# **CHƯƠNG 1: KHẢO SÁT, ĐÁNH GIÁ VÀ LỰA CHỌN PHƯƠNG PHÁP GIẢI QUYẾT BÀI TOÁN.**

* 1. **Text detection**

Sau đây là các phương pháp tiếp cận để giải quyết bài toán text detection mà em đã tìm hiểu được, trong đó 3 phương pháp đầu tiên giải quyết bài toán theo hướng toán học cổ điển mà không sử dụng đến 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 [1].**

Ý tưởng chính của phương pháp:

* Phương pháp này sử dụng SWT kết hợp với các phương pháp khác để tăng đọ 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 . Trong bài báo này tập trung vào việc phát hiện text trong những bức ảnh thực tế, do đó gặp rất nhiều khó khăn do có nhiều biến thể của đoạn văn bản (kích thước, font chữ, góc nghiêng,…).

Thuật toán này bao gồm 4 bước:

* Component Extraction: Sử dụng *Edge detection* để tạo ra *Edge map*  làm đầu vào cho module *SWT* để tạo ra *SWT image*. Những pixel lân cận trong *SWT image* được nhóm lại với nhau để tạo thành các components.
* Component Analysis: Nhiều *Component* 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 *train* để định nghĩa và loại bỏ các component không phải là text.
* Candidates Linking: Các *Component* được giữ lại coi như là các C*haracter Candidates*. Bước đầu tiên của candidate linking là liên kết các character candidates thành từng cặp cho 2 *Candidates* liền kề nhau nếu như chúng giống nhau về hình dạng và màu sắc. Các cặp candidates được tổng hợp lại thành các chuỗi (chains) theo kiểu đệ quy.
* Chains Analysis: Phân tích các chuỗi để xác định chính xác có thuộc vùng kí tự hay không.
  + 1. **Scene Text Detection via Holistic, Multi-Channel Prediction [2].**

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 class (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 2 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. Vì vậy, phương pháp trước đây 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 |

* + 1. **Orientation Robust Text Line Detection in Natural Scene Images [3].**

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 |

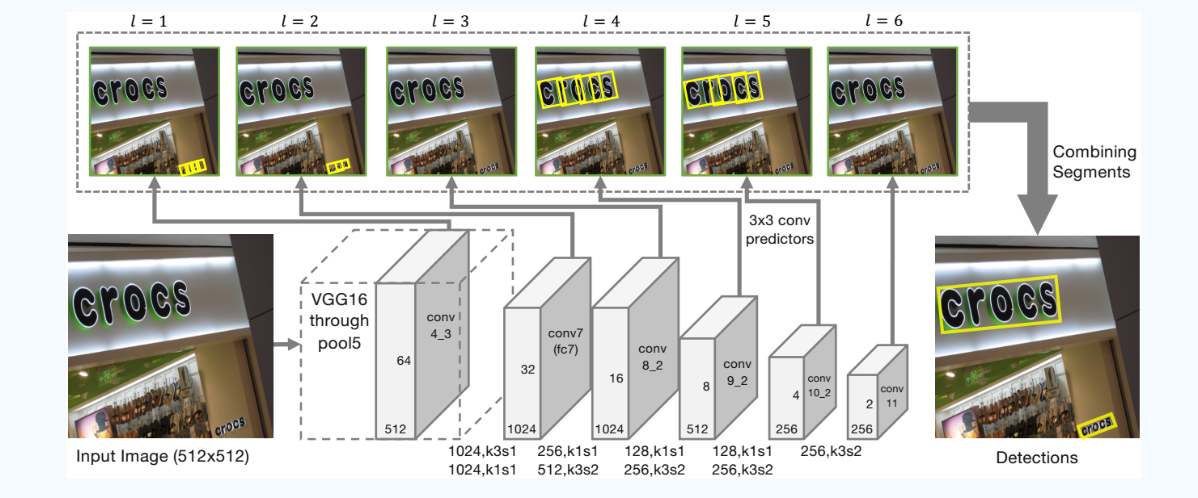
* + 1. **Detecting Oriented Text in Natural Images by Linking Segments [4].**

Đặc điểm nổi bật của phương pháp: phương pháp đã giải quyết tốt vấn đề phát hiện text trong hình ảnh tự nhiên, nó giải quyết tốt 2 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 Latin ví dụ là chữ Trung Quốc.
* Tốc độ xử lý nhanh, phù hợp với các ứng dụng realtime.

Phương pháp tiếp cận:

* 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.
* Cách tiến hành:



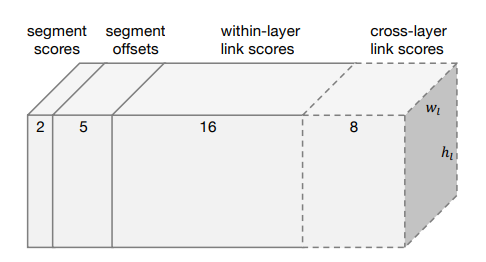
Bước 1:

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



* 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** |

* + 1. **Multi-Oriented Text Detection with Fully Convolutional Networks [5].**

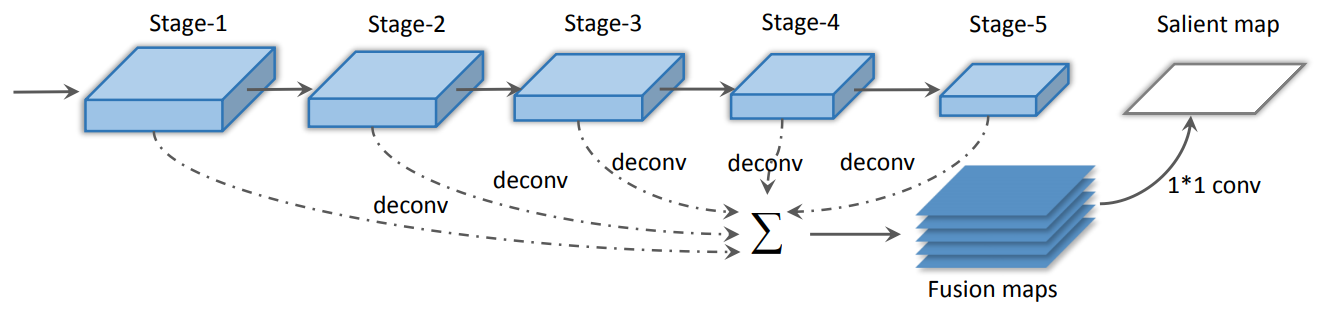
Ý tưởng chính của phương pháp:

* 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.

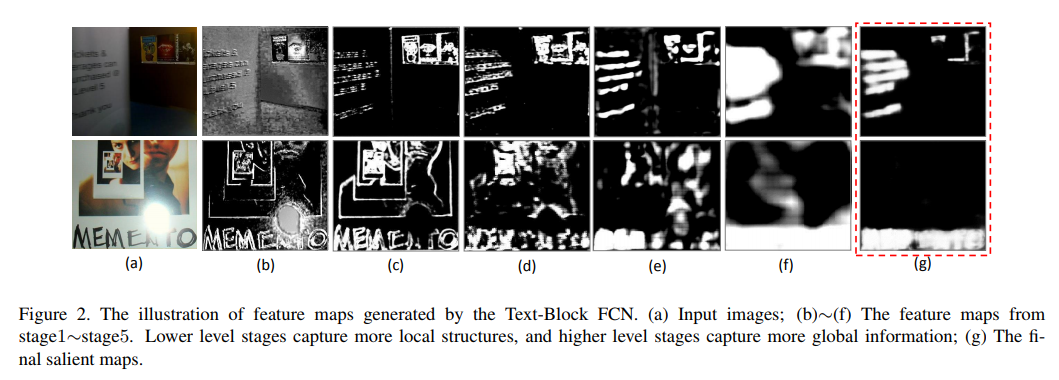
Các bước thực hiện: gồm 3 bước:

Bước 1: Text block detection:

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



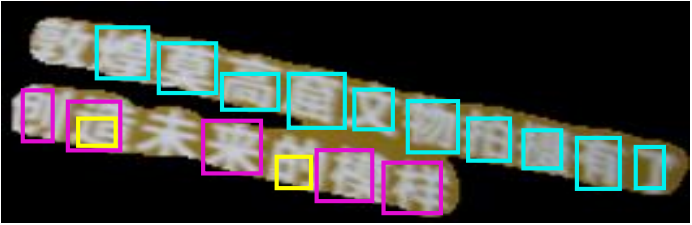
Mô tả phương pháp:



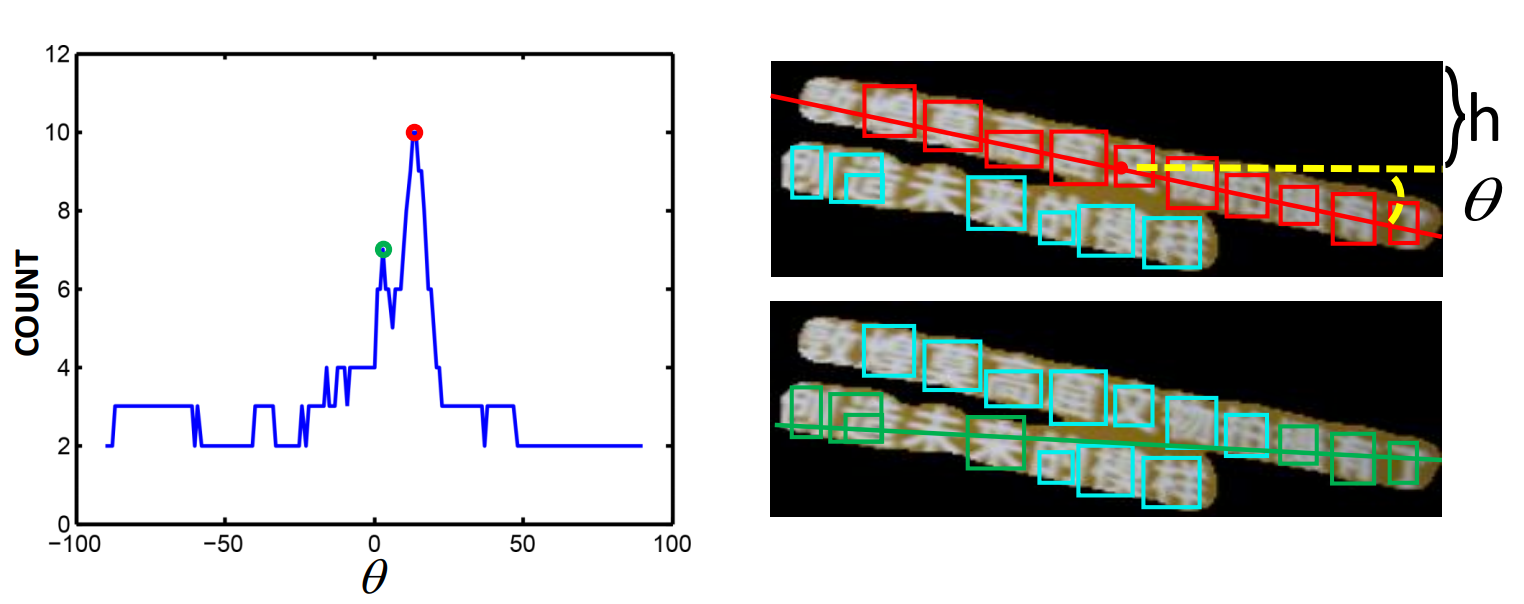
Các vùng text được phát hiện ở bước 1 được gọi là các **Text Block.**

Bước 2: Text Line Candidate Generation:

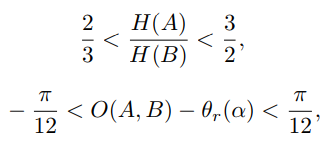
* Trích xuất kí tự: bước này sử dụng phương pháp MSER để trích xuất ra các kí tự từ **Text Box.**



* 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:

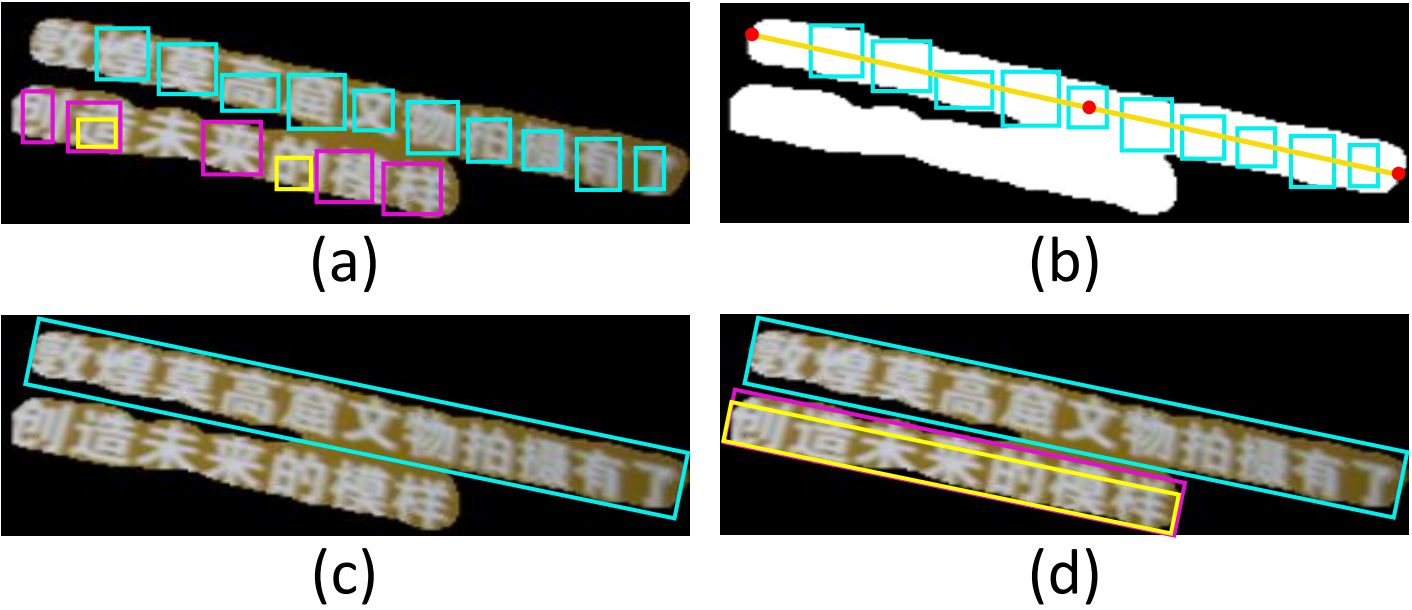


* Tạo ra **Text Line:**
  + Đầu tiên: các kí tự được nhóm lại với nhau bằng điều kiện sau:



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:

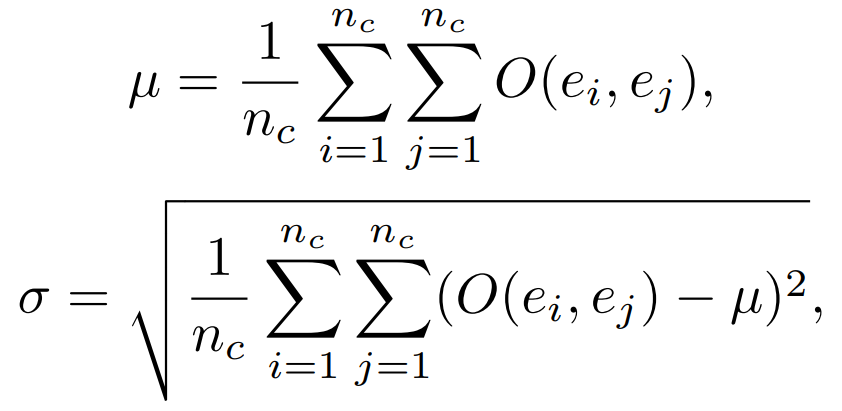
* Mục đích của bước này là loại bỏ các TextLine candidates sai.
* Để làm điều đó paper sử dụng model “Character-Centroid FCN”
* Mô tả Input-Output của model:
  + Input: **Text Line Candidates**
  + Output: The probability maps of character centroids



* 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 [6]**

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**

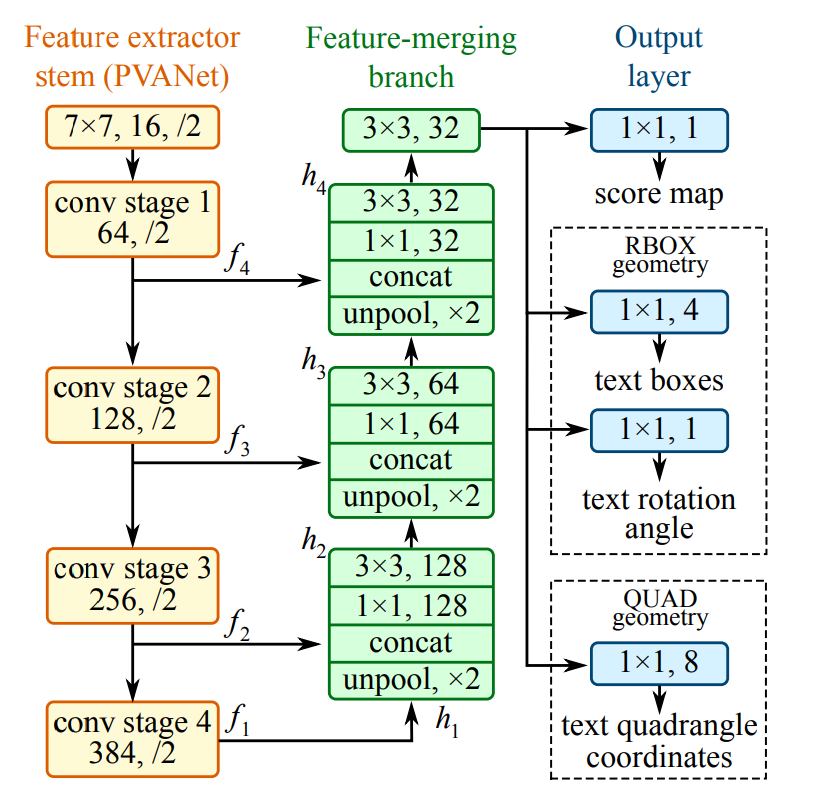
Đặc điểm nổi bật của phương pháp:

* 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.

Đánh giá tổng quát và lựa chọn phương pháp có thể sử dụng: thông qua 4 tiêu chí sau:

* Tốc độ:
* Độ chính xác:
* Độ phức tạp của mô hình:
* Phù hợp với kiến thức, khả năng của mình.

Đánh giá:

1. Tốc độ: sắp xếp theo thứ tự tăng dần FPS của từng phương pháp.

EAST(**13.2**) > Segment & Linking(**8.9**) > Corner & Region(**5.7**) **>** Fully convolutional(**0.48**).

1. Độ chính xác: sắp xếp theo thứ tự tăng dần F1 của từng phương pháp.

Corner & Region(**0.82**) > Segment & Linking(**0.77**) > EAST(**0.76**) > Fully convolutional(**0.74**).

1. Độ phức tạp: được sắp xếp theo thứ tự tăng dần:

Fully convolutional < EAST < Segment & Linking < Corner & Region.

Đánh giá tổng quát: Để áp dụng cho bài toán đặt ra ngay từ đầu là xây dựng app di động. nên tốc độ xử lý của phương pháp là điều quan trọng nhất, do đó ta sẽ bỏ qua các phương pháp truyền thống. đối với 4 phương pháp sử dụng mạng neuron, Tôi đề xuất sử dụng phương pháp EAST bởi vì:

* Tốc độ xử lý cao (FPS = 13.2)
* Độ chính xác cao (F1 = 0.76)
* Việc xây dựng lại model từ các paper là rất khó khăn, do đó tôi ưu tiên sử dụng phương pháp đã có sẵn source code, có thể implement và retrain model theo tham số và dataset mà mình muốn.
* Code của phương pháp được publish trên github, khi đó ta có thể customize code để xây dựng một service riêng cho ứng dụng của mình.

<https://github.com/argman/EAST>

* Cũng có thể demo phương pháp tại:

http://east.zxytim.com/

Kết quả sau khi áp dụng API Google Vision:

* Chỉ cần sử dụng API này để phát hiện và nhận diện một cách chính xác các đối tượng văn bản trong ảnh.
* Vì vậy ta sẽ sử dụng luôn Google Vision API cho quá trình Detection và Recognition. Sử dụng Google Translate API cho quá trình dịch văn bản sang Tiếng Việt.
* Mục đích của đồ án: Tập trung thiết kế và phát triển ứng dụng OCR.

Các giai đoạn phát triển:

Giai đoạn 1: Khảo sát dự án, phân tích tính ứng dụng và tính khả thi của dự án.

* Nhu cầu đi du lịch ngày càng trở nên phổ biến, một trong những hình thức du lịch đó là du lịch tâm linh. Khi đến các chùa, miếu người ta thường tò mò về ý nghĩa của những câu, những chữ tượng hình được in, khắc trên các cột, cửa trong khu vực đền, chùa. Cùng với sự định hướng của giáo viên hướng dẫn, cũng những kiến thức cơ bản đã có, em muốn xây dựng hệ thống OCR giúp hỗ trợ phát hiện và dịch được ý nghĩa của những câu chữ đó.
* ứng dụng được phát triển trên nền tảng di động, tiện lợi và dễ dàng sử dụng trong khi đi du lịch
* Vấn đề mà ứng dụng cần giải quyết được:
  + Giao diện đẹp mắt, dễ sử dụng.
  + Thực hiện tác vụ: phát hiện văn bản, nhận dạng, dịch một cách đầy đủ và chính xác.
  + Ứng dụng chạy mượt mà, thực hiện các tác vụ trong thời gian ngắn, ít bị delay.
  1. **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.

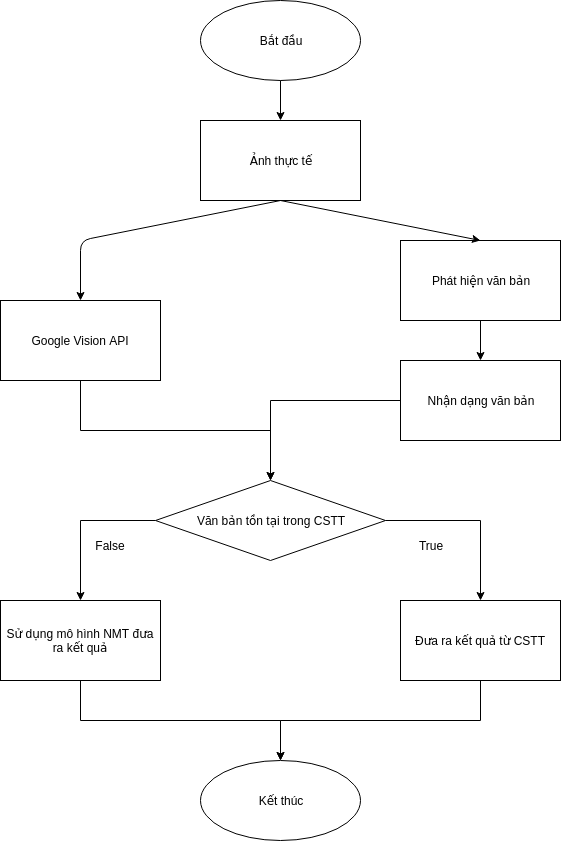
* Drawing and Recognizing Chinese Characters with Recurrent Neural Network
* Learning Spatial-Semantic Context with Fully Convolutional Recurrent Network for Online Handwritten Chinese Text Recognition
* Stroke Sequence-Dependent Deep Convolutional Neural Network for Online Handwritten Chinese Character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

Các phương pháp giải quyết bài toán Text Recognition:

<https://github.com/diaomin/crnn-mxnet-chinese-text-recognition>

ref: <https://github.com/senlinuc/caffe_ocr>

# **CHƯƠNG 2: MÔ HÌNH THỬ NGHIỆM**



## **2.1. Google Vision API**

Giải quyết 2 bài toán text detecttion và text recognition.

Input: Ảnh thực tế

Output:

* Tọa độ của vùng chứa text trong ảnh (ground truth).
* Xác suất của mỗi ground truth.
* Symbol trong mỗi ground truth.
* Xác suất của mỗi symbol.

## **2.2. Phát hiện và nhận dạng văn bản**

Bài toán đã được tìm hiểu và trình bày ở Chương 1

## **2.3. Neural Machine Translation**

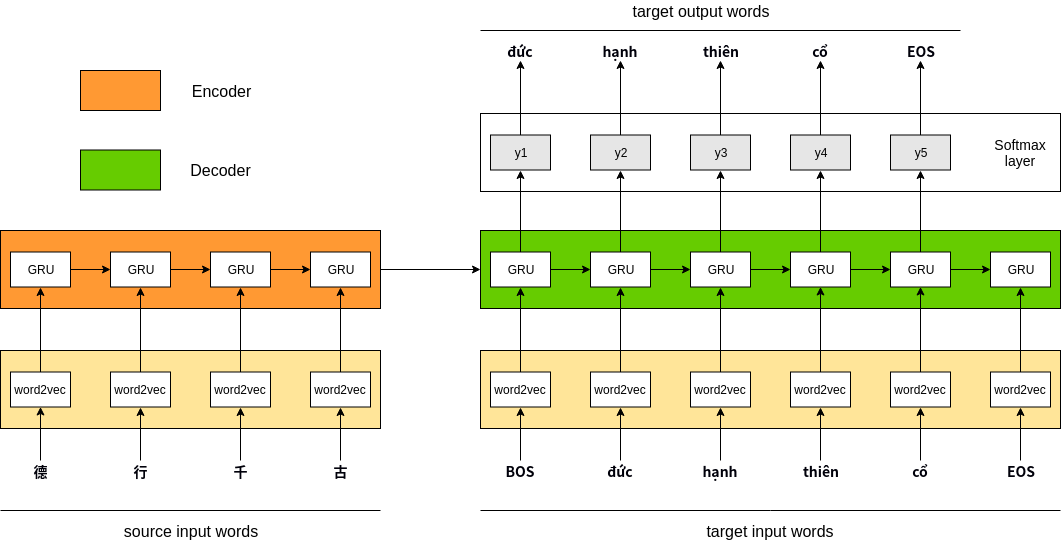
Mục đích của phần này là giải quyết bài toán dịch nghĩa của các câu lấy được từ module trên và các câu này không tồn tại hệ cơ sở tri thức có sẵn.

Bộ dữ liệu được sử dụng trong giai đoạn này là:

“5000 Hoành phi câu đối Hán Nôm - Nhà xuất bản Văn Hóa Thông Tin”

**2.3.1. Mô hình**

Ở đây, em 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:



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.

**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:

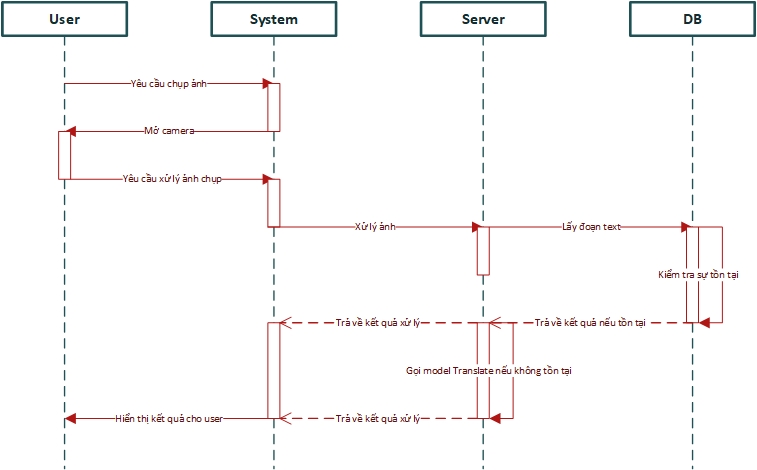
Trong đó: m là số lượng từ đôi trong câu cần đánh giá mà tìm thấy trong các câu tham khảo, Wt là tổng số lượng từ đôi trong câu cần đánh giá, Wr là tổng số lượng từ đôi trong câu tham khảo.

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:

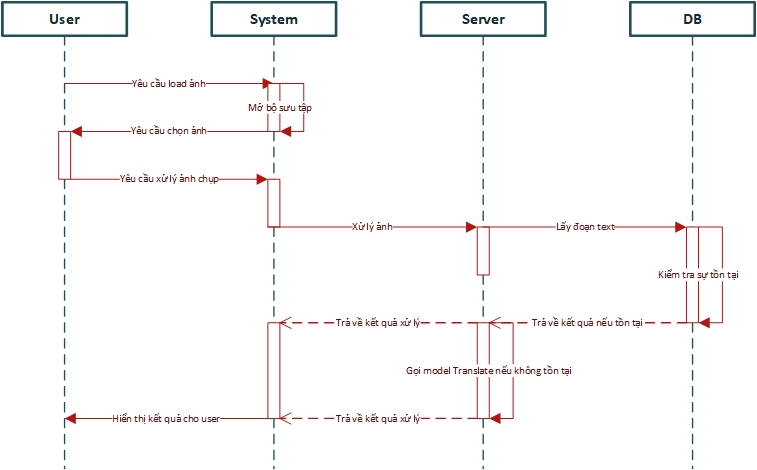
# CHƯƠNG 3: XÂY DỰNG ỨNG DỤNG DEMO

## **3.1. Phân tích thiết kế hệ thống**

- Sơ đồ tuần tự: chức năng dịch thông tin từ ảnh chụp



- Sơ đồ tuần tự: chức năng dịch thông tin trong ảnh được load lên từ máy.



## **3.2. Thiết kế giao diện người dùng**

## **3.3. Công nghệ sử dụng**

- Ngôn ngữ lập trình Java:

- Công nghệ sử dụng: Android SDK

- IDE: Android Studio

**Phụ Lục**

Bảng kế hoạch thời gian thực hiện đồ án:

* Chia ứng dụng thành các class, các đối tượng
* Phân tích sự liên quan, sự tác động của đối tượng này đển đối tượng khác

Module google API, module Translate, module query,

Các usercase:

* Chụp ảnh
* Load ảnh từ thư viện
* Trích xuất dữ liệu từ ảnh (đưa ra groundtruth)
* Dịch văn bản
* Truy vấn dữ liệu

# **TÀI LIỆU THAM KHẢO**

[1] <http://pages.ucsd.edu/~ztu/publication/cvpr12_textdetection.pdf>

Cong Yao, Xiang Bai, Wenyu Liu, Yi Ma, and Zhuowen Tu, “Detecting texts of arbitrary orientations in natural images,” in CVPR’12, 2012, pp. 1083–1090.

[2] <https://arxiv.org/pdf/1606.09002.pdf>

C. Yao, X. Bai, N. Sang, X. Zhou, S. Zhou, and Z. Cao. Scene text detection via holistic, multi-channel prediction. CoRR, abs/1606.09002, 2016.

[3]<https://www.cv-foundation.org/openaccess/content_cvpr_2014/papers/Kang_Orientation_Robust_Text_2014_CVPR_paper.pdf>

X. C. Yin, X. Yin, K. Huang, and H. Hao. Robust text detection in natural scene images. IEEE Trans. on PAMI, 36(5):970–983, 2014.

[4] <https://arxiv.org/pdf/1703.06520.pdf>

[5] <https://arxiv.org/pdf/1604.04018.pdf>

Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu, and X. Bai. Multioriented text detection with fully convolutional networks. In Proc. of CVPR, 2015.

[6] <https://arxiv.org/pdf/1802.08948.pdf>

[7] <https://research.ijcaonline.org/etc2016/number2/etc2016263.pdf>

L. Kang, Y. Li, and D. Doermann. Orientation robust text line detection in natural images. In Proc. of CVPR, 2014.

[8] <http://cmp.felk.cvut.cz/~cernyad2/TextCaptchaPdf/Detecting%20Text%20in%20Natural%20Scenes%20with%20Stroke%20Width%20Transform.pdf>

B. Epshtein, E. Ofek, and Y. Wexler. Detecting text in natural scenes with stroke width transform. In Proc. of CVPR, 2010.

[9] <https://research.ijcaonline.org/etc2016/number2/etc2016263.pdf>

X. Yin, W. Pei, J. Zhang, and H. Hao. Multi-orientation scene text detection with adaptive clustering. PAMI, 37(9):1930–1937, 2015

[10] <https://arxiv.org/pdf/1704.03155.pdf>

X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and

J. Liang. East: An efficient and accurate scene text detector. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017

[11] <https://arxiv.org/pdf/1606.06539.pdf>

[12] <http://discovery.ucl.ac.uk/1569458/1/TPAMI-2016-08-0656-R2.pdf>

[13] <https://arxiv.org/pdf/1610.04057.pdf>

[14] <https://arxiv.org/pdf/1505.04925.pdf>

[15] S. Xie and Z. Tu. Holistically-nested edge detection. In Proc. of ICCV, 2015.

<https://www.cv-foundation.org/openaccess/content_iccv_2015/papers/Xie_Holistically-Nested_Edge_Detection_ICCV_2015_paper.pdf>

[16] <https://arxiv.org/pdf/1609.08144.pdf> Google Neural Machine Translation

[17] "BLEU," [Online]. Available: <https://en.wikipedia.org/wiki/BLEU>. [Accessed

2018].

[18] "METEOR," [Online]. Available: <https://en.wikipedia.org/wiki/METEOR>.

[Accessed 2018].