong hình ảnh, hay các tài liệu đã scan thành định dạng mà máy tính có thể hiểu được. Từ đó, ta có thể dễ dàng chỉnh sửa, tìm kiếm và thực hiện nhiều tác vụ khác. Ở Việt Nam, bài toán nhận dạng chữ viết Việt đã và đang là một bài toán hấp dẫn và đầy thách thức đối với các nhà nghiên cứu. Cụ thể, Viện nghiên cứu Trí tuệ Nhân tạo VinAI Reasearch của Tập đoàn Vingroup đã phối hợp với Sở Thông tin và Truyền thông Thành phố Hồ Chí Minh tổ chức Cuộc thi AI Challenge 2021 với chủ đề là "Nhận diện chữ tiếng Việt trong ảnh ngoại cảnh và sinh hoạt hằng ngày". Nhận dạng chữ viết Việt ngữ là một trong những bài toán thú vị và rất quan trọng. Nó có nhiều ứng dụng trong việc số hóa văn bản để lưu trữ, tìm kiếm trên các văn bản scan, hình ảnh. Với đầu vào là một ảnh chứa chuỗi các chữ và đầu ra là chuỗi các chữ nằm trong ảnh đó. Như chúng ta đã biết, dữ liệu hình ảnh là một trong những loại dữ liệu cực kỳ quan trọng bên cạnh dữ liệu văn bản và dữ liệu âm thanh. Bên trong hình ảnh có thể chứa nội dung văn bản và việc trích xuất được nội dung văn bản đó sẽ đem lại rất nhiều lợi ích cho chúng ta. Có thể kể tới một số ví dụ như:

- Tìm kiếm thông tin văn bản trong hình ảnh.
- Dịch văn bản qua một ngôn ngữ khác.
- Đọc thông tin giấy tờ tùy thân.
- Nhập hóa đơn tự động bằng cách chụp lại hóa đơn.

Những công dụng vừa kể trên đã thúc đẩy nhóm chúng tôi chọn đề tài "Vietnamese Handwritten Recognition". Tuy rằng đây là một bài toán khó và nhiều thách thức nhưng cộng đồng nghiên cứu thị giác thế giới nói chung và Việt Nam nói riêng cũng đã đạt được nhiều thành tựu trong việc giải quyết bài toán nên chúng tôi sẽ trình bày ba mô hình để giải quyết bài toán mà nhóm đã tìm hiểu được dựa trên các nghiên cứu có sẵn: (1) Convolutional recurrent neural network kết hợp với Connectionist temporal classifications loss; (2) VietOCR; (3) Transformer-based OCR.

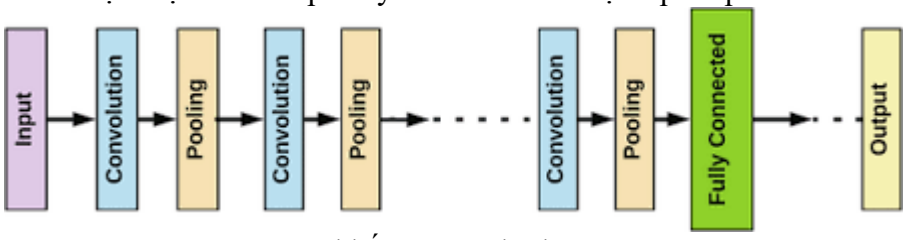
# III. CÁC NGHIÊN CỨU LIÊN QUAN

## 1. CNN

Convolutional Neural Network (CNN) là **"state-of-the-art"** cho các tác vụ phân loại hình ảnh (Image classification task). Convolutional Neural Network (CNN hay ConvNet) là một loại multi-layer neural network đặc biệt được lấy cảm hứng từ cơ chế của hệ thống quang học của các sinh vật sống.. Vào năm 1990, LeCun et al. giới thiệu mô hình CNN thực tế và phát triển LeNet-5. Huấn luyện bằng thuật toán backpropagation giúp LeNet-5 nhận diện các visual pattern trực tiếp từ pixel mà không sử dụng các cơ chế kỹ thuật cho việc tách các feature riêng biệt. Và ít hơn số connections cũng như số parameters của CNN so với feedforward neural networks thông thường với cùng kích cỡ, khiến cho xây dựng mô hình dễ hơn. Mặc dù có ưu thế hơn hẳn, hiệu suất của CNN trong các vấn đề phức tạp như phân loại hình ảnh độ phân giải cao, bị hạn chế do thiếu dữ liệu đào tạo lớn, thiếu phương pháp regularization tốt hơn và khả năng tính toán của máy tính lúc bấy giờ.

Ngày nay, chúng ta có những bộ dữ liệu lớn hơn với hàng triệu dữ liệu được gắn nhãn có độ phân giải lớn với hàng ngàn categories như ImageNet [10], LabelMe [11] etc. Với sự ra đời của các GPU mạnh mẽ và phương pháp regularization tốt hơn, CNN mang lại hiệu suất vượt trội trong các tác vụ phân loại hình ảnh (Image classification task). Vào năm 2012 một mạng học sâu convolution neural, AlexNet, được giới thiệu bởi Krizhevsky et al. đã thể hiện hiệu suất xuất sắc cho thử thách ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [13]. Thành công của AlexNet đã trở thành nguồn cảm hứng cho các mô hình CNN khác nhau như ZFNet [14], VGGNet [15], GoogleNet [16], ResNet [17], DenseNet [18], CapsNet [19], SENet [20] vào những năm kế tiếp.

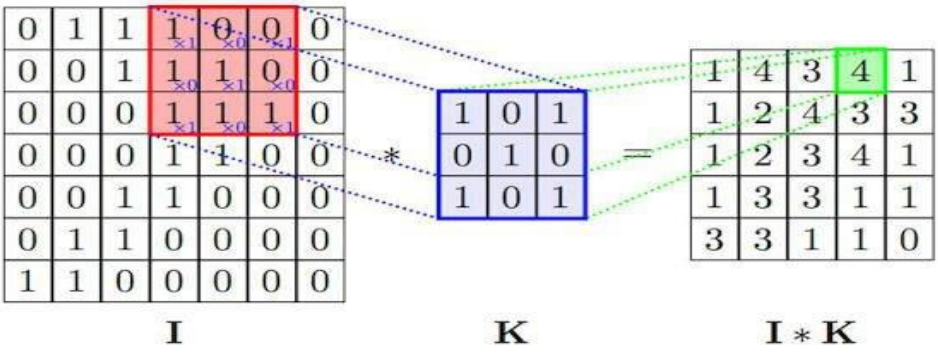
Một mạng CNN cơ bản bao gồm một hoặc nhiều khối convolution và lớp sub-sampling, sau đó một hoặc nhiều lớp fully connected và một lớp output như hình sau.




Các khối của mô hình CNN

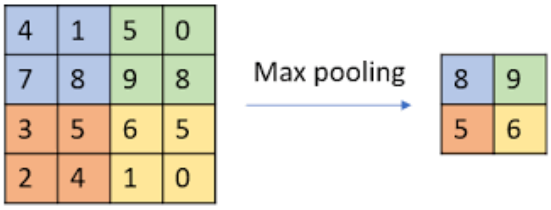
### 1.1. Convolutional Layer

Lớp convolution is là phần quan trọng của mô hình CNN. Ảnh nói chung là đứng yên trong tự nhiên. Điều đó có nghĩa là cấu tạo một phần của hình ảnh cũng tương đồng như bất kỳ phần nào khác. Vì vậy, feature học được ở một vùng có thể trùng với pattern tương tự ở vùng khác. Trong một hình ảnh lớn, chúng tôi lấy một phần nhỏ và đi qua tất cả các điểm trong hình ảnh lớn (Input). Trong khi đi qua tại bất kỳ điểm nào, chúng tôi convolve vào một vị trí duy nhất (Input). Mỗi phần nhỏ của hình ảnh đi qua hình ảnh lớn được gọi là Filter (Kernel). Filters sau đó được cấu tạo dựa vào kỹ thuật back propagation. Hình sau cho thấy convolutional operation thông thường.



### 1.2. Sub-sampling và lớp Pooling

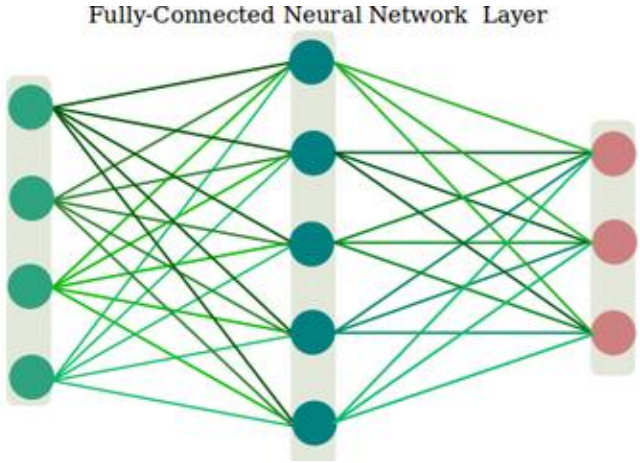
Pooling hiểu đơn giản là down-sampling bức ảnh. Nó lấy một vùng nhỏ của convolutional output làm Input và sub-samples nó để tạo ra single output duy nhất.. Một số kỹ thuật pooling như là max pooling, mean pooling, average pooling,... Max pooling lấy giá trị pixel lớn nhất của một vùng như hình dưới. Pooling làm giảm số lượng tham số phải tính nhưng làm cho mạng bất biến đối với các phép dịch về hình dạng, kích thước và tỷ lệ.




Max Pooling operation

### 1.3. Fully-connected Layer (FC Layer)

Phần cuối cùng của CNN cơ bản là lớp fully connected. Lớp này nhận Input từ tất cả các nơ-ron ở lớp trước và thực hiện operation với từng nơ-ron trong lớp hiện tại để tạo ra Output.

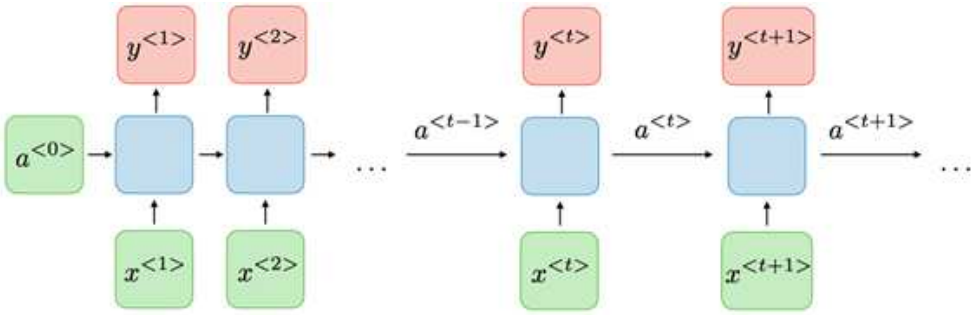



Fully-connected layer

## 2. RNN

Mạng Recurrent neural dựa trên công việc của David Rumelhart năm 1986. Hopfield networks – một loại mạng RNN đặc biệt được John Hopfield công bố lại vào năm 1982.

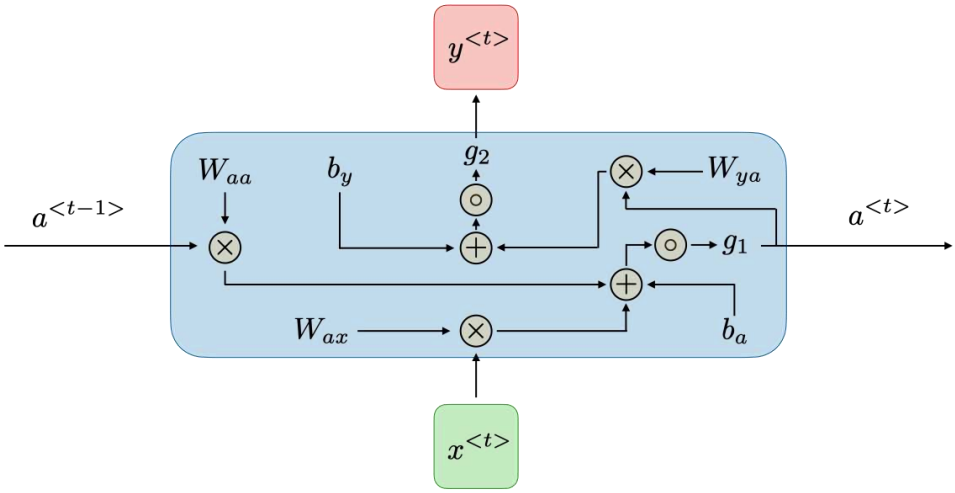
A recurrent neural network (RNN) là một loại đặc biệt của neural networkđược điều chỉnh để làm việc cho dữ liệu chuỗi thời gian hoặc dữ liệu liên quan đến chuỗi. Feed forward neural networks thông thường chỉ dành cho các điểm dữ liệu, các điểm này độc lập với nhau. Tuy nhiên, nếu chúng ta có dữ liệu theo một trình tự sao cho một điểm dữ liệu phụ thuộc vào điểm dữ liệu trước đó, chúng ta cần phải sửa đổi mạng nơ-ron để kết hợp các mối quan hệ phụ thuộc giữa các điểm dữ liệu này. Các RNN có khái niệm "bộ nhớ" giúp chúng lưu trữ các trạng thái hoặc thông tin của các input trước đó để tạo ra output tiếp theo của chuỗi.



Cho mỗi timestep $t$, activation $a^{<t>}$ và the output $y^{<t>}$ được thể hiện như sau:

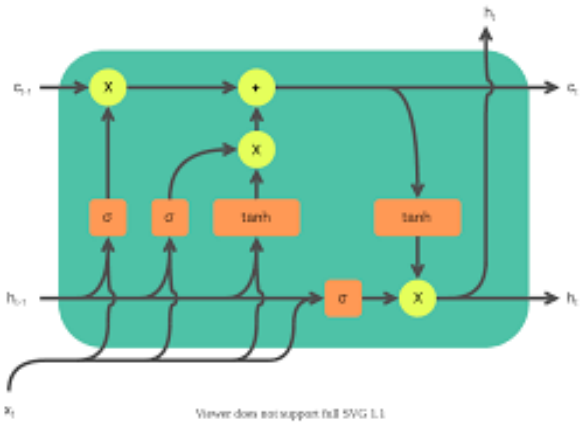
$$a^{<t>} = g_1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a) \quad \text{and} \quad y^{<t>} = g_2(W_{ya}a^{<t>} + b_y)$$

Trong đó $W_{ax}$, $W_{aa}$, $W_{ya}$, $b_a$, $b_y$ là hệ số đang được chia sẽ với nhau tạm thời và $g_1$, $g_2$, activation functions.

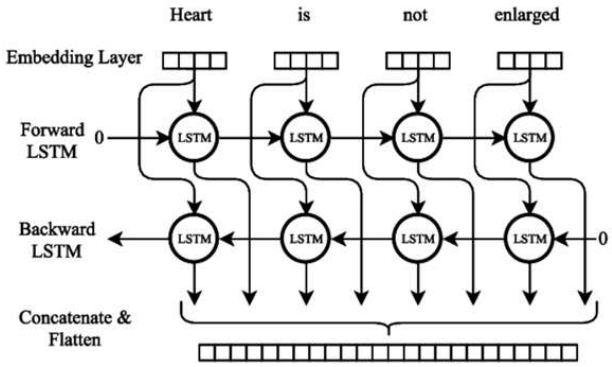



Tuy nhiên, các unit RNN thông thường gặp phải vấn đề về **Vanishing gradient**, điều này giới hạn phạm vi ngữ cảnh mà nó có thể lưu trữ và thêm gánh nặng cho quá trình đào tạo. Long-Short Term Memory (LSTM) là một loại RNN unit được thiết kế đặc biệt để giải quyết vấn đề này. Long short-term memory (LSTM) networks lần đầu tiên được giới thiệu bởi Hochreiter và Schmidhuber vào năm 1997 và thiết lập các kỷ lục về accuracy trong nhiều vùng ứng dụng.

LSTM bao gồm ô nhớ (memory cell) và 3 cổng nhân là input, output và cổng quên (forget gates). Về mặt khái niệm, ô nhớ lưu trữ các ngữ cảnh trong quá khứ và các cổng đầu vào và đầu ra cho phép ô lưu trữ các ngữ cảnh trong một khoảng thời gian dài. Trong khi đó, bộ nhớ trong ô có thể được xóa bằng cổng quên. Thiết kế đặc biệt của LSTM cho phép nó nắm bắt được các phụ thuộc trong phạm vi dài, thường xảy ra trong các chuỗi dựa trên hình ảnh.




LSTM chỉ là một chiều (directional), tức là nó chỉ dùng những nội dung quá khứ. Bidirectional LSTM (biLSTM), là một mô hình xử lý chuỗi mà bao gồm hai LSTMs: một lấy input theo hướng từ trái sang phải và đầu kia theo hướng ngược lại. Không giống như LSTM tiêu chuẩn, đầu vào đi theo cả hai hướng và nó có khả năng sử dụng thông tin từ cả hai phía. BiLSTMs tăng một cách hiệu quả lượng thông tin có sẵn cho mạng, cải thiện ngữ cảnh có sẵn cho thuật toán (ví dụ như biết những từ nào ngay sau đó và đứng trước một từ trong câu).




## 3. CTC

Nhiều tác vụ học trình tự trong thế giới thực yêu cầu dự đoán chuỗi nhãn từ dữ liệu đầu vào không được phân đoạn, nhiễu. Trong nhận dạng giọng nói, ví dụ một tín hiệu âm thanh được phiên âm thành các từ hoặc các đơn vị từ phụ. Recurrent neural networks (RNNs) là mạng có thể học trình tự chuỗi vô cùng mạnh mẽ có vẻ rất phù hợp với những công việc như vậy. Tuy nhiên, vì chúng yêu cầu dữ liệu đào tạo được phân đoạn trước và xử lý sau để chuyển output của chúng thành chuỗi nhãn, khả năng ứng dụng của chúng cho đến nay vẫn bị hạn chế.

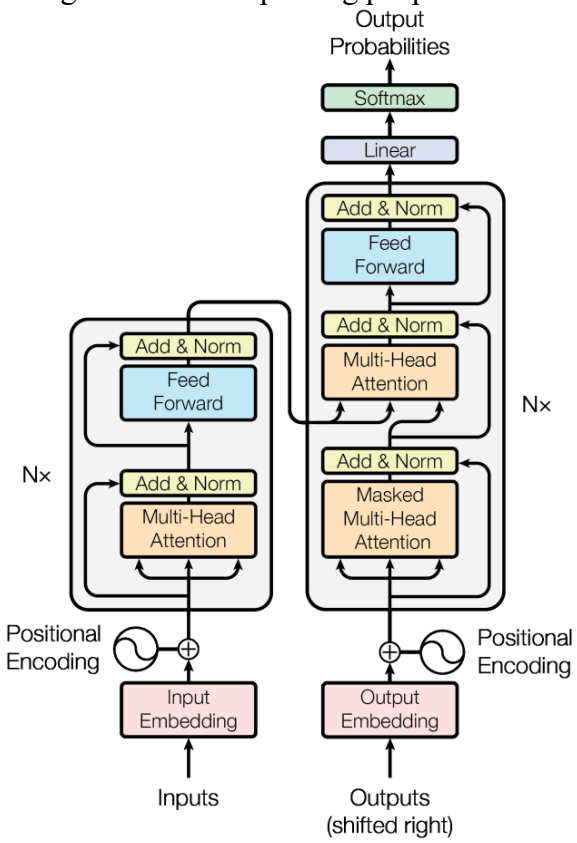
**Connectionist temporal classification** (**CTC**) là một loại neural network output và liên quan tới scoring function, để huấn luyện Recurrent neural networks (RNNs) như là mạng LSTM để giải quyết các vấn đề về trình tự chuỗi trong đó có chiều không gian về thời gian. Nó có thể được sử dụng cho các tác vụ như nhận dạng chữ viết tay trực tuyến hoặc nhận dạng dạng âm thanh giọng nói. CTC đề cập đến outputs, tính điểm và độc lập với cấu trúc mạng nơ-ron đang sử dụng.

Input là một chuỗi các quan sát và outputs là một chuỗi các nhãn, có thể bao gồm khoảng trắng. Khó khăn của việc đào tạo xuất phát từ việc có nhiều quan sát hơn là nhãn. Ví dụ trong speech audio có thể có nhiều lát thời gian tương ứng với một âm vị. Vì chúng ta không biết sự liên kết của chuỗi được quan sát với các nhãn mục tiêu, chúng tôi dự đoán phân phối xác suất ở mỗi timestep. Mạng CTC có continuous output (ví dụ như softmax), được trang bị thông qua đào tạo để mô hình hóa xác suất của nhãn. Các chuỗi nhãn được coi là tương đương nếu chúng chỉ khác nhau về sự liên kết, bỏ qua các khoảng trống. CTC không cố gắng tìm hiểu ranh giới và thời gian. Các chuỗi nhãn tương đương có thể xảy ra theo nhiều cách - điều này làm cho việc ghi điểm trở thành một nhiệm vụ không hề dễ, nhưng có thuật toán forward–backward hiệu quả cho việc đó.

CTC scores sau đó có thể được sử dụng với thuật toán lan truyền ngược (back-propagation) để cập nhật trọng số mạng nơ-ron.

## 4. Transformers

Được đề xuất bởi Vaswani et al. (2017) cho dịch máy và kể từ đó đã trở thành phương pháp tiên tiến nhất trong nhiều tác vụ NLP. Transformer là một kiến trúc mô hình tránh cơ chế **recurrence** và thay vào đó dựa hoàn toàn vào cơ chế **attention** để thu hút sự phụ thuộc toàn cục giữa Input và Output. Trước Transformers, các mô hình truyền thứ tự chiếm ưu thế (dominant sequence transduction) dựa trên các mạng recurrent phức tạp hoặc mạng nơ-ron convolutional bao gồm một bộ mã hóa và một bộ giải mã. Transformer cũng sử dụng bộ mã hóa và giải mã, nhưng việc loại bỏ đi cơ chế recurrence mà ưu tiên cơ chế attention cho phép song song hóa nhiều hơn đáng kể so với các phương pháp như RNN và CNN.




## 5. Vision transformers

Transformer tiêu chuẩn nhận như đầu vào một chuỗi token embeddings 1D. Để xử lý hình ảnh 2D, hình ảnh có kích thước x ∈ $R^{H\times W\times C}$, được reshape về thành một chuỗi flattened 2D patches $x_p$ ∈$R^{N\times(P^2.C)}$, trong đó (H, W) là độ phân giải của hình ảnh gốc, C là số kênh, (P, P) là độ phân giải của mỗi image patch và N = HW/$P^2$ là số lượng patches kết quả, cũng đóng vai trò là chuỗi đầu vào vào hiệu dụng cho Transformer. Transformer sử dụng latent vectơ không đổi có kích thước D thông qua tất cả các lớp của nó, vì vậy chúng tôi làm phẳng các patches và ánh xạ tới chiều D bằng phép chiếu tuyến tính có thể huấn luyện (trainable linear projection). Chúng tôi coi đầu ra của phép chiếu này là các patch embeddings.

Tương tự BERT's [class] token, ta thêm trước learnable embedding vào chuỗi embedded patches ($z_0^0 = x_{class}$),nơi có trạng thái tại output của Transformer encoder ($z_l^0$) đóng vai trò là đại diện hình ảnh $y$ .Cả hai pre-training và fine-tuning,một đầu phân loại được gắn vào $z_l^0$. Đầu phân loại được cho vào MLP với một lớp ẩn tại thời điểm pre-training và bởi single linear layer tại thời điểm fine-tuning. Position embeddings được thêm vào patch embeddings để giữ lại thông tin vị trí. Ta sử dụng các 1D position embeddings tiêu chuẩn có thể tự học. Chuỗi kết quả của các vectơ embedding vai trò là đầu vào cho bộ mã hóa (encoder). Bộ mã hóa Transformer (Transformer encoder) (Vaswani et al., 2017) bao gồm các lớp xen kẽ của các khối MLP và multiheaded self-attention. Layernorm (LN) được áp

dụng trước mỗi khối và residual connections sau mỗi khối (Wang et al., 2019; Baevski & Auli, 2019).

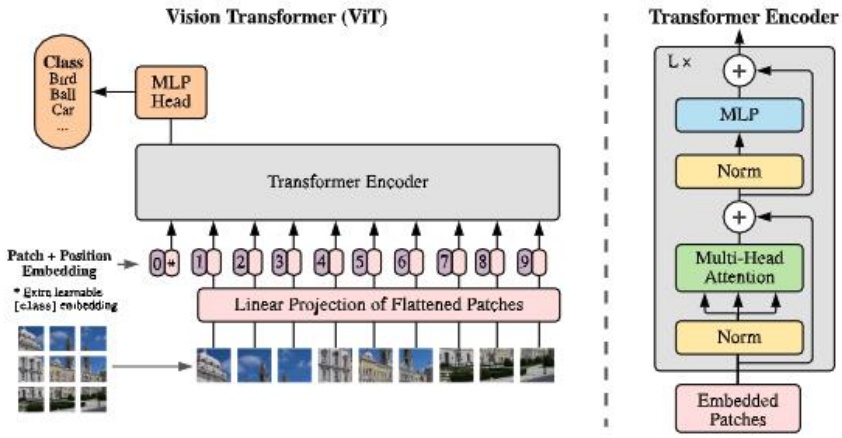
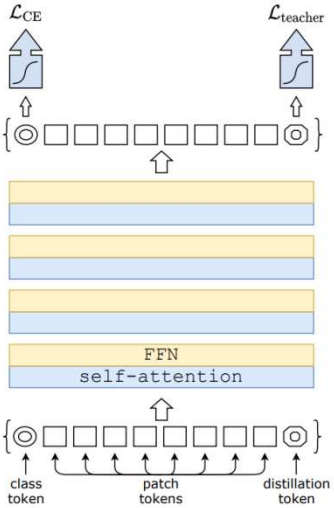



Figure 1: Model overview. We split an image into fixed-size patches, linearly embed each of them, add position embeddings, and feed the resulting sequence of vectors to a standard Transformer encoder. In order to perform classification, we use the standard approach of adding an extra learnable "classification token" to the sequence. The illustration of the Transformer encoder was inspired by Vaswani et al. (2017).

## 6. Data efficient image transformers

Vision transformer (ViT) đã trình bày kết quả xuất sắc với việc được đào tạo với một tập dữ liệu hình ảnh có nhãn nội bộ lớn (JFT-300M [46], 300 triệu hình ảnh). Bài báo kết luận rằng transformer *"không khái quát hóa tốt khi được đào tạo về lượng dữ liệu không đủ"*, và việc đào tạo các mô hình này yêu cầu đến các nguồn tài nguyên máy tính lớn.

Data-Efficient Image Transformer, DeiT, được đề xuất. Mặc dù kiến trúc phần lớn giống với ViT, nhưng nó được đào tạo trên ImageNet chỉ sử dụng một máy tính duy nhất trong vòng chưa đầy ba ngày, không có dữ liệu bên ngoài. Nó dựa trên distillation token đảm bảo rằng mô hình con học hỏi từ mô hình cha thông qua attention.




## 7. Bert

BERT, là viết tắt của Bidirectional Encoder Representations from Transformers. Việc thực hiện gần như giống với model gốc transformer. Không giống như các mô hình biểu diễn ngôn ngữ gần đây, BERT được thiết kế để pre-train các biểu diễn hai chiều sâu từ văn bản không được gắn nhãn bằng cách kết nối chung trên cả ngữ cảnh bên trái và bên phải trong tất

cả các lớp. Do đó, mô hình pre-trained BERT có thể được fine-tuned với một lớp đầu ra bổ sung để tạo ra các mô hình "state-of-the-art" cho một loạt các nhiệm vụ, chẳng hạn như trả lời câu hỏi (question answering) và suy luận ngôn ngữ (language inference) mà không cần phải sửa đổi kiến trúc tùy thuộc tác vụ. BERT đơn giản về mặt khái niệm và mạnh mẽ. Nó thu được kết quả tốt nhất trên mười một tác vụ xử lý ngôn ngữ tự nhiên, bao gồm việc đẩy điểm GLUE lên 80,5% (cải thiện tuyệt đối 7,7% điểm), MultiNLI accuracy lên 86,7% (cải thiện tuyệt đối 4,6%), SQuAD v1.1 trả lời câu hỏi Test F1 đến 93,2 (cải thiện tuyệt đối 1,5 điểm) và SQuAD v2.0 Kiểm tra F1-score đến 83,1 (cải thiện tuyệt đối 5,1 điểm). Có hai bước trong BERT framework: pre-training và fine-tuning.

Trong quá trình pre-training, mô hình được đào tạo về dữ liệu không được gắn nhãn qua các tác vụ pre-training trước khác nhau.

### 7.1. Pretraining BERT

***a. Masked LM (MLM)***

Một phần input tokens đầu vào được che (masked) một cách ngẫu nhiên, và sau đó masked tokens đó sẽ được dự đoán, 15% của các WordPiece tokens được che một cách ngẫu nhiên trong mỗi chuỗi.

Chỉ những từ bị che (masked words) được dự đoán thay vì tái tạo lại toàn bộ dữ liệu đầu vào.

Nếu token thứ *i* được chọn, token thứ *i* được thay thế bằng (1) [MASK] token 80% thời gian (2) token ngẫu nhiên 10% thời gian (3) token thứ *i* không thay đổi 10% thời gian. Sau đó, Ti sẽ được sử dụng để dự đoán token ban đầu với cross entropy loss.

***b. Next Sentence Prediction (NSP)***

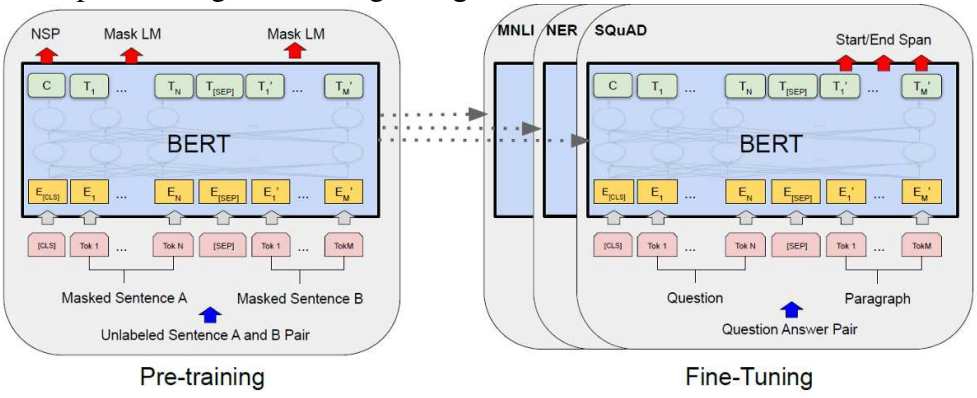
Downstream tasks như Question Answering (QA) và Natural Language Inference (NLI) dựa trên việc hiểu mối quan hệ giữa hai câu, điều này không trực tiếp nắm bắt được bằng mô hình ngôn ngữ. Task dự đoán câu tiếp theo được pretrained. Task dự đoán câu tiếp theo có thể được tạo ra từ bất kỳ ngữ liệu đơn ngữ nào.

Cụ thể, khi chọn các câu A và B cho mỗi ví dụ đào tạo trước, 50% thời gian B là câu thực sự tiếp theo A (được gắn nhãn là Next) và 50% thời gian đó là một câu ngẫu nhiên từ ngữ liệu (được gắn nhãn dưới dạng NotNext).

### 7.2. Fine-Tuning BERT

Ở input, câu A và câu B từ phần pre-training analogous to (1) sentence pairs in paraphrasing, (2) hypothesis-premise pairs in entailment, (3) question-passage pairs in question answering, and (4) a degenerate text-Ø pair in text classification or sequence tagging. Ở đầu ra, các biểu diễn token được đưa vào một lớp output cho các nhiệm vụ token-level, chẳng hạn như sequence tagging hoặc question answering và biểu diễn [CLS] được đưa vào lớp output để phân loại, chẳng hạn như entailment hoặc sentiment analysis.

So với pre-training, fine-tuning tương đối rẻ hơn.



## 8. RoBerta

RoBERTa là một phần mở rộng của BERT với những thay đổi đối với quy trình tiền đào tạo. Các sửa đổi bao gồm:

- Đào tạo mô hình lâu hơn, với batches lớn hơn, với nhiều dữ liệu hơn.
- Loại bỏ tác vụ next sentence prediction.
- Huấn luyện trên chuỗi dài hơn.
- Thay đổi masking pattern được áp dụng cho dữ liệu đào tạo. Các tác giả cũng thu thập một tập dữ liệu mới lớn (CC-News) có kích thước tương đương với các tập dữ liệu nội bộ không public khác, để kiểm soát tốt hơn các hiệu ứng kích thước tập huấn luyện.

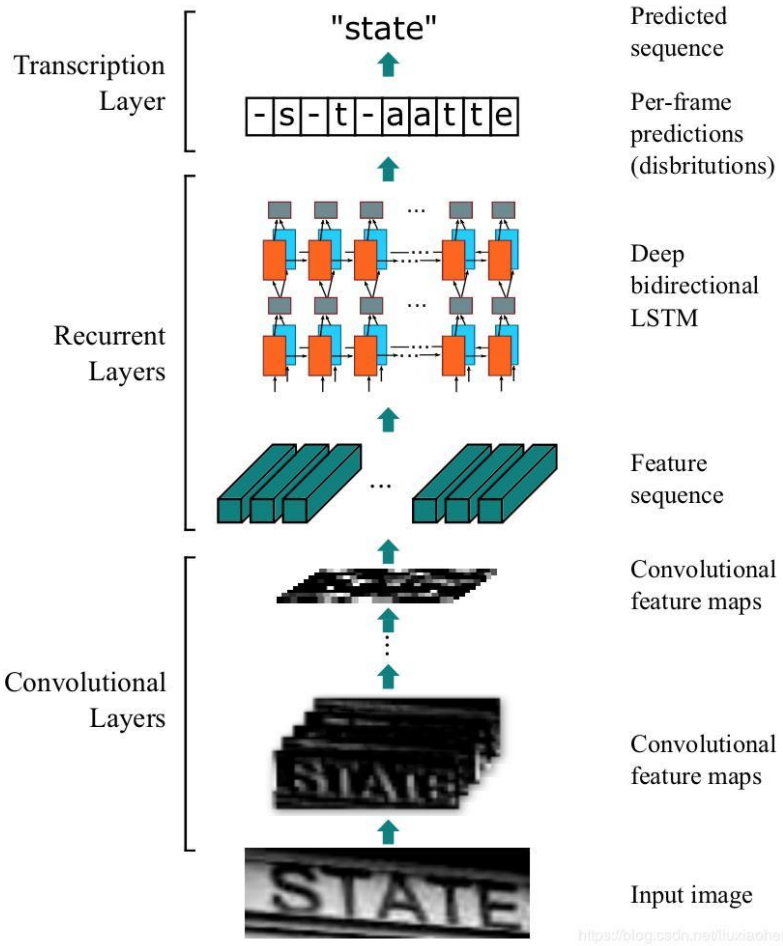
# IV. PHƯƠNG PHÁP TIẾP CẬN

Trong thế giới thực, chẳng hạn như văn bản cảnh, chữ viết tay và bản nhạc, có xu hướng xảy ra theo trình tự, chuỗi chứ không phải cô lập. Không giống như nhận dạng đối tượng chung, việc nhận dạng các đối tượng dạng chuỗi như vậy thường yêu cầu hệ thống dự đoán một loạt các nhãn đối tượng, thay vì một nhãn duy nhất. Do đó, việc nhận dạng các đối tượng như vậy có thể được coi là một bài toán nhận dạng trình tự. Một tính chất độc đáo khác của các đối tượng giống chuỗi là độ dài của chúng có thể thay đổi đáng kể. Ví dụ các từ tiếng Anh có thể bao gồm 2 ký tự như "OK" hoặc 15 ký tự như "congratulations". Do đó, các mô hình CNN sâu phổ biến nhất như VGG, Res-Net không thể được áp dụng trực tiếp vào dự đoán trình tự, vì mô hình thường hoạt động trên đầu vào và đầu ra có kích thước cố định, do đó không có khả năng tạo chuỗi nhãn có độ dài thay đổi.

Các mô hình mạng nơ-ron Recurrent (RNN), transformers, chủ yếu được thiết kế để xử lý các trình tự. Một trong những ưu điểm là nó không cần vị trí của từng phần tử trong ảnh đối tượng trình tự trong cả đào tạo và kiểm tra. Tuy nhiên, bước tiền xử lý để chuyển đổi hình ảnh đối tượng đầu vào thành một chuỗi các đặc trưng hình ảnh, thường là điều cần thiết.

Với sự ra đời của các mô hình Image transformer dựa trên mô hình transformer, ta có thể sử các mô hình để triết xuất đặc trưng hay mô hình CNN. Do sử dụng cơ chế attention so sánh global các phần ảnh với nhau thay vì sử dụng convolutional chỉ có thể so sánh các phần ảnh local ở low feature với nhau, mô hình Image transformer có inductive bias thấp hơn CNN. Như vậy các mô hình image transformer trực quan hóa dữ liệu tốt hơn, với điều kiện là dữ liệu đầu phải lớn.

## 1. CRNN+CTC



Kiến trúc mạng của CRNN bao gồm ba thành phần: các lớp convolutional, các lớp recurrent và một lớp phiên mã (transcription layer). Ở cuối CRNN, các lớp convolutional tự động trích xuất chuỗi đặc trưng từ mỗi hình ảnh đầu vào. Trên đầu của mạng convolutional, một mạng recurrent được xây dựng để đưa ra dự đoán cho từng khung của chuỗi đặc trưng, được xuất ra bởi các lớp convolutional. Lớp phiên mã ở đầu CRNN được sử dụng để dịch các dự đoán trên mỗi khung hình (per-frame) bởi lớp recurrent thành một chuỗi nhãn. Mặc dù CRNN bao gồm các loại kiến trúc mạng khác nhau (ví dụ CNN và RNN), nó có thể được huấn luyện chung với một loss function.

### 1.1. Feature Sequence Extraction

Trong mô hình CRNN, thành phần của các lớp convolutional được xây dựng bằng cách lấy các lớp convolutional và lớp max-pooling từ mô hình CNN chuẩn (các lớp fully-connected đủ sẽ bị loại bỏ). Thành phần như vậy được sử dụng để trích xuất một biểu diễn đặc trưng tuần tự từ một hình ảnh đầu vào. Trước khi được đưa vào mạng, tất cả các hình ảnh cần được scale đến cùng một chiều cao. Sau đó, một chuỗi các vectơ đặc trưng được trích xuất từ các feature map được tạo ra bởi thành phần của các lớp convolutional, là input cho các lớp recurrent. Cụ thể, mỗi vectơ đặc trưng của chuỗi đặc trưng được tạo từ trái sang phải trên feature maps theo cột. Điều này có nghĩa là vectơ đặc trưng thứ *i* là sự ghép nối của các cột thứ *i* của tất cả các maps. Chiều rộng của mỗi cột trong cài đặt của chúng tôi được cố định là một pixel duy nhất. Khi các lớp convolution, max-pooling và elementwise activation function kích hoạt theo từng phần tử hoạt động trên các vùng cục bộ (local region), chúng là bất biến dịch (translation invariant). Do đó, mỗi cột của feature maps tương ứng với một vùng hình chữ nhật của bức ảnh gốc (được gọi là receptive field) và các vùng hình chữ nhật đó có cùng thứ tự với các cột tương ứng của chúng trên feature maps từ trái sang phải.

### 1.2. Sequence Labeling

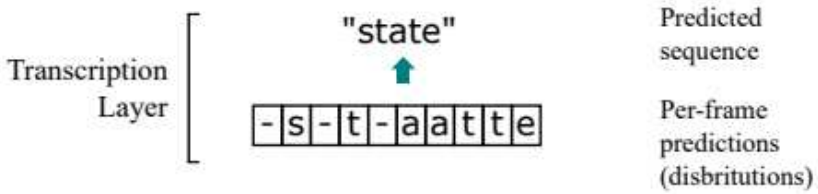
Mạng thần kinh bidirectional Recurrent sâu được xây dựng trên đầu các lớp convolutional, như các lớp recurrent. Các lớp recurrent dự đoán phân phối nhãn $y_t$ cho mỗi khung $x_t$ trong chuỗi đặc trưng x = $x_1$, ..., $x_t$. Có ba ưu điểm cho việc sử dụng các lớp recurrent. **Thứ nhất**, RNN có một khả năng mạnh mẽ trong việc nắm bắt thông tin theo ngữ cảnh trong một chuỗi. Sử dụng các dấu hiệu theo ngữ cảnh để nhận dạng chuỗi dựa trên hình ảnh ổn định và hữu ích hơn so với việc xử lý từng ký hiệu một cách độc lập. Lấy nhận dạng văn bản cảnh làm ví dụ, các ký tự rộng có thể yêu cầu nhiều hơn một khung liên tiếp để mô tả đầy đủ. Bên cạnh đó, một số ký tự không rõ ràng sẽ dễ phân biệt hơn khi quan sát ngữ cảnh của chúng, ví dụ dễ dàng nhận ra *"il"* bằng cách đối chiếu chiều cao của ký tự hơn là nhận ra từng ký tự một cách riêng biệt. **Thứ hai**, RNN có thể back-propagates sai lệch lỗi tới input của nó, tức là lớp convolutional, cho phép chúng ta huấn luyện chung các lớp recurrent và các lớp convolutional trong một mạng thống nhất. **Thứ ba**, RNN có thể hoạt động trên các chuỗi có độ dài tùy ý đi từ đầu đến cuối.

Tuy nhiên, RNN unit truyền thống gặp phải vấn đề vanishing gradient, hạn chế phạm vi ngữ cảnh mà nó có thể lưu trữ và thêm gánh nặng cho quá trình đào tạo. Long-Short Term Memory (LSTM) là một loại RNN unit được thiết kế riêng biệt để giải quyết vấn đề này. Thiết kế đặc biệt của LSTM cho phép nó nắm bắt các phụ thuộc tầm xa, thường xảy ra trong các chuỗi dựa trên hình ảnh.

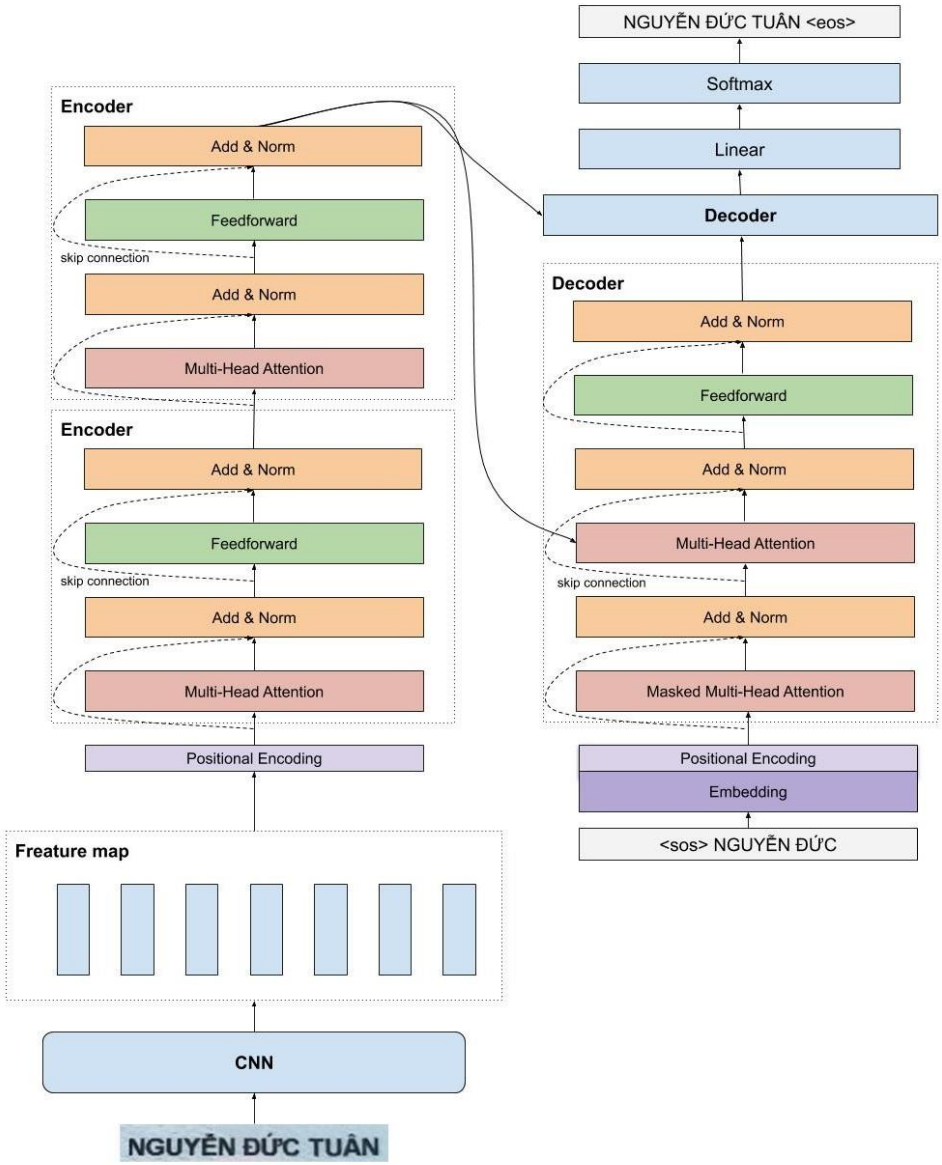
LSTM là một chiều (directional), nó chỉ sử dụng các ngữ cảnh trong quá khứ. Tuy nhiên, trong chuỗi dựa trên hình ảnh, bối cảnh từ cả hai hướng đều hữu ích và bổ sung cho nhau. Do đó, ta kết hợp hai LSTM, một tiến và một lùi, thành một LSTM hai chiều (bidirectional). Hơn nữa, nhiều LSTM hai chiều có thể được xếp chồng lên nhau, dẫn đến một LSTM hai chiều sâu.

### 1.3. Transcription

Phiên mã là quá trình chuyển đổi các dự đoán trên mỗi khung hình do RNN thực hiện thành một chuỗi nhãn. Về mặt toán học, phiên mã là để tìm chuỗi nhãn có xác suất cao nhất dựa trên các dự đoán trên mỗi khung hình.




## 2. VietOCR




VietOCR là một thư viện do ***Quoc Pham*** viết ra, sử dụng CNN và Transformer để nhận diện văn bản tiếng Việt được huấn luyện trên 10 triệu ảnh. Mô hình thể hiện kết quả ấn trượng đối với cả text và word level.

Về kiến trúc, VietOCR bao gồm 2 phần chính: một mô hình trích xuất đặc trưng của ảnh và một mô hình ngôn ngữ. Cụ thể, mô hình trích xuất đặc trưng ảnh có nét tương tự với

mô hình trích xuất đặc trưng của CRNN với việc sử dụng mô hình chuẩn VGG19 (đã được tinh chỉnh) và mô hình ngôn ngữ là Transformer.

Đối với mô hình trích xuất đặc trưng ảnh VGG19, tác giả đã thay đổi stride size (kích thước trượt) của các lớp pooling cuối là WxH = 2x1, vì ảnh thường có kích thước chiều dài lớn hơn chiều rộng, không thay đổi stride size phù hợp sẽ dẫn đến kết quả nhận diện tệ.

## 3. TrOCR

TrOCR được xây dựng với kiến trúc Transformer, bao gồm image Transformer để trích xuất các đặc trưng trực quan và text Transformer cho mô hình ngôn ngữ. Cấu trúc bộ mã hóa-giải mã (encoder-decoder) Transformer thông thường được áp dụng trong TrOCR. Bộ mã hóa được thiết kế để có được biểu diễn của image patches và bộ giải mã sẽ tạo ra chuỗi chữ trong khi chú ý đến đầu ra (output) của bộ mã hóa và output trước của decoder.

**Encoder**

Mô hình DeiT (Touvron et al., 2021a) được sử dụng để khởi tạo bộ mã hóa trong mô hình TrOCR.

**Decoder**

Mô hình RoBERTa được sử dụng cho quá trình khởi tạo bộ giải mã. Khi tải các mô hình RoBERTa vào bộ giải mã, các cấu trúc không khớp chính xác. Ví dụ lớp encoder-decoder attention không có trong các mô hình RoBERTa. Để giải quyết vấn đề này, bộ giải mã được khởi tạo bằng mô hình RoBERTa và lớp mà bị thiếu được khởi tạo ngẫu nhiên.

# V. THỰC NGHIỆM

## 1. Thang đo

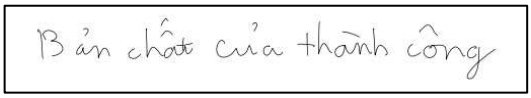
Ở đồ án lần này, chúng tôi sử dụng character error rate (CER) và word error rate (WER). Character error rate (CER) word error rate (wer) là thước đo phổ biến về hiệu suất cho hệ thống nhận dạng giọng nói tự động (automatic speech recognition). Cả hai đều bắt nguồn từ khoảng cách Levenshtein. Character error rate (CER) hoạt động trên ký tự trong khi word error rate (wer) hoạt động trên từ.

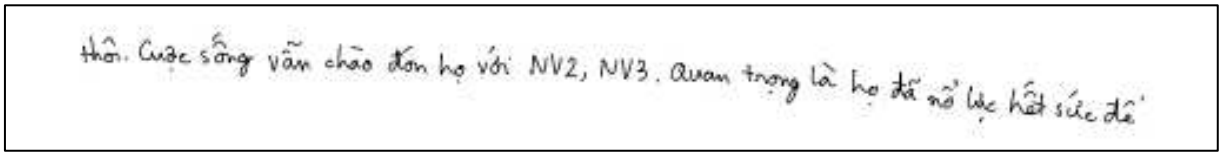
$$CER = \frac{S + D + I}{N}$$

$$WER = \frac{S_w + D_w + I_w}{N_w}$$

## 2. Bộ dữ liệu

Nhóm sử dụng bộ dữ liệu lấy từ ICFHR2018 Competition on Vietnamese Online Handwritten Text Recognition Database bao gồm 7282 bức ảnh. Label của ảnh thuộc bộ dữ liệu gồm 160 ký tự đặc biệt. Tất cả label đều độ lớn dưới 128 ký tự.



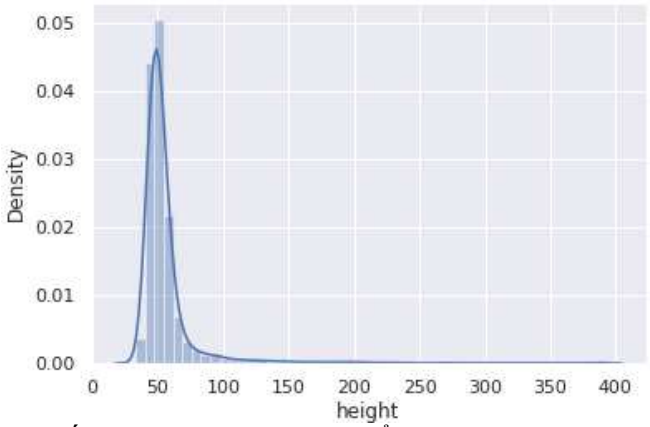




Ví dụ ảnh từ tập dataset

**Width**

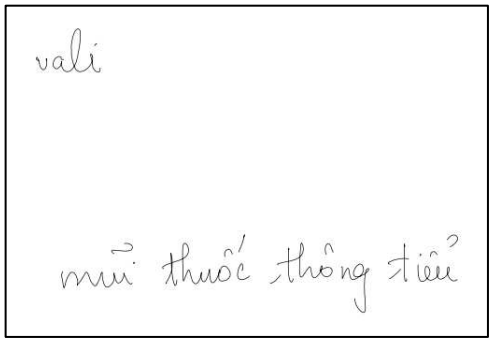
99% ảnh của dataset có width là 515px

**Height**




Dataset có một số ảnh thuộc word-level lẫn vào:

Dataset có một số ảnh không crop vào giữa ảnh:




Đối với những ảnh thuộc word-level sẽ bị loại. Còn ảnh không crop đúng ngay ảnh sẽ được crop lại.

## 3. Kết quả thực nghiệm

| | CER | WER |
|---|---|---|
| **VietOCR** | 0.119 | 0.294 |
| **TrOCR** | 0.092 | 0.213 |

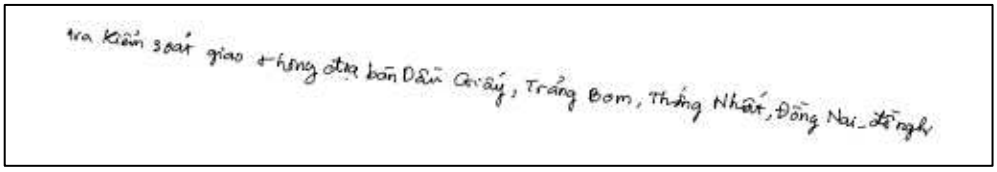
Từ bảng ta có thể thấy TrOCR vượt trội hơn VietOCR. Tuy nhiên TrOCR thời gian huấn và evaluate chậm hơn nhiều Vietocr.

Đáng tiếc mô hình CRNN không thể sử dụng cho dataset vì không thể học được gì cho quá trình huấn luyện. Tất cả các ảnh đều dự đoán như nhau.

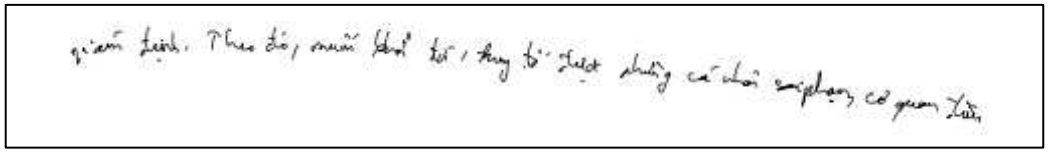
```
prediction[309]

't  àn tàn tàn tàn.'
```

Một phần có thể khác với cái mô hình trước sử dụng các mô hình pretrained. Mô hình này được huấn luyện từ đầu. Một nguyên nhân khác có thể do ảnh có quá nhiều chữ dẫn tới việc các mô hình RNN không hiệu quả như đã được biết RNN không xử lý tốt với các Long tem memory.

Một số trường hợp mô hình nhận diện không tốt:



Chữ bị nghiêng trong ảnh



Chữ bị cong trong ảnh

## 4. Kết luận

Chúng ta sử dụng 3 phương pháp để xử lý bài toán nhận diện chữ viết tay Việt Nam. Mô hình CRNN không thể dự đoán gì tuy nhiên VietOCR và TrOCR cho kết quả khá khả quan. Với TrOCR nhỉnh hơn chút tuy nhiên thời gian huấn luyện với evaluate lại lâu hơn.

# VI. TÀI LIỆU THAM KHẢO

[1]. LeCun, Y.; Boser, B.; Denker, J. S.; Henderson, D.; Howard, R. E.; Hubbard, W.; Jacker, L. D. (June 1990). "Handwritten digit recognition with a back-propagation network.

[2]. Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton "ImageNet Classification with Deep Convolutional Neural Networks".

[3]. Rumelhart, David E; Hinton, Geoffrey E, and Williams, Ronald J (Sept. 1985). Learning internal representations by error propagation.

[4]. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin Attention Is All You Need.

[5]. Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale.

[6]. Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, Hervé Jégou Training data-efficient image transformers & distillation through attention.

[7]. Baoguang Shi, Xiang Bai, Cong Yao An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition.

[8]. Minghao Li, Tengchao Lv, Lei Cui, Yijuan Lu, Dinei Florencio, Cha Zhang, Zhoujun Li, Furu Wei TrOCR: Transformer-based Optical Character Recognition with Pre-trained Models.

[9]. https://stanford.edu/~shervine/l/vi/teaching/cs-230/cheatsheet-recurrent-neural-networks

[10]. Farhana Sultana, A. Sufian, Paramartha Dutta "Advancements in Image Classification using Convolutional Neural Network".

[11]. Hung Tuan Nguyen, Cuong Tuan Nguyen, Pham The Bao, Masaki Nakagawa "A database of unconstrained Vietnamese online handwriting and recognition experiments by recurrent neural networks".

[12]. https://github.com/pbcquoc/vietocr.