# LỜI MỞ ĐẦU

Học sâu là một nhánh lớn của ngành học máy dựa trên tập hợp các thuật toán để cố gắng mô hình dữ liệu trừu tượng hóa ở mức cao, bằng cách sử dụng nhiều tầng xử lý phức tạp, hoặc bao gồm nhiều biến đổi phi tuyến. Trong những năm gần đây, nó đã phát triển một cách rất nhanh chóng và được ứng dụng trong nhiều lĩnh vực của công nghệ thông tin để giải quyết các vấn đề thực tế trong cuộc sống. Với mỗi vấn đề đặc trưng cho mỗi bài toán, các nhà nghiên cứu đã tìm ra các mô hình mạng nơ-ron sâu (Deep Neural Networks), mạng nơ-ron tích chập (CNN), mạng nơ-ron hồi quy (RNN), … nhằm xử lý các bài toán trong lĩnh vực như thị giác máy tính, xử lý ngôn ngữ tự nhiên, dịch máy, nhận dạng âm thanh, xử lý tiếng nói, sinh trắc học, …

Ngày nay những thành tựu trong lĩnh vực thị giác máy tính cũng như xử lý ngôn ngữ tự nhiên đã cho phép chúng ta có thể xử lý các vấn đề phức tạp như phát hiện và nhận dạng câu, chữ trong hình ảnh tự nhiên. Thành tựu đó là bước đầu cho việc xử lý và hỗ trợ cho con người có thể hiểu được những ngôn ngữ phức tạp khó có thể viết ra ví dụ kiểu chữ tượng hình trong một số ngôn ngữ như chữ Hán, chữ Kanji (tiếng Nhật).

Việc tò mò về nội dung của các câu chữ Hán Nôm được khắc trên các cột, cổng, … trong đền chùa, các khu di tích lịch sử đã thôi thúc tôi tìm hiểu và xây dựng ứng dụng nhận dạng và dịch nghĩa chúng. Tôi quyết định đưa việc giải quyết bài toán này làm đề tài đồ án tốt nghiệp.

# CHƯƠNG 1: ĐẶT VẤN ĐỀ

* 1. Giới thiệu bài toán

Khi vào bất cứ đền, chùa nào ta đều thấy những bức hoành phi, câu đối bằng chữ Hán, một số gia đình cũng sắm hoành phi câu đối làm đẹp và sang cho ngôi nhà của mình. Nhưng đa số khách và chủ nhà vẫn chưa hiểu hết được các bức hoành phi, câu đối ấy có ý nghĩa gì, bởi chúng đều được viết bằng chữ Hán. Hoành phi là một tấm biển gỗ có hình thứ trình bày theo chiều ngang, treo trên cao bên ngoài các gian thờ tại đình chùa, từ đường, … Nội dung của hoành phi thường thể hiện lòng tôn kính với tổ tông, bề trên, thần thánh,… Hoành phi được treo ở những nơi tôn nghiêm vừa mang tính chất nghi lễ lại vừa có giá trị nghệ thuật. Câu đối thuộc thể loại văn biền ngẫu gồm hai vế đối nhau nhằm biểu thị một ý chí, quan điểm tình cảm của tác giả trước một sự việc, một hiện tượng nào đó trong đời sống xã hội. Câu đối có nguồn gốc từ Trung Quốc và được xem là tinh hóa của chữ Hán, người Trung Quốc quan niệm: “Nếu thơ văn là tinh hoa của chữ nghĩa thì câu đối là tinh hoa của tinh hoa”. Không những thế, câu đối còn thể hiện trí thông minh, sự nhanh nhạy và phong cách tài tử trong việc đối đáp ứng xử. Câu đối tuy không dài, không nhiều chữ như những bài văn bài thơ, chỉ có hai vế nhưng vẫn thể hiện được ý tưởng, quan điểm một cách rõ ràng, cô đọng và súc tích. Mỗi câu đối được coi là một tác phẩm nghệ thuật, qui mô của tác phẩm tuy nhỏ, nhưng sức chứa, sức gợi rất lớn. Ngôn từ của câu đối được cân nhắc chọn lọc, âm điệu nhịp nhàng, kết cấu chặt chẽ, nhiều câu tài tình đến mức người đọc người nghe cảm thấy kỳ thú một cách bất ngờ khi nó bật ra ý tưởng mới lạ. Do ảnh hưởng của văn hóa Trung Quốc câu đối cũng là một trong những thể loại văn học của Việt Nam và một số nước châu Á.

Đến nay, chúng ta vẫn còn thấy những câu đối , hoành phi thường được treo ở những ngôi nhà cổ, đền, chùa, miếu, nhà thờ họ, cổng làng, đình làng. Hiện nay đa số người được giao trách nhiệm trông coi các đền, chùa, khu di tích… không biết chữ Hán, nên khi giải thích cho du khách ý nghĩa các hoành phi, câu đối đều lúng túng, hoặc giải thích không chuẩn xác. Vậy những hoành phi câu đối được thể hiện rất công phu ấy nhằm mục đích gì? Phải chăng chỉ làm sang, để thỏa mãn ý tưởng, nhận thức của những nhà xây dựng? Đôi khi những nhà hảo tâm hiến tặng cũng không nghĩ đến việc dịch câu đối đó cho người đọc dễ hiểu, nên chưa phục vụ đa số quần chúng nhân dân, chưa giúp cho người dân có thể hiểu, cảm, đồng điệu và thấm được cái hồn trong mỗi câu đối đó.

Từ những lý do đó tôi quyết định tìm hiểu và xây dựng một ứng dụng giúp mọi người có thể hiểu được nội dung ý nghĩa của các câu đối, hoành phi giúp cho mọi người thỏa mãn tính tò mò, nhu cầu tìm hiểu văn hóa của cha ông ta ngày xưa. Để thực hiện điều đó, đồ án của tôi cần được nghiên cứu giải quyết 3 bài toán chính là : Phát hiện phần nội dung văn bản trong ảnh (Text Detection ), Nhận dạng phần nội dung văn bản (Text Regcognition) và dịch sang Tiếng Việt (Translation).

* 1. Mục đích và bố cục của đồ án

Mục đích của đồ án:

* Xây dựng ứng dụng di động có tính năng hỗ trợ người dùng chụp ảnh, dịch nghĩa các câu từ tiếng Hán sang tiếng Việt.

Bố cục của đồ án được chia thành 3 phần:

* Phần 1: Tìm hiểu các cách tiếp cận, hướng giải quyết cho bài toán Text Detection, Text Regconition và Translation. Tìm hiểu các ý tưởng chính của các công trình nghiên cứu đã có, lý thuyết thuật toán, độ chính xác, ưu nhược điểm của mỗi phương pháp.
* Phần 2: Đánh giá, lựa chọn phương pháp phù hợp sau đó đưa vào xây dựng ứng dụng thử nghiệm. Đánh giá tổng quan việc sử dụng các phương pháp sau đó quyết định chọn phương pháp phù hợp với bài toán, ngữ cảnh hiện tại. Nêu cụ thể quá trình code, các bước tiến hành, khó khăn gặp phải, thành tựu đạt được trong quá trình thử nghiệm và xây dựng ứng dụng. Chia ra làm 2 phần: mô hình Deep Learning và xây dựng ứng dụng Android.
* Phần 3: Kết luận, đưa ra hướng phát triển.Nêu ra những thành quả đạt được, khó khăn gặp phải, cách giải quyết vấn đề đó, Những phần chưa làm được, định hướng phát triển ứng dụng trong tương lai.

# **CHƯƠNG 2: CÁC CÁCH TIẾP CẬN ĐỂ GIẢI QUYẾT BÀI TOÁN**

* 1. **Bài toán Text Detection**

Việc phát hiện nội dung văn bản trong hình ảnh tự nhiên là một thách thức trong lĩnh vực thị giác máy tính (computer vision). Bên cạnh đó, việc phát hiện văn bản là chữ tượng hình cũng gặp nhiều khó khăn hơn so với việc phát hiện văn bản chữ Latinh. Do đặc thù của bài toán đang cần giải quyết với chữ Hán nên tôi sử dụng bộ dữ liệu MSRA-TD500 vì các lý do: thứ nhất đây là bộ dữ liệu tiếng Hán, thứ 2 bộ dữ liệu chủ yếu là ảnh được chụp thực tế, tự nhiên nó giúp bao phủ được các vấn đề thường gặp khi phát hiện văn bản trong hình ảnh tự nhiên.

|  |  |
| --- | --- |
|  |  |
|  |  |

*Hình 1: Một số ảnh trong bộ MSRA-TD500*

Sau đây là các phương pháp tiếp cận để giải quyết bài toán text detection mà tôi đã tìm hiểu được, trong đó 3 phương pháp đầu tiên giải quyết bài toán theo hướng toán học cổ điển 4 phương pháp tiếp theo sử dụng các mô hình mạng nơ-ron. Kết quả đánh giá mỗi phương pháp được so sánh với nhau trên bộ dữ liệu MSRA-TD500.

* + 1. *Detecting Texts of Arbitrary Orientations in Natural Images.*

Phương pháp tiếp cận:

Phương pháp này sử dụng *Stroke Width Transform* (SWT) [] kết hợp với các phương pháp khác để tăng độ chính xác khi giải quyết bài toán Text Detection trong hình ảnh tự nhiên. Hầu hết các phương pháp tiếp cận cho bài toán này là tập trung phát hiện vị trí biên của các vùng text . Phương pháp này đưa ra hướng giải quyết vấn đề phát hiện nội dung trong ảnh tự nhiên nhưng chưa giải quyết được các vấn đề thường gặp như kích thức văn bản khác nhau, font chữ, góc nghiêng, …

Các bước tiến hành:

* Bước 1: Component Extraction

Sử dụng Edge detection để tạo ra Edge Map làm đầu vào cho module SWT tạo ra SWT image. Những pixel lân cận trong SWT image được nhóm lại với nhau để tạo thành các *Components*.

* Bước 2: Component Analysis

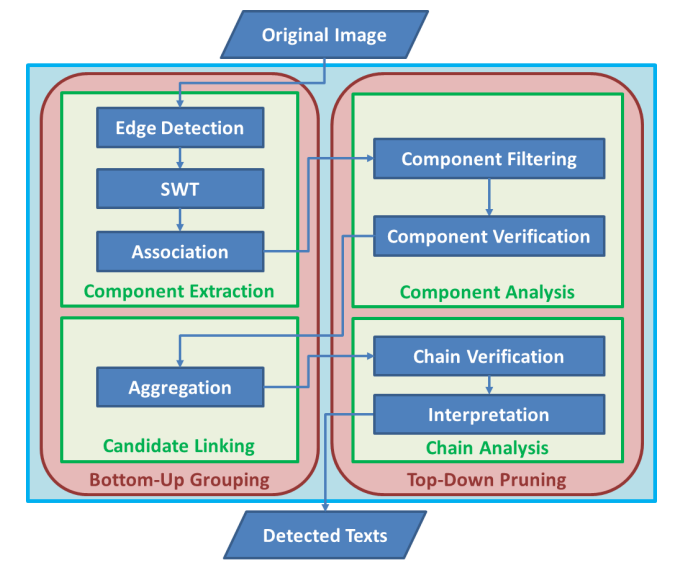
Có nhiều *Components* được lấy ra ở bước trên không thuộc bất kì vùng chứa text nào, cho nên ở bước này sẽ sử dụng bộ phân lớp đã được huấn luyệnđể định nghĩa và loại bỏ các *Components* không phải là văn bản.

* Bước 3: Candidates Linking

Các *Components* được giữ lại được coi như là các C*haracter Candidates*. Bước đầu tiên của Candidate Linking sẽ liên kết các *Character Candidates* thành từng cặp cho nếu hai *Candidates* liền kề nhau giống nhau về hình dạng và màu sắc. Các cặp *Candidates* được tổng hợp lại thành các chuỗi (*Chains*) theo kiểu đệ quy.

* Bước 4: Chains Analysis

Phân tích các chuỗi để xác định chính xác *Chains* có thuộc vùng văn bản hay không.



*Hình 2: Minh họa các bước phương pháp Detecting Texts of Arbitrary Orientations in Natural Images*

* + 1. *Scene Text Detection via Holistic, Multi-Channel Prediction.*

Điểm nổi bật của phương pháp:

* Có khả năng phát hiện được nhiều vùng văn bản, kể cả văn bản với các hướng khác nhau trong một ảnh tự nhiên.
* Sử dụng phương pháp tiếp cận tương tự như *Semantic Segmentation* với 2 nhãn (text và non-text).
* Thuật toán được đánh giá tốt hơn đáng kể so với các thuật toán trước đây.

Phương pháp tiếp cận:

* Bài báo giải quyết vấn đề phát hiện văn bản tưởng tự như giải quyết vấn đề của bài toán phân vùng ngữ nghĩa (*Semantic Segmentation*) khác biệt ở đây là chỉ có hai class là text và non-text (phương pháp được dựa trên nghiên cứu của Xie & Tu "Holistically-Nested Edge Detection" [15]), dự đoán dựa trên việc phân tích tổng quan toàn bộ bức ảnh cách này trái ngược với những cách tiếp cận trước đó là đưa ra dự đoán mang tính cục bộ tức là không sử dụng đầy đủ tất cả các thông tin trong toàn bộ bức ảnh.
* Phương pháp này đồng thời dự đoán được xác suất của vùng text, kí tự và mối quan hệ giữa các kí tự lân cận. Đặc biệt, nó có khả năng phát hiện được nhiều vùng của văn bản, các văn bản có hướng khác nhau.

Bảng so sánh kết quả so với các phương pháp khác:

|  |  |  |  |
| --- | --- | --- | --- |
| Algorithm | Precision | Recall | F-measure |
| **Proposed** | 0.7651 | **0.7531** | **0.7591** |
| Zhang et al. [5] | **0.83** | 0.67 | 0.74 |
| Kang et al. [7] | 0.71 | 0.62 | 0.66 |
| Yin et al. [3] | 0.71 | 0.61 | 0.66 |
| TD-Mixture [1] | 0.63 | 0.63 | 0.60 |
| TD-ICDAR [1] | 0.53 | 0.52 | 0.50 |
| Epshtein et al. [8] | 0.25 | 0.25 | 0.25 |
|  |  |  |  |

*Bảng 1: Scene Text Detection via Holistic, Multi-Channel Prediction trên bộ dữ liệu MSRA-TD500*

* + 1. *Orientation Robust Text Line Detection in Natural Scene Images.*

Đặc điểm nổi bật của phương pháp:

* Phát hiện đối tượng văn bản mà không phụ thuộc nhiều vào *font* chữ và ngôn ngữ trong hình ảnh tự nhiên.
* Sử dụng phương pháp: Maximally Stable Extremal Regions (MSER) mục đích chính của phương pháp này là phát hiện các đặc điểm nổi bật của đối tượng trong ảnh như là độ tương phản và màu sắc so với các vùng lân cận.

Phương pháp tiếp cận:

* Bước 1: MSER extraction
* Bước 2: Local text line hypotheses
* Bước 3: Pairwise edges in HOCC
* Bước 4: Results for HOCC
* Bước 5: Results for texture classification

Bảng so sánh kết quả so với các phương pháp khác:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Pricision | Recall | F-measure |
| **Our method** | **0.71** | 0.62 | **0.66** |
| Yao et al. [1] | 0.63 | **0.63** | 0.60 |
| Epshtein et al. [8] | 0.25 | 0.25 | 0.25 |
|  |  |  |  |

*Bảng 2: Orientation Robust Text Line Detection in Natural Scene Images trên bộ dữ liệu MSRA-TD500*

* + 1. *Detecting Oriented Text in Natural Images by Linking Segments.*

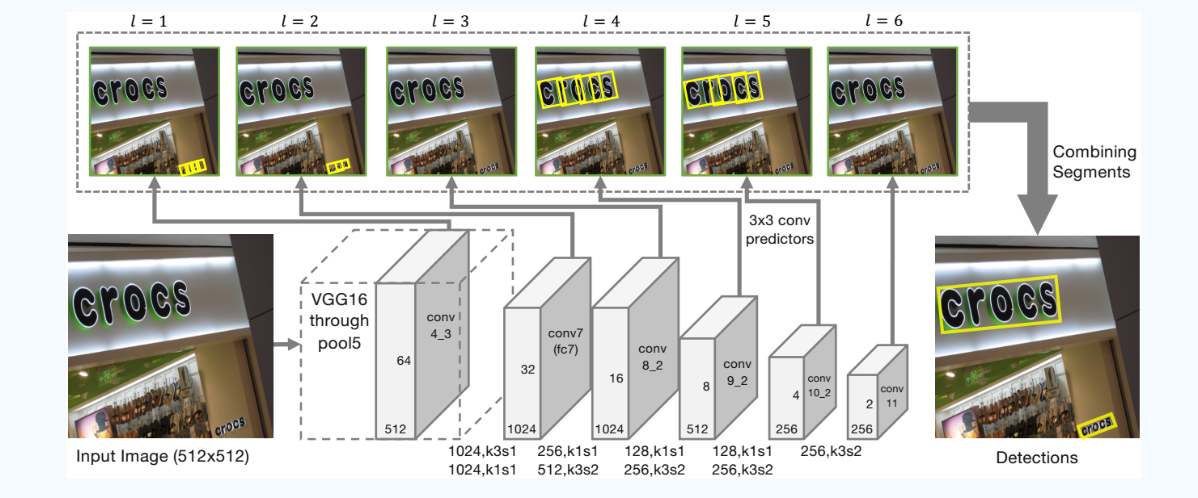
Đặc điểm nổi bật của phương pháp: phương pháp đã giải quyết tốt vấn đề phát hiện văn bản trong hình ảnh tự nhiên, nó giải quyết tốt hai vấn đề quan trọng mà các phương pháp trước đây chưa làm được đó là:

* Phát hiện được các dòng chữ dài kể cả không phải các chữ cái Latinh ví dụ là chữ Trung Quốc.
* Tốc độ xử lý nhanh, phù hợp với các ứng dụng realtime.

Khái niệm cơ bản:

* Segment: là một hộp giới hạn bao gồm một phần tử được định nghĩa bởi 5 thông số (x, y, w, h, θ).
* Link: một liên kết dùng để kết nối các segment liền kề, cho biết các segment thuộc cùng một dòng văn bản.
* Word: là một vùng bao phủ toàn bộ văn bản, có thể chứa một hoặc nhiều segment.

Các bước tiến hành:

*Hình 3: Minh họa mô hình mạng sử dụng cho Detecting Oriented Text in Natural Images by Linking Segments*

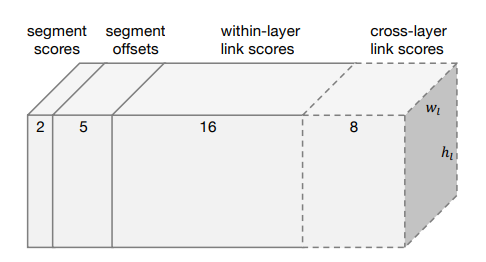
* Bước 1:

Các *Segment* (hộp màu vàng) và các link (không được vẽ ở đây) được phát hiện bởi các lớp convolution (đánh số từ 1 đến 6) mỗi convolution đại diện cho việc phát hiện segment với các kích thước khác nhau. Sau đó nó được kết hợp với nhau bằng phương pháp combining segments (ở bước 2) tạo thành các word.

Ở đây, sự khác biệt của mạng so với mạng SSD gốc là :

* + Thông tin về góc được thêm vào, thay vì 4 giá trị như mạng SSD thì ở đây có 5 giá trị (x, y, w, h, θ).
  + Chỉ sử dụng duy nhất một kích thước của defaut box.
  + Thay đổi lớp pooling cuối cùng của mạng SSD trở thành conv11 để detect links.

Output của bước 1: ứng với mỗi conv



*Hình 4:*

* Segment score có 2 giá trị là text hoặc non-text.
* Segment offset có 5 giá trị là (x, y, w, h, θ).
* 2 tầng còn lại là **within-layer link score** và **cross-layer link score** chứa giá trị thể hiện sự liên kết giữa các segment.
* Bảng so sánh kết quả so với các phương pháp khác:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Method | Precision | Recall | F-measure | FPS |
| Kang et al. [7] | 0.71 | 0.62 | 0.66 | - |
| Yao et al. [1] | 0.63 | 0.63 | 0.60 | 0.14 |
| Yin et al. [9] | 0.81 | 0.63 | 0.71 | 0.71 |
| Yin et al. [3] | 0.71 | 0.61 | 0.66 | 1.25 |
| Zhang et al. [5] | 0.83 | 0.67 | 0.74 | 0.48 |
| Yao et al. [2] | 0.77 | **0.75** | 0.76 | ~1.61 |
| **SegLink** | **0.86** | 0.70 | **0.77** | **8.9** |
|  |  |  |  |  |

* Segment score có 2 giá trị là text hoặc non-text.
* Segment offset có 5 giá trị là (x, y, w, h, θ).
* 2 tầng còn lại là **within-layer link score** và **cross-layer link score** chứa giá trị thể hiện sự liên kết giữa các segment.
  + 1. *Multi-Oriented Text Detection with Fully Convolutional Networks.*

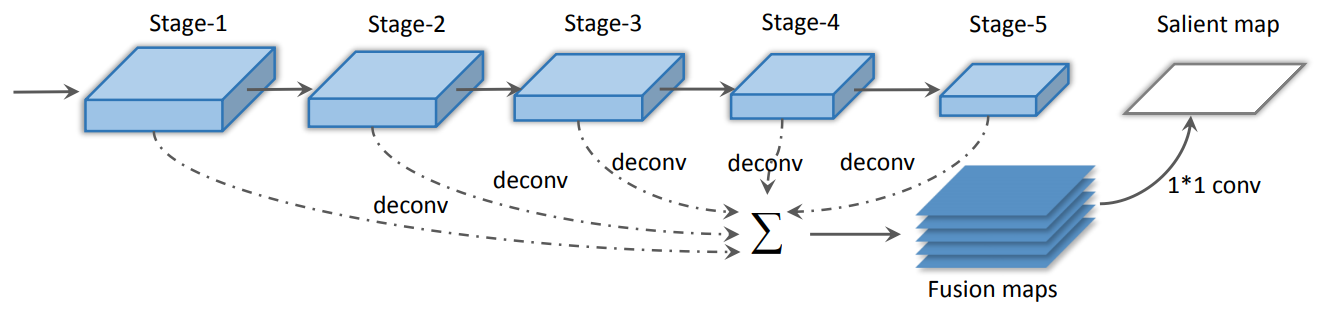
Ý tưởng chính của phương pháp:

* Đầu tiên một mạng neuron (FCN) được train để dự đoán vùng text trong tổng thể của bức ảnh. Sau đó là sử dụng phương pháp khác để phân loại (yes or no) các kí tự trong vùng được dự đoán ở trên.
* Phương pháp được đánh giá là có hiệu quả khi sử dụng phát hiện với các ngôn ngữ, font chữ và các hướng văn bản khác nhau.

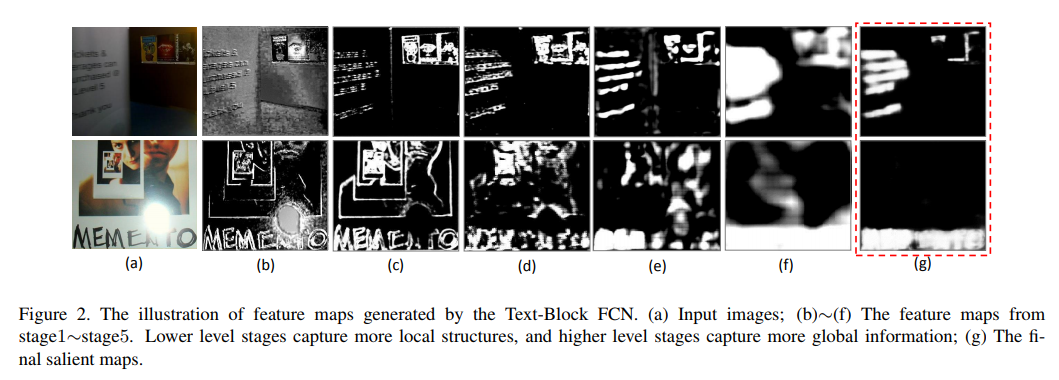
Các bước thực hiện: gồm 3 bước:

Bước 1: Text block detection:

* Text-Block FCN Network (convert the VGG 16)



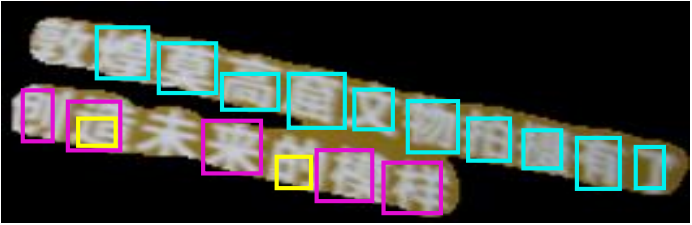
Mô tả phương pháp:



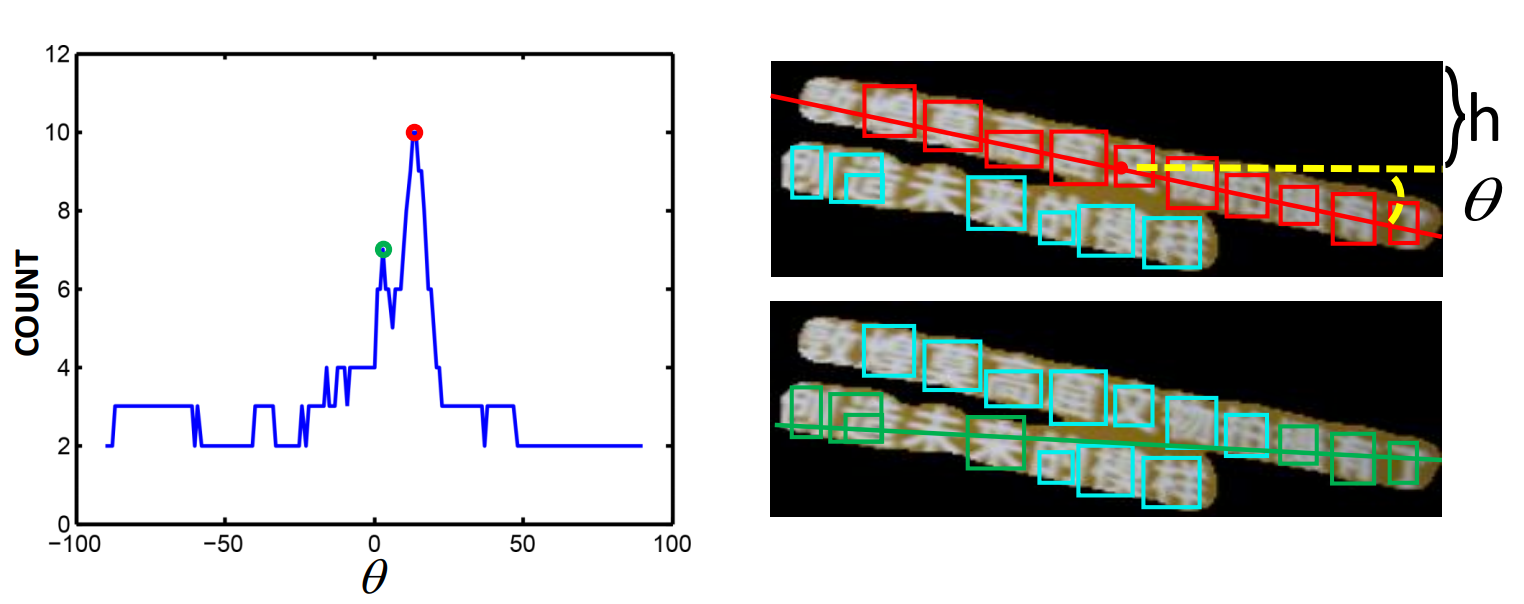
Các vùng text được phát hiện ở bước 1 được gọi là các **Text Block.**

Bước 2: Text Line Candidate Generation:

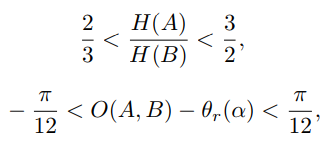
* Trích xuất kí tự: bước này sử dụng phương pháp MSER để trích xuất ra các kí tự từ **Text Box.**



* Phát hiện hướng văn bản:
  + Vẽ một đường thẳng đi qua các hộp kí tự được phát hiện ở trên
  + Hướng của văn bản được xác định bởi một đường thẳng đi qua nhiều hộp kí tự nhất.
  + Công thức xác định hướng của văn bản:
  + Mô tả phương pháp:

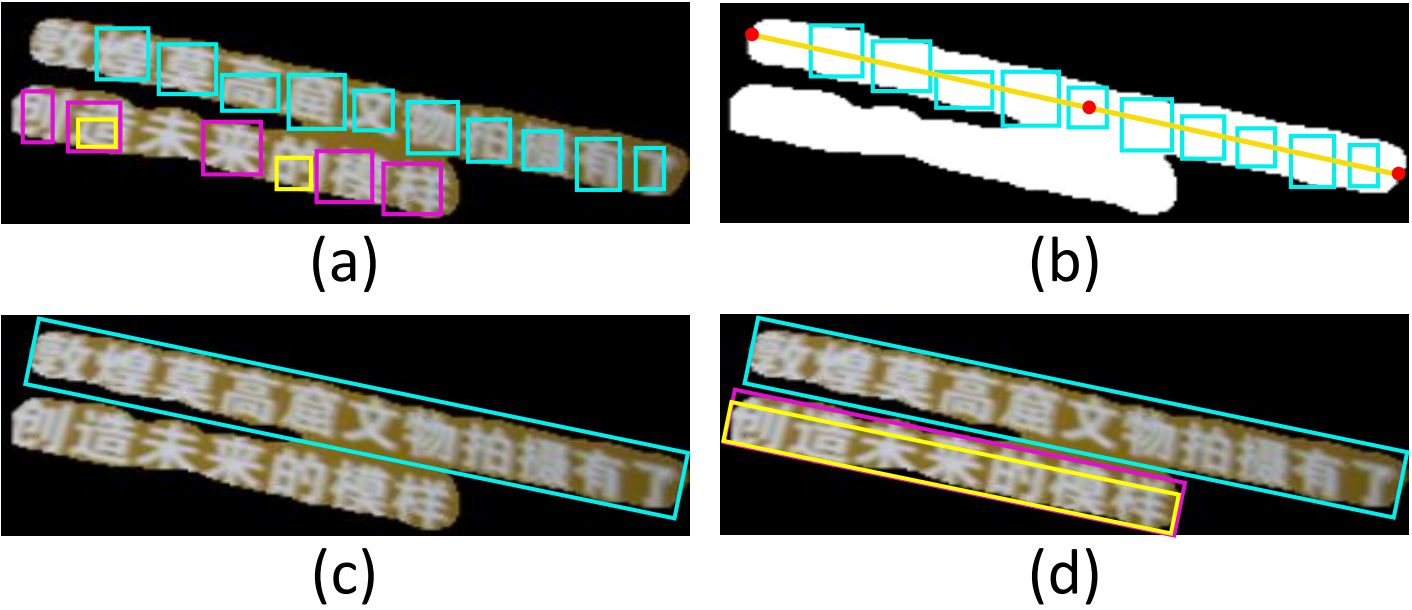


* Tạo ra **Text Line:**
  + Đầu tiên: các kí tự được nhóm lại với nhau bằng điều kiện sau:



Trong đó:

* + - H(A), H(B) là chiều cao của hộp A, B.
    - O(A, B) là góc của đường thẳng nối tâm A, B với phương ngang.
    - θr là góc của text line được ước tính ở trên.
  + Mỗi nhóm sẽ tạo ra được 1 Bounding Box (c) nó được coi là các **Text Line candidates** (d).



Bước 3: Text Line Candidates Classification:

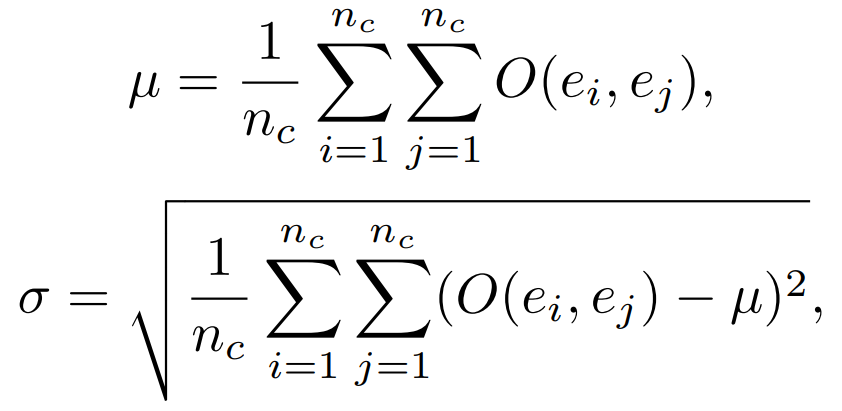
* Mục đích của bước này là loại bỏ các TextLine candidates sai.
* Để làm điều đó paper sử dụng model “Character-Centroid FCN”
* Mô tả Input-Output của model:
  + Input: **Text Line Candidates**
  + Output: The probability maps of character centroids



* Tiêu chí đánh giá output:

Với mỗi output bao gồm: ɛ = {(ei, si)}

* + Tiêu chí *Intensity Criterion:* là false Text Line Candidate nếu thoản mãn 1 trong 2 điều kiện
    - Số lương *Character Centroids < 2*
    - Giá trị trung bình xác suất < 0.6 (trong đó savg = )
  + Tiêu chí *Geometric Criterion:* sự xuất hiện của các centroids thường xấp xỉ là một đường thẳng



Trong đó:

* + - O(ei, ej) là góc giữa 2 centroids i và j.

Các TextLine candidates được cho là true nếu thỏa mãn:

Bảng so sánh kết quả so với các phương pháp khác:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Algorithm | Precision | Recall | F-Measure | FPS |
| **Proposed** | **0.83** | **0.67** | **0.74** | 0.48 |
| Yin et al. [9] | 0.81 | 0.63 | 0.71 | 0.71 |
| Kang et al. [7] | 0.71 | 0.62 | 0.66 | - |
| Yin et al. [3] | 0.71 | 0.61 | 0.66 | **1.25** |
| Yao et al. [1] | 0.63 | 0.63 | 0.60 | 0.14 |

* + 1. *Multi-Oriented Scene Text Detection via Corner Localization and Region Segmentation*

Điểm nổi bật của phương pháp:

* Khác với những phương pháp trước đây, phương pháp này kết hợp giữa object detection và sematic segmentation.
* Phương pháp dựa trên 2 cơ sở lý thuyết:
  + Một hình chữ nhật có thể xác định được bằng các đỉnh.
  + Phân vùng ngữ nghĩa có hiệu quả trong việc xác định vùng chứa text.

Phương pháp tiếp cận: Mạng neuron của phương pháp là một mạng fully convolutional đóng vai trò: trích rút đặc trưng (feature extraction), phát hiện góc (corner detection), và xác định vị trí vùng text (position-sensitive segmentation).

* Feature extraction: phần khung của mô hình là pre-trained của model VGG16.
* Corner detection: Dựa theo lý thuyết sử dụng SSD và DSSD phương pháp phát hiện góc bằng defaut boxes.
* Position-sensitive segmentation: Lấy ý tưởng từ phương pháp InstanceFNC, sử dụng position-sensitive segmentation để tạo ra segmentation maps.

Bảng so sánh kết quả so với các phương pháp khác:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Method | Precision | Recall | F-measure | FPS |
| TD-Mixture [1] | 0.63 | 0.63 | 0.60 | - |
| TD-ICDAR [1] | 0.53 | 0.52 | 0.50 | - |
| Kang et al. [7] | 0.71 | 0.62 | 0.66 | - |
| Zhang et al. [5] | 0.83 | 0.67 | 0.74 | 0.48 |
| Yao et al. [2] | 0.77 | 0.75 | 0.76 | ~1.61 |
| EAST [10] | 0.82 | 0.62 | 0.7 | 6.52 |
| EAST+ [10] | 0.81 | 0.67 | 0.76 | **13.2** |
| SegLink [4] | 0.86 | 0.70 | 0.77 | 8.9 |
| **ours** | **0.87** | **0.76** | **0.82** | 5.7 |

*Ghi chú: + là viết tắt của mạng nơ-ron không bao gồm VGG16*

* + 1. *EAST: An Efficient and Accurate Scene Text Detector*

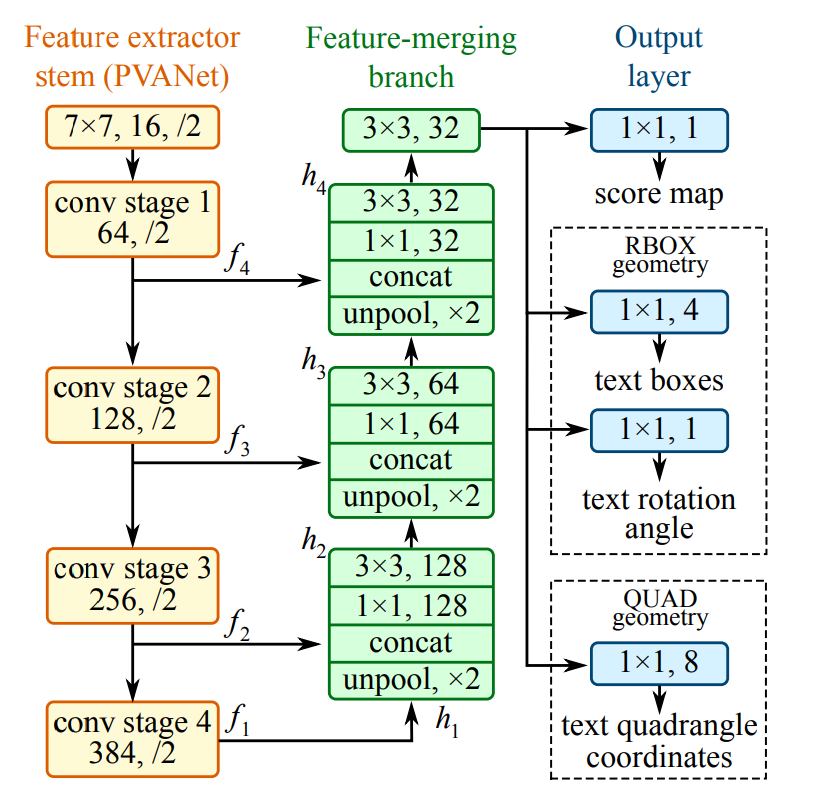
Đặc điểm nổi bật của phương pháp:

* Mang lại tốc độ phát hiện nhanh và chính xác do dùng mạng neuron mà không qua nhiều bước như các phương pháp truyền thống khác.
* Ở phương pháp này, đưa ra một mạng deep FCN trực tiếp đưa ra kết quả cuối cùng. Phương pháp sử dụng một mạng neuron không quá phức tạp do được lược bỏ những thành phần không cần thiết đã vượt qua các phương pháp trước đó cả về độ chính xác lẫn tốc độ xử lý.

Mạng neuron:

Dưới đây là sơ đồ biểu diễn các thành phần của mạng bao gồm 3 thành phần:

* Feature extractor stem: là một convolutional network pre-trained trên tập dữ liệu ImageNet với các tầng conv và pooling xen kẽ nhau
* Feature-merging branch: là bước merge các feature map lại với nhau.
* Output layer: đây là tầng cuối cùng của mạng bao gồm: text boxes, góc nghiêng và tọa độ của text boxes.



Bảng so sánh kết quả so với các phương pháp khác:

|  |  |  |  |
| --- | --- | --- | --- |
| Algorithm | Precision | Recall | F-measure |
| Ours+PVANET2x | **0.8728** | **0.6743** | **0.7608** |
| Ours + PVANET | 0.8356 | 0.6713 | 0.7445 |
| Ours + VGG16 | 0.8167 | 0.6160 | 0.7023 |

**Đánh giá tổng hợp:**

* Tất cả số liệu trên được đánh giá dựa trên bộ dữ liệu MSRA-TD500
* Phương pháp có Precision, Recall, F1 lớn nhất là phương pháp 1.6:

P = 87.6%, R = 76.2%, F1 = 81.5%

* Phương pháp nhanh nhất là EAST+:

FPS = 13.2

* Theo đánh giá cá nhân: phương pháp phù hợp nhất để giải quyết bài toán hiện tại là phương pháp: EAST+ và phương pháp SegLink do tốc độ xử lý nhanh, độ chính xác tương đối cao, và mô hình không quá phức tạp.

**Đánh giá ưu, nhược điểm của phương pháp đã tìm hiểu:**

Các phương pháp truyền thống:

Ưu điểm:

* Không tốn thời gian xây dựng mô hình, traning,…
* Xác định vị trí của đối tượng dựa vào thuật toán và các phương pháp xử lý ảnh.

Nhược điểm:

* Tốc độ xử lý, phát hiện đối tượng chậm, không phù hợp cho ứng dụng realtime.
* Độ chính xác không cao khi áp dụng trên bộ dữ liệu là ảnh tự nhiên do background phức tạp, kích thước và font chữ không tuân theo quy tắc nhất định.

Các phương pháp mới, có xử dụng mạng neuron.

Ưu điểm:

* Độ chính xác khá cao, phát hiện linh hoạt được đối tượng trong tự nhiên nơi có background phức tạp, font chữ và kích thước tùy biến.
* Tốc độ xử lý cao, áp dụng được trong các mô hình, ứng dụng realtime.

Nhược điểm:

* Hầu hết các mô hình tương đối phức tạp. Tốn nhiều thời gian tìm hiểu, xây dựng mô hình, training.
  1. Bài toán Text Recognition

Sau đây là các phương pháp tiếp cận đã tìm hiểu được để xử lí bài toán Text Recognition, các bài báo đều thực nghiệm qua bộ dữ liệu: CASIA-OLHWDB.

* Drawing and Recognizing Chinese Characters with Recurrent Neural Network
* Learning Spatial-Semantic Context with Fully Convolutional Recurrent Network for Online Handwritten Chinese Text Recognition
* Stroke Sequence-Dependent Deep Convolutional Neural Network for Online Handwritten Chinese Character Recognition
* High Performance Offline Handwritten Chinese Character Recognition Using GoogLeNet and Directional Feature Maps
  1. Bài toán Translation

Dịch máy (Machine Translation) là một nhánh của xử lý ngôn ngữ tự nhiên thuộc phân ngành trí tuệ nhân tạo, nó là sự kết hợp giữa ngôn ngữ, dịch thuật và khoa học máy tính. Như tên gọi, dịch máy thực hiện dịch một ngôn ngữ này (ngôn ngữ nguồn) sang một hoặc nhiều ngôn ngữ khác (ngôn ngữ đích) một cách tự động, không có sự can thiệp của con người trong quá trình dịch. Dịch máy là một bài toán đã được nghiên cứu từ đầu năm 1950, cho đến nay đã có rất nhiều công trình nghiên cứu cũng như giải pháp để giải quyết bài toán dịch máy, các phương pháp nổi tiếng có thể nhắc đến như: dịch máy thống kê, sử dụng các thuật toán dóng hàng từ, … Cho đến năm 2014, sự bùng nổ của mạng Neuron, việc giải quyết bài toán dịch máy bằng phương pháp mạng Neuron cho độ hiểu quả cao hơn nhiều so với những phương pháp trước đây. Sau đây tôi sẽ trình bày chi tiết mô hình Neural Machine Translation và mô hình tôi sẽ sử dụng trong đồ án của mình.

* + 1. Một số khái niệm cơ bản
* Neural Network: Mạng nơ-ron nhân tạo (*Neural Network* - **NN**) là một mô hình lập trình rất đẹp lấy cảm hứng từ mạng nơ-ron thần kinh. Kết hợp với các kĩ thuật học sâu (*Deep Learning* - **DL**), NN đang trở thành một công cụ rất mạnh mẽ mang lại hiệu quả tốt nhất cho nhiều bài toán khó như nhận dạng ảnh, giọng nói hay xử lý ngôn ngữ tự nhiên. Cấu trúc cơ bản của NN được mô tả như hình bên dưới:

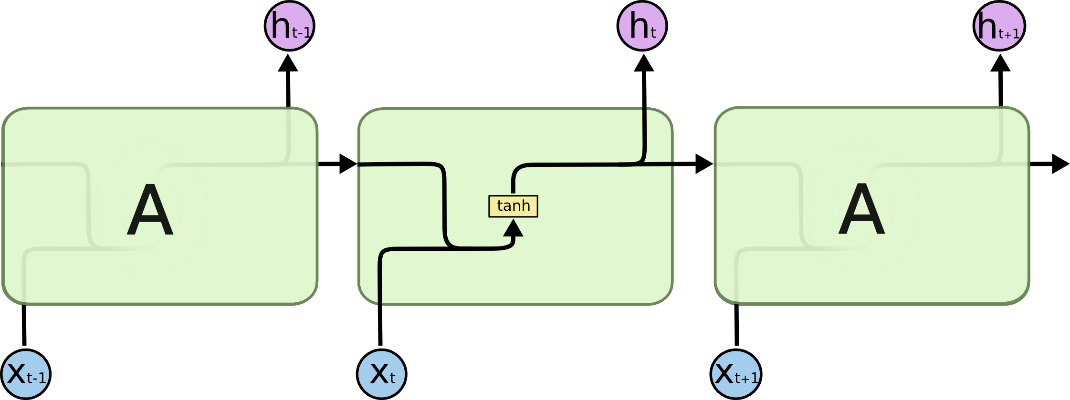
Một mạng NN sẽ có 3 loại tầng:

* + **Tầng vào** (*input layer*): Là tầng bên trái cùng của mạng thể hiện cho các đầu vào của mạng.
  + **Tầng ra** (*output layer*): Là tầng bên phải cùng của mạng thể hiện cho các đầu ra của mạng.
  + **Tầng ẩn** (*hidden layer*): Là tầng nằm giữa tầng vào và tầng ra thể hiện cho việc suy luận logic của mạng.

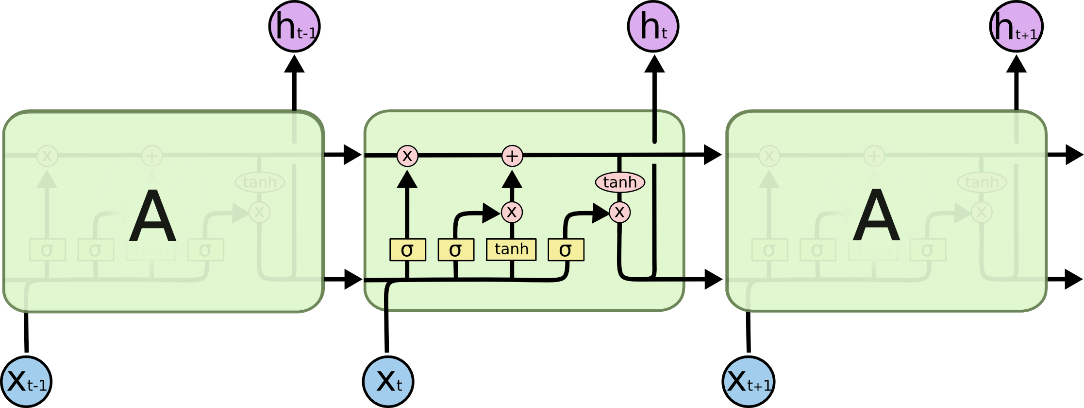
Một mạng nơ-ron chỉ có 1 tầng vào và 1 tầng ra nhưng có thể có nhiều tầng ẩn.



* Reccurent Neural Network (RNN): RNN ra đời với ý tưởng chính là sử dụng một bộ nhớ để lưu lại thông tin từ từ những bước tính toán xử lý trước để dựa vào nó có thể đưa ra dự đoán chính xác nhất cho bước dự đoán hiện tại, cấu trúc này phù hợp với dữ liệu dạng chuỗi. RNN được ứng dụng và thành công ở rất nhiều bài toán, đặc biệt là ở lĩnh vực NLP (xử lý ngôn ngữ tự nhiên). Trên lý thuyết thì đúng là RNN có khả năng nhớ được những tính toán (thông tin) ở trước nó, nhưng mô hình RNN truyền thống không thể nhớ được những bước ở xa do bị mất mát đạo hàm nên những thành công của mô hình này chủ yếu đến từ một mô hình cải tiến khác là LSTM (Long Short-Term Memory)
* Long Short-Term Memory (LSTM) về cơ bản cũng giống với RNN truyền thống ngoài việc thêm các cổng tính toán ở hidden layer để quyết định giữ lại các thông tin nào. Mọi mạng hồi quy đều có dạng là một chuỗi các mô-đun lặp đi lặp lại của mạng nơ-ron. Với mạng RNN chuẩn, các mô-dun này có cấu trúc rất đơn giản, thường là một tầng *tanh*.



LSTM cũng có kiến trúc dạng chuỗi như vậy, nhưng các mô-đun trong nó có cấu trúc khác với mạng RNN chuẩn. Thay vì chỉ có một tầng mạng nơ-ron, chúng có tới 4 tầng tương tác với nhau một cách rất đặc biệt.



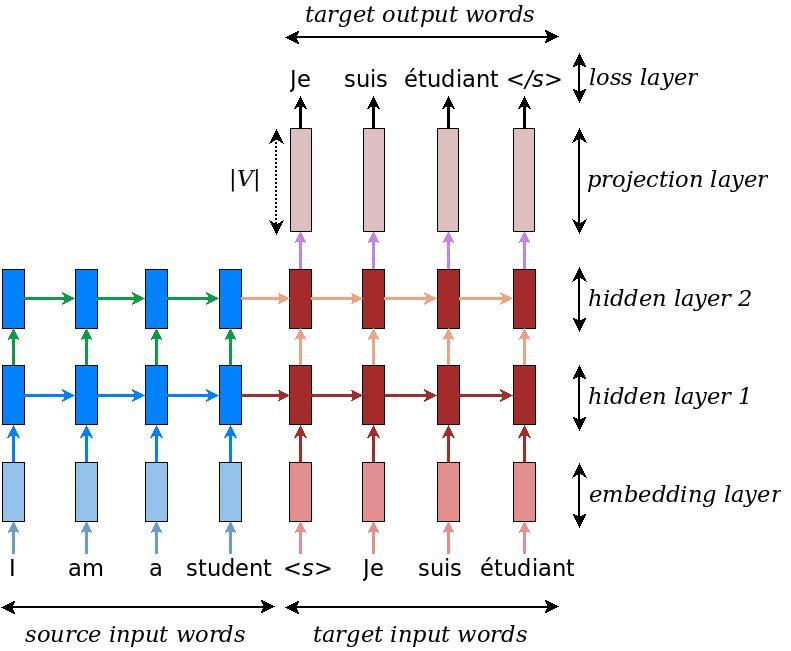
Ý tưởng cốt lõi của LSTM là trạng thái tế bào (cell state) Trạng thái tế bào là một dạng giống như băng truyền. Nó chạy xuyên suốt tất cả các mắt xích (các nút mạng) và chỉ tương tác tuyến tính đôi chút. Vì vậy mà các thông tin có thể dễ dàng truyền đi thông suốt mà không sợ bị thay đổi. LSTM có khả năng bỏ đi hoặc thêm vào các thông tin cần thiết cho trạng thái tế báo, chúng được điều chỉnh cẩn thận bởi các nhóm được gọi là cổng (gate). Các cổng là nơi sàng lọc thông tin đi qua nó, chúng được kết hợp bởi một tầng mạng sigmoid và một phép nhân.

* GRU: Ý tưởng của GRU cũng khá giống với LSTM: GRU chỉ có 2 cổng: cổng thiết lập lại và cổng cập nhập. Cổng thiết lập lại sẽ quyết định cách kết hợp giữa đầu vào hiện tại với bộ nhớ trước, còn cổng cập nhập sẽ chỉ định có bao nhiêu thông tin về bộ nhớ trước nên giữa lại. Như vậy RNN thuần cũng là một dạng đặc biệt của GRU, với đầu ra của cổng thiết lập lại là 1 và cổng cập nhập là 0. Cùng chung ý tưởng sử dụng cơ chế cổng điều chỉnh thông tin, nhưng chúng khác nhau ở mấy điểm sau:
  + GRU có 2 cổng, còn LSTM có tới 3 cổng.
  + GRU không có bộ nhớ trong c\_t*ct*​ và không có cổng ra như LSTM.
  + 2 cổng vào và cổng quên được kết hợp lại thành cổng cập nhập z*z* và cổng thiết lập lại r*r* sẽ được áp dụng trực tiếp cho trạng thái ẩn trước.
  + GRU không sử dụng một hàm phi tuyến tính để tính đầu ra như LSTM.

Về cơ bản: Mặc dù GRU hay LSTM đều có những điểm mạnh riêng, nhưng về cơ bản mạng GRU sẽ cho tốc độ huấn luyện nhanh hơn so với mạng LSTM do GRU có ít tham số hơn, nhưng khi có đầy đủ dữ liệu huấn luyện thì LSTM lại tỏ ra mạnh mẽ hơn.

* Sequence-to-sequence (seq2seq): Mô hình seq2seq cơ bản bao gồm hai mạng neural thành phần được gọi là mạng mã hóa (encoder) và mạng giải mã (decoder) để sinh ra chuỗi đầu ra t1:m từ một chuỗi đầu vào x1:n. Mạng neural encoder mã hóa chuỗi đầu vào thành một vector c có độ dài cố định. Mạng neural decoder sẽ lần lượt sinh từng từ trong chuỗi đầu ra dựa trên vector c và những từ được dự đoán trước đó cho tới khi gặp từ kết thúc câu. Trong mô hình seq2seq, chúng ta có thể sử dụng các kiến trúc mạng khác nhau cho thành phần encoder và decoder như RNN hay mạng neural convolutional.
  + 1. Mô hình

Mô hình dịch là một mô hình ngôn ngữ, trước tiên hệ thống đọc một câu nguồn sử dụng tầng Encode để tạo ra một vector đặc trưng để biểu diễn ý nghĩa của câu. Sau đó là một tầng Decode xử lý vector trên để tạo ra một bản dịch (minh họa như hình dưới). Theo đó, phương pháp giúp giải quyết vấn đề dịch máy là đưa ra một chuỗi đầu vào và sinh ra một chuỗi đầu ra. Các mô hình dịch máy khác nhau chủ yếu về kiến trúc của chúng, thông thường RNN được sử dụng phổ biến cho mô hình dịch máy nó được sử dụng cho cả bộ Encode và Decode. Tuy nhiên, các mô hình sẽ khác nhau về: tính định hướng (một hoặc hai chiều), độ sâu (một tầng hay nhiều tầng), loại cell nhớ (LSTM, GRU)



**2.3.2. Các độ đo đánh giá**

Có hai độ đo tự động thường hay được sử dụng để đánh giá kết quả cho bài toán translate đó là BLEU [17] và METEOR [18].

* Độ đo BLEU: BLEU (Bilingual Evaluation Understudy) [17]

là độ đo đánh giá độ giống của câu văn cần đánh giá so với tập các câu văn tham khảo, điểm BLEU có giá trị trong khoảng từ 0 đến 1 (nếu tính theo % thì sẽ có giá trị trong khoảng từ 0 đến 100), điểm càng cao thì câu văn cần đánh giá càng giống với câu văn tham khảo. Mục tiêu của BLEU là so sánh các cặp n từ liền nhau trong câu cần đánh giá với các cặp n từ liền nhau trong các câu tham khảo và đếm số cặp trùng khớp. Số cặp trùng khớp càng nhiều cho thấy câu cần đánh giá này càng giống với câu tham khảo.

pn =

Trong đó: p1 ứng với các từ đơn, p2 ứng với các từ ghép, p3 ứng với chuỗi 3 từ liên tiếp, p4 ứng với chuỗi 4 từ liên tiếp.

Công thức tính BLEU như sau:

BLEU = BP.exp ()

BP =

Trong đó, r là tổng số lượng cặp tương ứng từ câu cần đánh giá tới câu tham khảo và c là tổng số phần tử trong câu cần đánh giá.

* Độ đo METEOR

METEOR (Metric for Evaluation of Translation with Explicit Ordering) [18] chỉ tính toán trên từ đôi - cụm 2 từ liên tiếp. Cách tính điểm METEOR như sau:

Trong đó: m là số lượng từ đôi trong câu cần đánh giá mà tìm thấy trong các câu tham khảo, Wt là tổng số lượng từ đôi trong câu cần đánh giá, Wr là tổng số lượng từ đôi trong câu tham khảo.

METEOR sử dụng p để tính điểm trừ của câu cần đánh giá, giá trị này càng nhỏ càng tốt, p thể hiện số lượng các cặp từ có ánh xạ mà không kề nhau trong câu cần đánh giá, c là số lượng các cụm từ có ánh xạ trong câu ứng viên, giá trị c càng nhỏ thì các cặp kề nhau càng nhiều, kết quả cuối cùng của METEOR là:

* 1. Google Cloud Vision API
     1. REST API là gì?

REST (Representational State Transfer) API (Application Programming Interface) là một tiêu chuẩn dùng trong việc thết kế các thiết kế API cho các ứng dụng web để quản lý các resource. REST là một trong những kiểu thiết kế API được sử dụng phổ biến nhất ngày nay. Trọng tâm của REST quy định cách sử dụng các HTTP method (như GET, POST, PUT, DELETE, …) và cách định dạng các URL cho ứng dụng web để quản lý các resource.

* + 1. Google Cloud Vision là gì?

Google Cloud Vision (GCV) cung cấp các mô hình đã được training sẵn dưới dạng API một cách đơn giản giúp các lập trình viên có thể dễ dàng xử lý và phân tích hình ảnh. GCV cho phép các nhà phát triển hiểu được nội dung của hình ảnh bằng cách gói gọn các mô hình học máy một cách hiệu quả và dễ sử dụng thông qua REST API. Các ưu điểm chính của GCV:

* Dễ dàng phát hiện các nhóm đối tượng rộng lớn trong hình ảnh từ hoa, động vật, con người, văn bản, …
* Optical Character Recognition (OCR): cho phép giải quyết bài toán phát hiện văn bản trong hình ảnh, tự động nhận dạng ngôn ngữ với rất nhiều ngôn ngữ trên thế giới.
* GCV cho độ chính xác và tốc độ phát hiện cao, phù hợp với các ứng dụng real time.
* Có thể kết hợp với việc sử dụng sức mạnh của Google Image Search (công cụ tìm kiếm hình ảnh của Google) để hỗ trợ việc tìm kiếm các hình ảnh liên quan.
* GCV còn cho phép bạn phát hiện các nội dung hình ảnh không phù hợp từ nội dung người lớn đến nội dung bạo lực.

Qua những yêu điểm đó, cùng với việc xem xét phạm vi của đồ án, tôi đã sử dụng GCV để giải quyết bài toán phát hiện và nhận dạng văn bản trong hình ảnh tự nhiên.

* 1. Định hướng giải quyết vấn đề

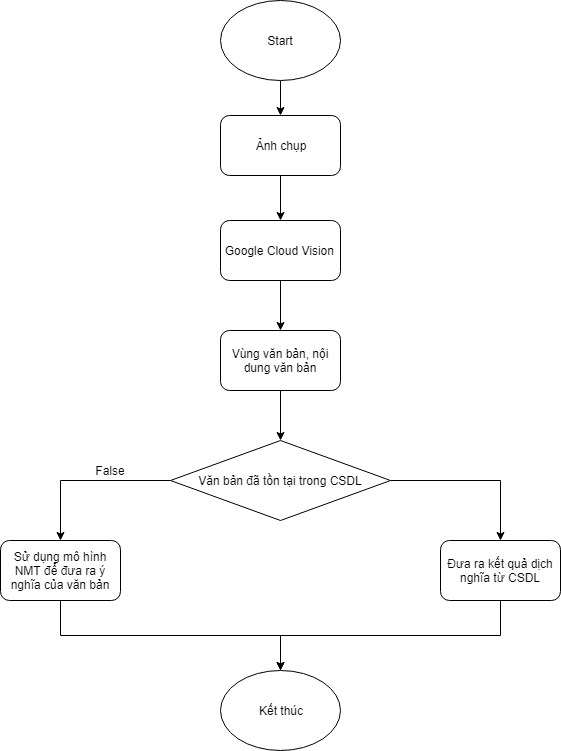
Qua quá trình tìm hiểu, cùng với việc xem xét phạm vi, thời gian thực hiện đồ án tốt nghiệp, cùng với những kiến thức đã tích lũy được trong quá trình học và tìm hiểu, tôi quyết định giải quyết bài toán phát hiện và nhận dạng văn bản trong tự nhiên theo hướng sau:

* Sử dụng Google Cloud Vision API cho quá trình phát hiện và nhận dạng hoành phi, câu đối
* Sử dụng mô hình đơn giản hóa Neural Machine Translation của google cho quá trình dịch nghĩa ra Tiếng Việt.
* Sử dụng dữ liệu trong sách 5000 Hoành Phi câu đối Hán Nôm do Nhà xuất bản Văn Hóa thông tin tổng hợp và xuất bản để training và test cho mô hình dịch máy.

Quá trình thử nghiệm và xây dựng mô hình cụ thể được trình bày ở Chương 3.

# CHƯƠNG 3: QUÁ TRÌNH THỬ NGHIỆM VÀ XÂY DỰNG ỨNG DỤNG

* 1. Tổng quan hệ thống và công nghệ sử dụng



Sơ đồ biểu diễn tổng quan hệ thống

Tổng quan hệ thống:

* Input: Ảnh chụp thực tế của những bộ hoành phi, câu đối.
* Output: Ý nghĩa của nó bằng Tiếng Việt.

Trước tiên, hình ảnh được upload lên server từ điện thoại di động thông qua ứng dụng Translator, hình ảnh có thể được chụp hoặc tải lên từ bộ nhớ điện thoại. Tiếp theo, nó sẽ được xử lý bằng Google Cloud Vision để trích rút các văn bản (ở đây là các bộ hoành phi, câu đối). Các đoạn văn bản được tìm kiếm trong bộ cơ sở dữ liệu (CSDL) có sẵn được lấy từ cuốn “5000 Hoành Phi, câu đối Hán Nôm”. Nếu văn bản đã tồn tại trong bộ CSDL thì đưa ra kết quả, ngược lại ta đưa nó vào mô hình dịch để dự đoán ý nghĩa của văn bản rồi đưa ra kết quả. Kết quả được trả về dưới dạng Json và hiển thị trên màn hình điện thoại di động.

Ngôn ngữ chủ yếu tôi sử dụng là Python, các thư viện phục vụ cho đồ án mà tôi có sử dụng Keras (Tensorflow backend), numpy, pandas,…

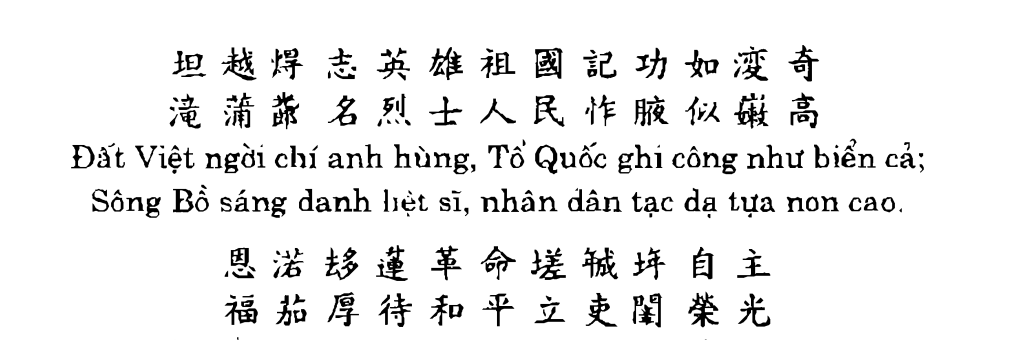
Ứng dụng được viết trên nền tảng Android.

IDE sử dụng: Pycharm, Android Studio, Visual Studio Code.

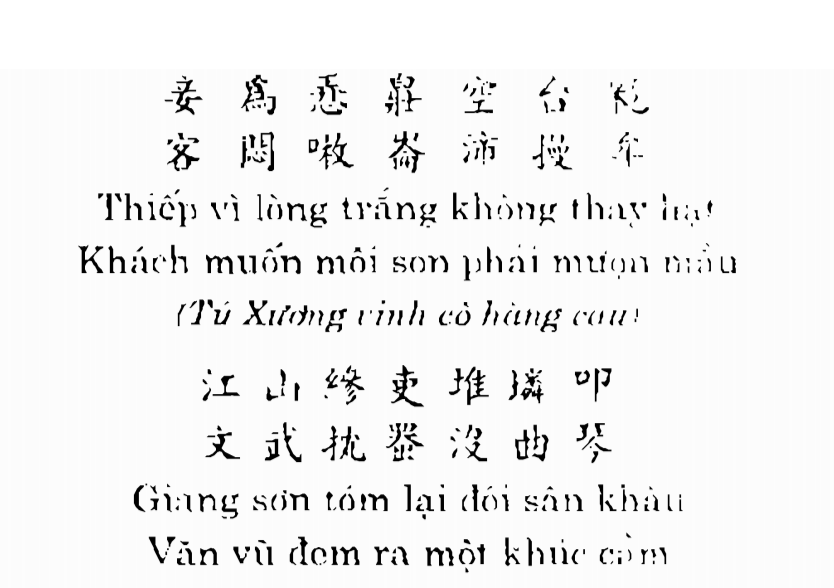
* 1. Mô hình translation
     1. Quá trình xử lý, làm sạch dữ liệu

Dữ liệu tôi dùng làm bộ dữ liệu huấn luyện cho mô hình dịch máy được lấy từ cuốn sách “5000 Hoành Phi, câu đối Hán Nôm” đây là cuốn sách đã được xuất bản khá lâu (từ năm 2002) rồi và không còn bán trên thị trường. Tôi chỉ tìm được sách dưới dạng pdf, các trang sách được scan và trình bày dưới dạng pdf nên chất lượng khá kém. Để có thể sử dụng làm bộ dữ liệu huấn luyện cho mô hình tôi đã thực hiện tiền xử lý theo các bước sau:

* Tách mỗi trang sách (dạng pdf) thành các ảnh riêng biệt
* Sử dụng Google Cloud Vision cho mỗi ảnh nhằm mục đích chuyển văn bản dạng ảnh thành văn bản thực.



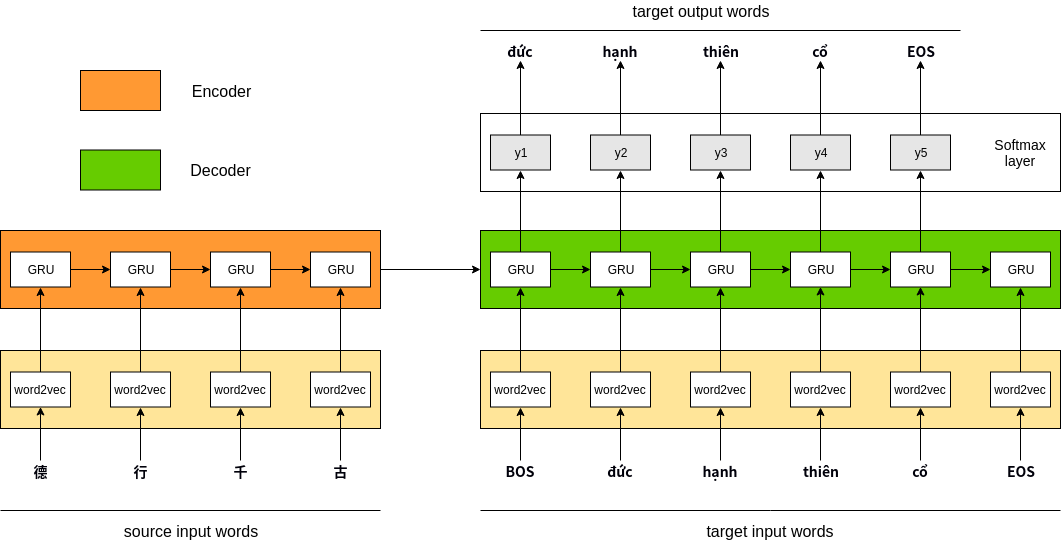
Có nhiều trang hình ảnh scan rất sắc nét



Nhưng có một số trang bị mờ hoặc nhiễu làm giảm độ chính xác của GCV.

Sau khi sử dụng GCV để chuyển ảnh thành văn bản, tôi nhận thấy mặc dù kết quả nhận dạng chữ Hán rất tốt nhưng với bản dịch tiếng việt có nhiều lỗi nhỏ về dấu ví dụ: “đầu xuân” bị chuyển thành “dâu xuân”, “tơ duyên” bị phát hiện thành “td duyên”, … Điều này làm ảnh hưởng rất lớn đến ý nghĩa của câu văn, vì vậy sau khi sử dụng GCV để chuyển hình ảnh thành văn bản thì tôi phải làm thêm bước sửa lỗi chính tả bằng mắt thường, quá trình này tuy đơn giản nhưng mất nhiều thời gian.

* + 1. Quá trình build model

Ở đây, tôi cài đặt và huấn luyện mô hình nhận đầu vào là câu Tiếng Hán (Nôm), đầu ra là một câu tiếng Việt. Cấu trúc mô hình được mô tả như hình dưới:

Như hình trên module nhận đầu vào thứ nhất là chuỗi các từ Tiếng Hán trong câu: x1, x2, x3, ….xn. và đầu ra cho các chuỗi từ đó bằng tiếng Việt. Phần Encoder và Decoder sử dụng một tầng mạng GRU. Đây là mô hình được đơn giản hóa từ Google Neural Machine Translation [16] ở đây, mô hình google giới thiệu sử dụng nhiều tầng (số lượng tầng thực tế có thể là 4, 6 hoặc 8 tầng) LSTM thay thế cho GRU và có kết hợp sử dụng kỹ thuật Attention để tăng độ chính xác của mô hình. Đầu ra tại timestep cuối cùng (trạng thái ẩn cuối cùng) của GRU encoder được đưa vào làm trạng thái khởi tạo cho GRU Decoder. Ở đây em thêm vào 2 từ là “BOS” và “EOS” vào trong bộ từ vựng để đánh dấu bắt đầu và kết thúc một câu, do các câu có độ dài khác nhau nên việc thêm vào EOS là cần thiết để biết chính xác khi nào là lúc kết thúc câu. Đầu ra của GRU Decoder sẽ làm đầu vào cho tầng softmax để tính phân phối xác xuất từ. Quá trình training model này em làm bằng cách tối thiểu hóa hàm mục tiêu “Categorical Cross Entropy”. Trong giai đoạn Predict, phần Encoder tương tự như giai đoạn training, ở phần Decoder target input thứ nhất là BOS để làm trạng thái khởi tạo cho GRU Decoder. lấy tầng có xác suất cao nhất sau tầng softmax làm đầu vào cho timestep thứ 2… cứ thể sinh ra output cho đến khi xuất hiện từ “EOS” hoặc sinh ra đến điểm giới hạn số từ trong câu thì dừng lại.

* + 1. Quá trình training
    2. Quá trình testing
    3. Quá trình đánh giá mô hình
  1. Xây dựng app android

# CHƯƠNG 4: KẾT LUẬN VÀ ĐỊNH HƯỚNG PHÁT TRIỂN