ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN KHOA
KHOA HỌC MÁY TÍNH

BÁO CÁO ĐỒ ÁN
XỬ LÝ ẢNH VÀ ỨNG DỤNG
CS406.O11.KHCL

ĐỀ TÀI: Phát Hiện Và Nhận Dạng Chữ Tiếng Việt Trong Ảnh Ngoại Cảnh

GIẢNG VIÊN HƯỚNG DẪN: **TS. MAI TIẾN DŨNG**

SINH VIÊN THỰC HIỆN: **BÙI QUỐC THỊNH – 20520934**
BÙI VIẾT ĐẠT - 20521162

MỤC LỤC

1

Giới thiệu môn học

1. Khái niệm về Trí tuệ nhân tạo

Al là trí thông minh được thể hiện bởi máy móc, không giống như trí thông minh tự nhiên được hiển thị bởi con người và động vật, liên quan đến ý thức và cảm xúc.

Học máy là nghiên cứu các thuật toán máy tính cải tiến tự động thông qua kinh nghiệm và dữ liệu. Nó được xem như một phần của trí tuệ nhân tạo.

Học sâu là một phần của họ các phương pháp học máy rộng hơn dựa trên mạng nơ-ron nhân tạo với học đại diện. Việc học có thể được giám sát, bán giám sát hoặc không giám sát.

1.1.1 Khái niệm về thị giác máy tính

Thị giác máy tính (tiếng Anh: Computer Vision) là một lĩnh vực bao gồm các phương pháp thu nhận, xử lý ảnh kỹ thuật số, phân tích và nhận dạng các hình ảnh và, nói chung là dữ liệu đa chiều từ thế giới thực để cho ra các thông tin số hoặc biểu tượng, ví dụ trong các dạng quyết định.

1.1.2 Khái niệm về xử lý hình ảnh

Xử lý hình ảnh là việc sử dụng máy tính kỹ thuật số để xử lý hình ảnh kỹ thuật số thông qua một thuật toán. Từ đó, chúng ta có thể nâng cao các bức ảnh hoặc trích xuất thông tin hữu ích từ chúng.

2

Giới thiệu về đề tài

Trong thời đại số hóa này, nhu cầu trích xuất thông tin văn bản từ các nguồn khác nhau đã tăng lên rất nhiều so với trước đây. May mắn thay, những tiến bộ gần đây trong Thị giác máy tính cho phép chúng ta đạt được những bước tiến lớn trong việc giảm bớt gánh nặng và khó khăn trong việc phát hiện văn bản cũng như phân tích và hiểu tài liệu. Trong Thị giác máy tính, phương pháp chuyển đổi văn bản có trong hình ảnh hoặc tài liệu được quét sang định dạng mà máy có thể đọc được và sau đó có thể được chỉnh sửa, tìm kiếm và có thể được sử dụng để xử lý thêm được gọi là Nhận dạng ký tự quang học (OCR).

Nhận dạng ký tự quang học (OCR) đã được phát triển để chuyển đổi các tài liệu dựa trên văn bản thành các tài liệu kỹ thuật số, ứng dụng của chúng đã tăng vọt trong khoảng thời gian gần đây. Theo các kịch bản và phương hướng nhận dạng khác nhau, các công cụ OCR có thể được chia thành các công cụ OCR chung khác nhau.

2.1 Ứng dụng nhận dạng chữ trong ảnh ngoại cảnh trong đời sống và mục đích

Truy xuất thông tin và nhập dữ liệu tự động - OCR đóng vai trò rất quan trọng đối với nhiều công ty và tổ chức có hàng nghìn tài liệu cần xử lý, phân tích và chuyển đổi hoặc số hoá để thực hiện các hoạt động hàng ngày.

Ví dụ: Trong các loại thông tin ngân hàng như chi tiết tài khoản, số tiền từ tấm séc hoặc chi phiếu có thể dễ dàng được trích xuất bằng OCR. Tương tự, tại các sân bay, trong khi kiểm tra thông tin hộ chiếu thì thay vì nhập tay thì có thể được trích xuất bằng OCR. Ngoài ra, các ứng dụng fintech như Momo, ZaloPay hoặc ứng dụng Internet Banking của các ngân hàng khi yêu cầu cung cấp Căn cước công dân hoặc Chứng minh nhân dân thì cũng truy xuất thông tin bằng OCR. Các ví dụ khác còn có truy xuất thông tin bằng OCR từ biên lai, hóa đơn, biểu mẫu, báo cáo, hợp đồng, v.v.

Nhận dạng biển số xe - OCR cũng có thể được sử dụng để nhận dạng biển đăng ký xe, sau đó có thể được sử dụng để theo dõi xe, thu phí, gửi xe, v.v.

Xe tự lái - OCR cũng có thể được sử dụng để xây dựng các mô hình cho xe tự lái. Nó có thể giúp nhận ra các biển báo giao thông. Nếu không có điều này, xe tự lái sẽ gây rủi ro cho cả người đi bộ và các phương tiện khác trên đường.

Trong bài báo cáo này, chúng ta sẽ thảo luận và triển khai các thuật toán học sâu được sử dụng trong OCR.

2.2 Phát hiện chữ trong ảnh ngoại cảnh

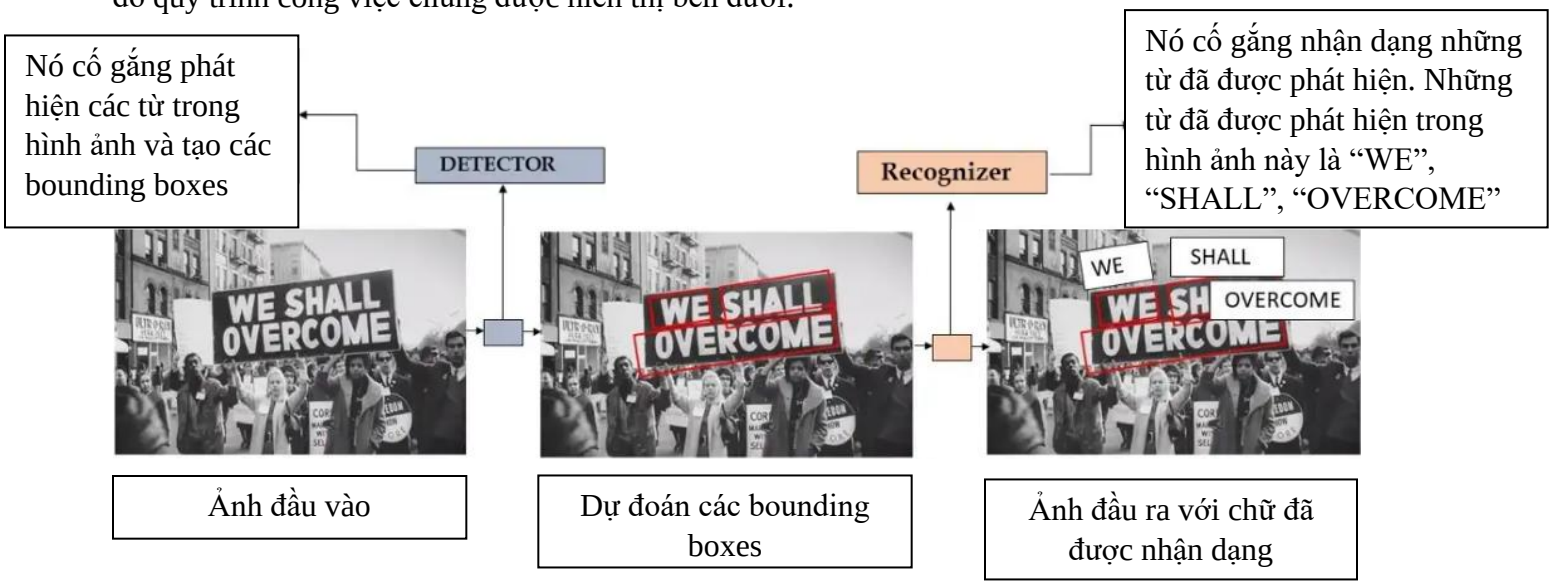
Phát hiện chữ trong ảnh ngoại cảnh (Scene Text Detection) là nhiệm vụ phát hiện các vùng văn bản trong nền phức tạp và gắn nhãn cho chúng bằng các hộp giới hạn (bounding boxes).

2.3 Nhận dạng chữ trong ảnh ngoại cảnh

Nhận dạng chữ trong ảnh ngoại cảnh là nhận dạng văn bản trong các hộp giới hạn (bounding boxes) sau khi đã phát hiện chúng.

Như chúng ta đã quen thuộc với các ứng dụng phát hiện và nhận dạng văn bản khác nhau. Trong báo cáo này, việc phát hiện và nhận dạng văn bản từ ảnh ngoại cảnh sẽ được nhấn mạnh cụ thể.

Vì vậy, trong trường hợp của chúng tôi, chúng tôi sử dụng bất kỳ hình ảnh hoặc cảnh tự nhiên nào (không đặc biệt là tài liệu, giấy phép hoặc số xe) và đối với một hình ảnh/cảnh nhất định, chúng tôi muốn bản địa hóa ký tự/từ/câu trong hình ảnh bằng hộp giới hạn (bounding boxes). Sau đó, chúng tôi muốn nhận dạng các văn bản đã được bản địa hóa, có thể thuộc bất kỳ ngôn ngữ nào. Sơ đồ quy trình công việc chung được hiển thị bên dưới:



3

Các kỹ thuật phát hiện và nhận dạng chữ trong ảnh ngoại cảnh

3.1 Phát hiện chữ trong ảnh ngoại cảnh bằng SAST

Trong thời đại ngày nay, nhờ sự gia tăng của các mạng nơ-ron sâu, nhiều phương pháp dựa trên mạng nơ-ron tích chập (CNN) đã được đề xuất để phát hiện văn bản trong ảnh ngoại cảnh.

Tuy nhiên, phát hiện văn bản có một số vấn đề do sự thay đổi đáng kể về kích thước, tỷ lệ khung hình, hướng, ngôn ngữ, hình dạng tùy ý và thậm chí cả nền phức tạp.

Đây là lý do tại sao tôi chọn SAST (Single-Shot Arbitrarily Shaped Text Detector) hay còn gọi là trình phát hiện văn bản có hình dạng tùy ý một lần dựa trên ngữ cảnh học tập đa tác vụ. Đây là là một mô hình hiệu quả trong việc dự đoán văn bản có hình dạng tùy ý tuy nhiên các vùng chữ nhỏ đang là vấn đề với nó.

Model	Backbone	Precision	Recall	Hmean
EAST	ResNet50_vd	88.71%	81.36%	84.88%
EAST	MobileNetV3	78.20%	79.10%	78.65%
DB	ResNet50_vd	86.41%	78.72%	82.38%
DB	MobileNetV3	77.29%	73.08%	75.12%
SAST	ResNet50_vd	91.39%	83.77%	87.42%
PSE	ResNet50_vd	85.81%	79.53%	82.55%
PSE	MobileNetV3	82.20%	70.48%	75.89%
DB++	ResNet50	90.89%	82.66%	86.58%

Cơ sở lý thuyết của mô hình SAST gồm 3 phần chính:

- Mạng gốc: ResNet50 + Feature Pyramid Network (FPN) để sản xuất ra bối cảnh từ ngữ – nâng cao sự thể hiện và trích xuất ra đặc trưng.
- Các nhánh đa nhiệm: Bản đồ TCL, TVO, TCO, TBO được dự đoán cho từng vùng văn bản.

- Phần xử lý hậu kỳ: Phân đoạn văn bản theo point-to-quad, liên kết các điểm thành hình tứ giác để tạo hộp giới hạn (bounding boxes) hoàn chỉnh cuối cùng.

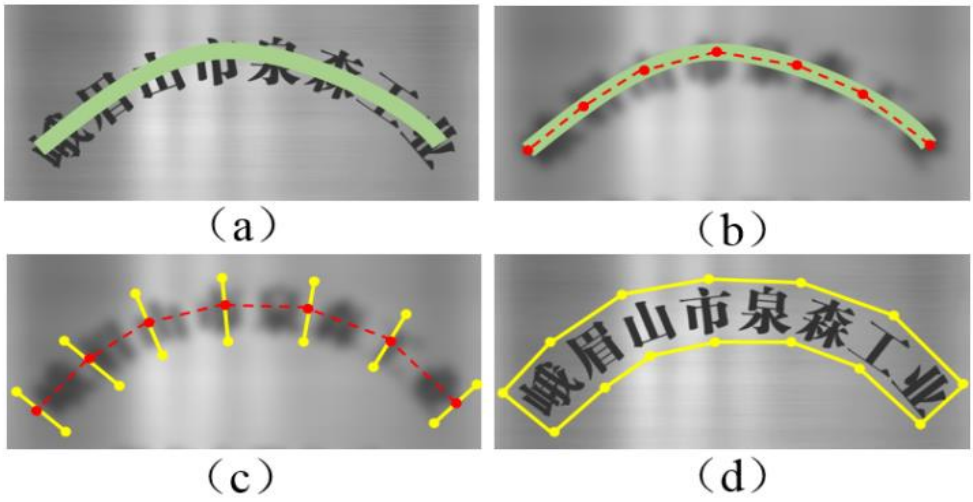


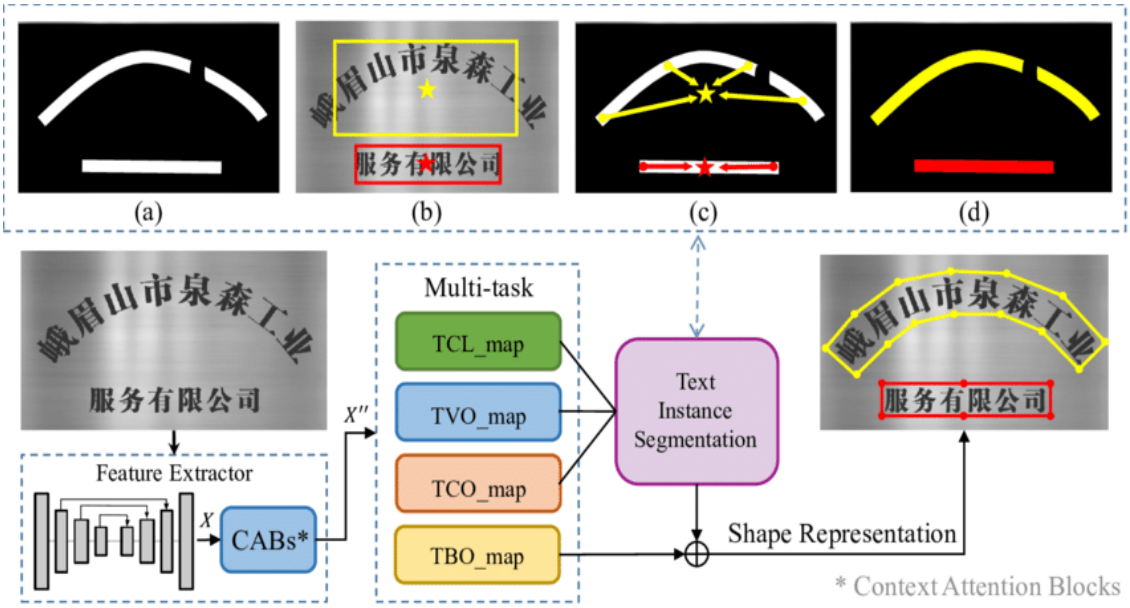
Figure 2: Arbitrary Shape Representation: a) The text line in TCL map; b) Sample adaptive number of points in the line; c) Calculate the corresponding border point pairs with TBO map; d) Link all the border points as final representation.

TCL: the center line of text. Chức năng là tìm văn bản trong đường viền.

TCO: text center offset. Chức năng là xác định điểm giữa của dòng văn bản và pixels của TCL.

TVO: text vertex offset. Chức năng là xác định số lượng điểm thích nghi mẫu trong dòng văn bản sau khi đã tìm được chúng gồm bốn đỉnh của hộp giới hạn (bounding box).

TBO: text border offset. Chức năng là xác định điểm giới hạn như upper point, lower point để dự đoán ra vùng của văn bản.



Quy trình của phương pháp đề xuất:

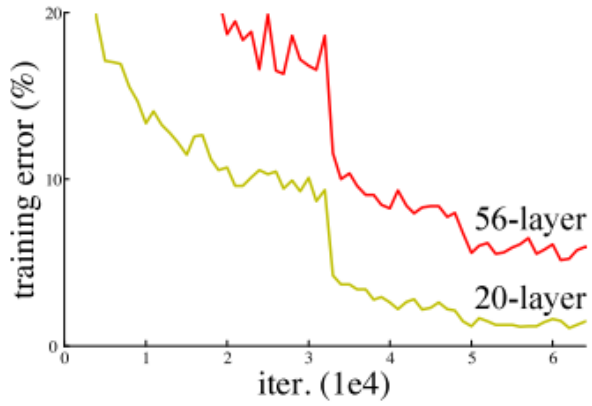
- *Trích xuất đặc trưng* từ hình ảnh đầu vào đi qua ResNet50 + FPN và tính toán bản đồ TCL, TBO, TCO, TVO dưới dạng bài toán đa nhiệm.
- *Đạt được khả năng phân đoạn phiên bản (Achieve instance segmentation)* bằng Mô-đun phân đoạn phiên bản văn bản (Text Instance Segmentation Module) và nối liền với TBO để cho ra cơ chế gán point-to-quad.
- *Khôi phục biểu diễn đa giác (Restore polygonal representation)* của các trường hợp văn bản có hình dạng tùy ý.

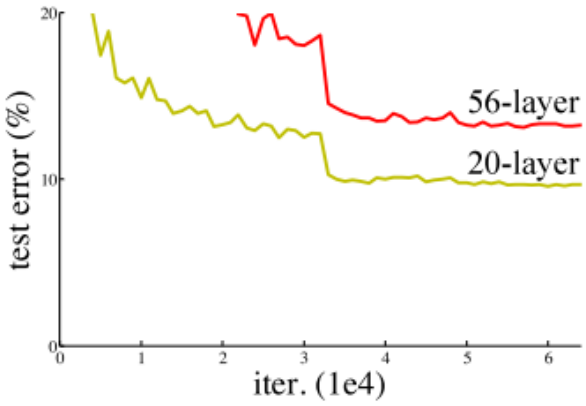
3.1.1 Giới thiệu về mạng backbone ResNet50

Mạng ResNet (R) là một mạng CNN được thiết kế để làm việc với hàng trăm lớp. Một vấn đề xảy ra khi xây dựng mạng CNN với nhiều lớp chập sẽ xảy ra hiện tượng độ dốc biến mất (vanishing gradient descent) dẫn tới quá trình học tập không tốt.

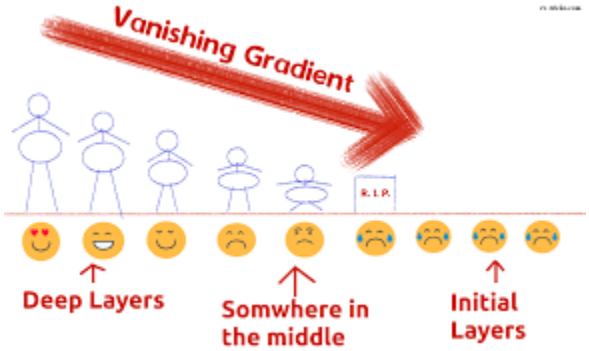
Hiện tượng độ dốc biến mất (vanishing gradient descent):

Trước hết thì thuật toán lan truyền ngược (Backpropagation Algorithm) là một kỹ thuật thường được sử dụng trong quá trình huấn luyện. Ý tưởng chung của thuật toán là sẽ đi từ lớp đầu ra (output layer) đến lớp đầu vào (input layer) và tính toán gradient của cost function tương ứng cho từng tham số (weight) của mạng. Gradient descent sau đó được sử dụng để cập nhật các tham số (parameters) đó.





Toàn bộ quá trình trên sẽ được lặp đi lặp lại cho tới khi mà các tham số của mạng được hội tụ. Thông thường chúng ta sẽ có một siêu tham số (hyperparameter) (số Epoch - số lần mà huấn luyện



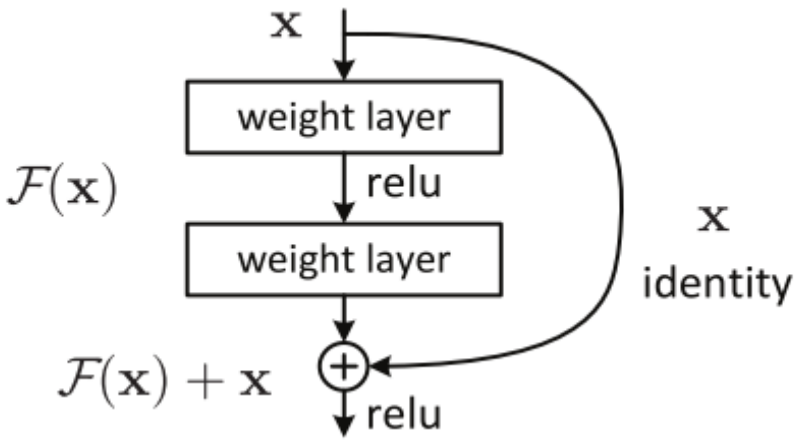
set được duyệt qua một lần và weights được cập nhật) định nghĩa cho số lượng vòng lặp để thực hiện quá trình này. Nếu số lượng vòng lặp quá nhỏ thì ta gặp phải trường hợp mạng có thể sẽ không cho ra kết quả tốt và ngược lại thời gian huấn luyện sẽ lâu nếu số lượng vòng lặp quá lớn.

Tuy nhiên, trong thực tế Gradient thường sẽ có giá trị nhỏ dần khi đi xuống các lớp thấp hơn. Dẫn đến kết quả là các cập nhật thực hiện bởi Gradient Descent không làm thay đổi nhiều weights của các lớp đó và làm chúng không thể hội tụ và mạng sẽ không thu được kết quả tốt. Hiện tượng như vậy gọi là vanishing gradient descent.

=> Mạng ResNet ra đời cũng giải quyết vấn đề đó.

Kiến trúc mạng ResNet:

Cho nên giải pháp mà ResNet đưa ra là sử dụng kết nối "tắt" đồng nhất để xuyên qua một hay nhiều lớp. Một khối như vậy được gọi là một Residual Block, như trong hình sau:



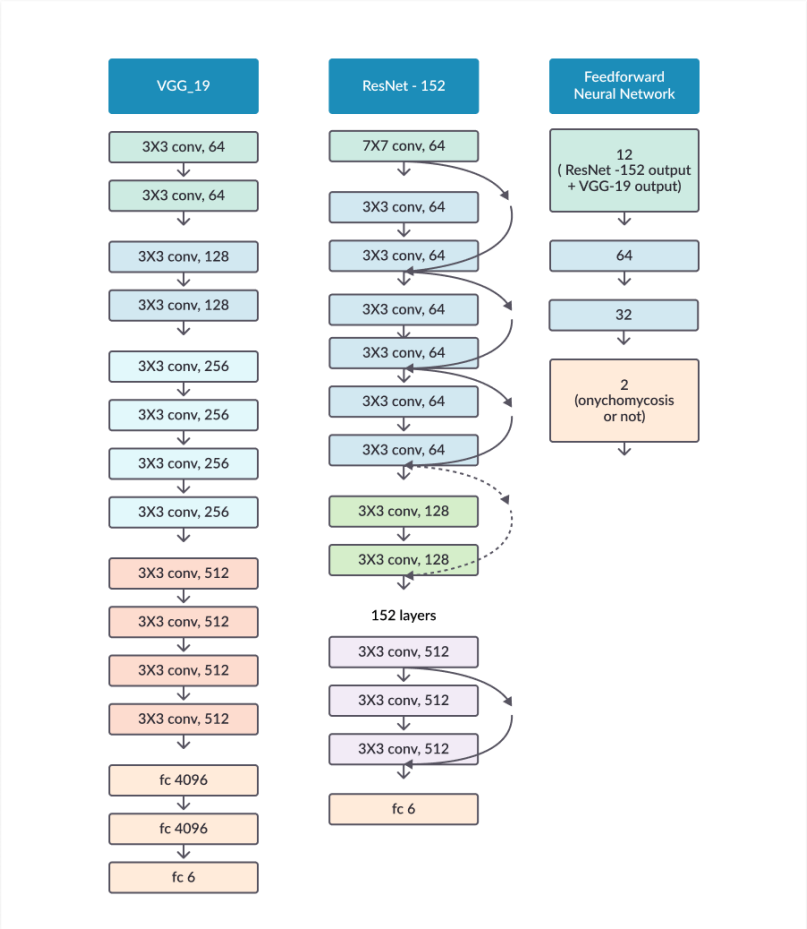
ResNet gần như tương tự với các mạng gồm có convolution, pooling, activation và fully connected layer. Ảnh bên trên hiển thị khối dư được sử dụng trong mạng. Xuất hiện một mũi tên cong xuất phát từ đầu và kết thúc tại cuối khối dư. Hay nói cách khác là sẽ bổ sung đầu vào X vào đầu ra của lớp, hay chính là phép cộng mà ta thấy trong hình minh họa, việc này sẽ chống lại việc đạo hàm bằng 0, do vẫn còn cộng thêm X. Với H(x) là giá trị dự đoán, F(x) là giá trị thật (nhãn), chúng ta muốn H(x) bằng hoặc xấp xỉ F(x). Việc F(x) có được từ x như sau:

X->weight1-> ReLU -> weight2

Giá trị H(x) có được bằng cách:

F(x) + x -> ReLU

Như chúng ta đã biết việc tăng số lượng các lớp trong mạng làm giảm độ chính xác, nhưng lại muốn có một kiến trúc mạng sâu hơn có thể hoạt động tốt.



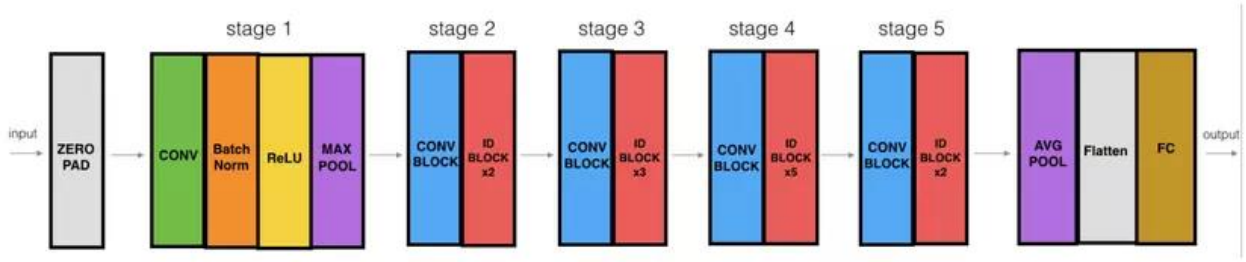
Hình 1. VGG-19 là một mô hình CNN sử dụng kernel 3x3 trên toàn bộ mạng, VGG-19 cũng đã giành được ILSVRC năm 2014.

Hình 2. ResNet sử dụng các kết nối tắt (kết nối trực tiếp đầu vào của lớp (n) với (n+x) được hiển thị dạng mũi tên cong. Qua mô hình nó chứng minh được có thể cải thiện hiệu suất trong quá trình huấn luyện mô hình khi mô hình có hơn 20 lớp.

Hình 3. Tổng cộng có 12 đầu ra từ ResNet-152 và VGG-19 đã được sử dụng làm đầu vào cho mạng có 2 lớp hidden. Đầu ra cuối cùng được tính toán thông qua hai lớp ẩn (hidden). Việc xếp chồng các lớp sẽ không làm giảm hiệu suất mạng.Với kiến trúc này các lớp phía trên có được thông tin trực tiếp hơn từ các lớp dưới nên sẽ điều chỉnh trọng số hiệu quả hơn.

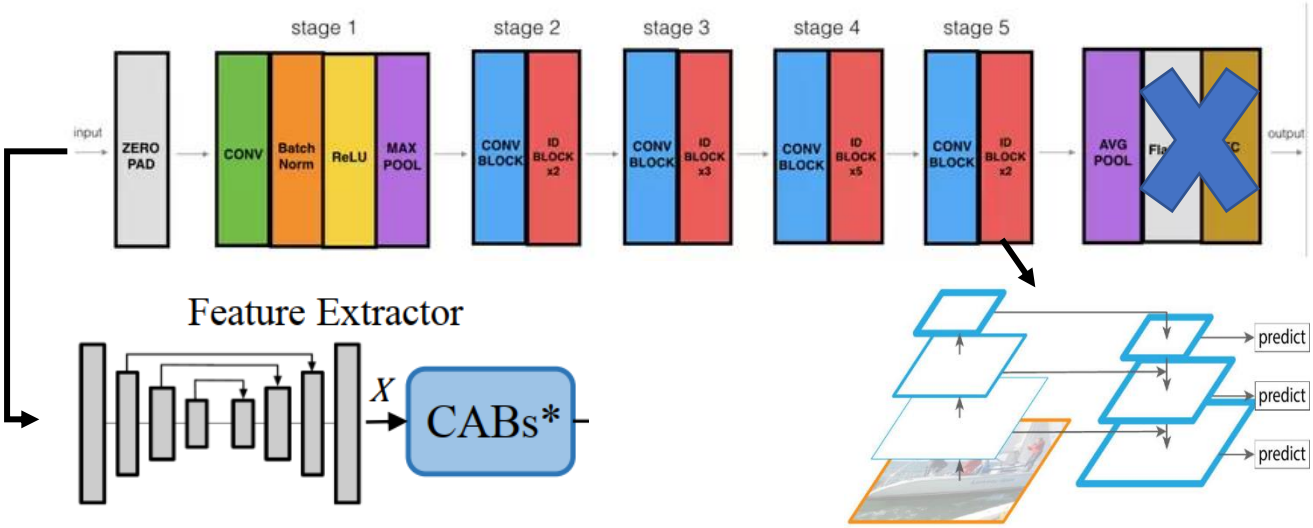
Xây dựng mạng ResNet50:

Hình dưới đây mô tả chi tiết kiến trúc mạng nơ ron ResNet:



"ID BLOCK" trong hình trên là viết tắt của từ Identity block và ID BLOCK x3 nghĩa là có 3 khối Identity block chồng lên nhau. Nội dung hình trên như sau:

- Zero-padding: Input với (3,3).
- Stage 1: Tích chập (Conv1) với 64 filters với shape (7,7), sử dụng stride (2,2). BatchNorm, MaxPooling (3,3).
- Stage 2: Convolutiontal block sử dụng 3 filter với size 64x64x256, f=3, s=1. Có 2 Identity blocks với filter size 64x64x256, f=3.
- Stage 3: Convolutional sử dụng 3 filter size 128x128x512, f=3, s=2. Có 3 Identity blocks với filter size 128x128x512, f=3.
- Stage 4: Convolutional sử dụng 3 filter size 256x256x1024, f=3, s=2. Có 5 Identity blocks với filter size 256x256x1024, f=3.
- Stage 5: Convolutional sử dụng 3 filter size 512x512x2048, f=3, s=2. Có 2 Identity blocks với filter size 512x512x2048, f=3.
- The 2D Average Pooling: sử dụng với kích thước (2,2).
- The Flatten.
- Fully Connected (Dense): sử dụng softmax activation.



FPN

Trong mạng này thì tôi sẽ thiết kế bỏ lớp fully connected đi và nhiều mức độ khác nhau của feature map của ResNet50 được hợp 3 lần trong mạng FPN.

Thì feature map X sau khi đi qua lớp ResNet xong thì bằng ¼ ảnh đầu vào.

3.1.2 Giới thiệu về mạng FPN

Feature Pyramid Network, hay FPN, là một trình trích xuất đặc trưng lấy một hình ảnh tỷ lệ đơn có kích thước tùy ý làm đầu vào và xuất ra các bản đồ đặc trưng có kích thước tương ứng ở nhiều cấp độ, theo kiểu tích chập hoàn toàn. Quá trình này độc lập với kiến trúc tích chập backbone. Do đó, nó hoạt động như một giải pháp chung để xây dựng các kim tự tháp đặc trưng bên trong các mạng tích chập sâu để sử dụng trong các tác vụ như phát hiện đối tượng.

Cấu trúc của kim tự tháp bao gồm hướng bottom up và top down.

Bottom up là tính toán chuyển tiếp của backbone ConvNet, tính toán hệ thống phân cấp đặc trưng bao gồm các bản đồ đặc trưng ở một số tỷ lệ với bước chia tỷ lệ là 2. Đối với đặc trưng kim tự tháp, một cấp kim tự tháp được xác định cho từng giai đoạn. Đầu ra của lớp cuối cùng của mỗi giai đoạn được sử dụng làm bộ tham chiếu của bản đồ đặc trưng. Đối với ResNets, chúng tôi sử dụng đầu ra kích hoạt đặc trưng theo khối còn lại cuối cùng của mỗi giai đoạn. Hạn chế là khi đi càng tiến lên cao thì các feature maps và pixels bị thu nhỏ lại nên nhận diện các ký tự nhỏ sẽ không được chính xác nên cần phải kết hợp với top down để dự đoán được rõ ràng.

Top down tạo ảo giác cho các tính năng có độ phân giải cao hơn bằng cách lấy mẫu các bản đồ đặc trưng thô hơn về mặt không gian nhưng mạnh hơn về mặt ngữ nghĩa từ các cấp độ kim tự tháp cao hơn. Các đặc trưng này sau đó được tăng cường với các đặc trưng từ lộ trình bottom up thông qua các kết nối bên. Mỗi kết nối bên hợp nhất các bản đồ đặc trưng có cùng kích thước không gian từ bottom up và top down. Bản đồ đặc trưng bottom up có ngữ nghĩa cấp thấp hơn, nhưng các kích hoạt của nó được bản địa hóa chính xác hơn vì nó được lấy mẫu phụ ít lần hơn.

3.1.3 Giới thiệu về Context Attention Block (CAB)

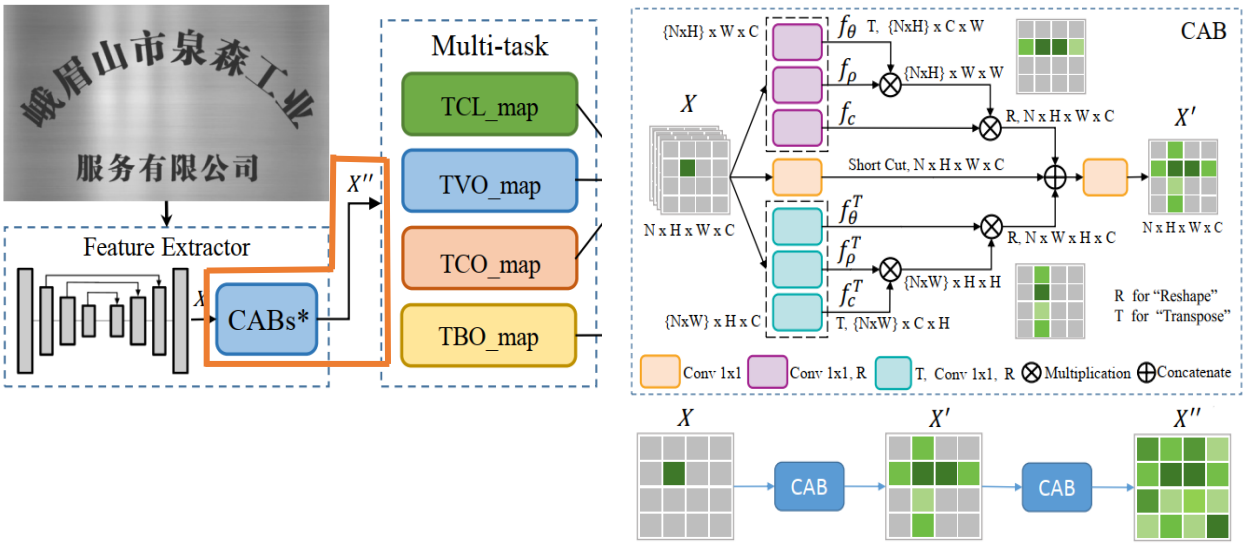


Figure 4: Context Attention Blocks: a single CAB module aggregates pixel-wise contextual information both horizontally and vertically, and long-range dependencies from all pixels can be captured by serially connecting two CABs.

Từ ảnh trích xuất đặc trưng thì sẽ cho ra kết quả là X".

Đầu tiên thì CAB sử dụng cơ chế self-attention mechanism để tổng hợp thông tin theo ngữ cảnh, để tăng cường tính miêu tả của đặc trưng.

Bản đồ đặc trưng X là đầu ra của ResNet50 với kích thước N x H x W x C.

N (Number of image): số ảnh

H (Height): chiều cao

W (Width): chiều rộng

C (Number of channel): số kênh

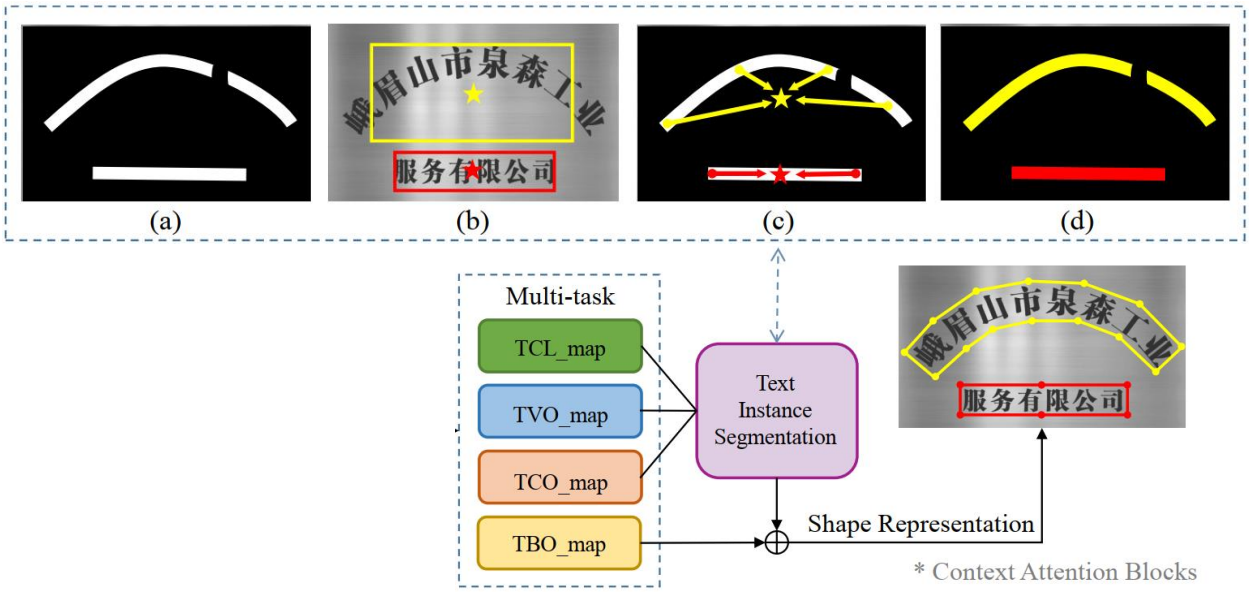
Để thu thập các ngữ cảnh (context) theo chiều ngang thì đầu tiên để cho ra kết quả {NxH} x W x C thì sẽ dung 3 lớp tích chập để thu được $f\theta$, $f\rho$ và fc. Sau đó sẽ reshape {NxH} x W x C rồi với $f\rho$ thì có kích thước bức hình là {NxH} x W x W. Sau khi tích chập mỗi lớp thì đi qua hàm sigmoid. Cuối cùng, để cải thiện ngữ cảnh (context) theo chiều ngang thì sẽ nhân với fc thì sẽ được kết quả N x H x W x W. Cuối cùng thì reshape lại sẽ có N x H x W x C.

Tương tự thì để thu được ngữ cảnh (context) theo chiều dọc thì ta sẽ chuyển vị các chiều của $f\theta$, $f\rho$ và fc sau khi được tích chập. Sau đó, chuyển vị kích cỡ ma trận đầu vào thành {NxW} x C x H. Đi qua element wise, nhân với chuyển vị của $f\rho$ thì có {NxW} x H x H. Tiếp tục nhân với chuyển vị của $f\theta$. Cứ mỗi lần đi qua element wise thì sẽ đi qua hàm sigmoid. Cuối cùng, reshape lại về N x W x H x C.

Sau khi có kết quả theo chiều ngang lẫn dọc thì sẽ nối lại thì sẽ được bức hình phát hiện được chữ chiều ngang lẫn chiều dọc.

Sau khi đi qua thêm một lần CAB nữa thì sẽ thu được ảnh phát hiện được các ký tự trong ảnh.

3.1.4 Giới thiệu khối Multitask



Với mỗi text instance thì chúng ta sẽ tính toán các điểm trung tâm và bốn góc của tứ giác.

TCL_map: the center line of text là một phiên bản thu nhỏ của vùng văn bản (text region), dùng để chỉ ra một kênh segmentation. Để xác định ở đây là một cái hình ảnh segmentation thì khi mà nó segment được vùng màu trắng ở hình (a) thì đây là một văn bản, còn vùng đen là không phải văn bản (non-text).

TVO_map: text vertex offset là các điểm của bốn đỉnh của hộp giới hạn (bounding boxes)

TCO_map: text context offset là điểm giữa và pixels của TCL.

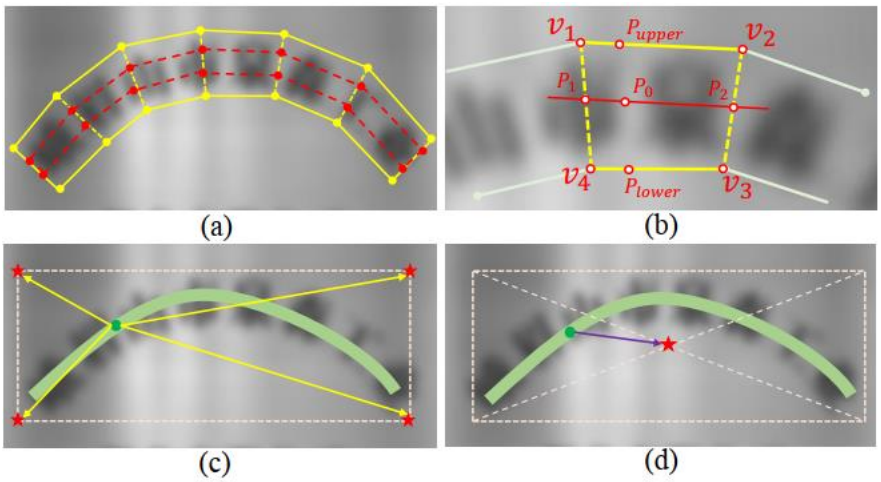


Figure 5: Label Generation: (a) Text center region of a curved text is annotated in red; (b) The generation of TBO map; The four vertices (red stars in c) and center point(red star in d) of bounding box, to which the TVO and TCO map refer.

- Đầu tiên, các đỉnh v1, v2, v3, v4 được xác định khá dễ. Số lượng TCO và TVO là 8 và 2 bởi vì nó sẽ gồm 2 kênh. Vì TCO chúng ta có 1 hộp giới hạn (bounding box) thì có 1 điểm giữa, vậy thì 2 kênh sẽ có 2 điểm và một TVO thì sẽ có 4 cạnh của một tứ giác, vậy thì 2 kênh thì sẽ được 8.

TBO_map: là điểm upper và lower point thì khi có 2 điểm nhân với 2 kênh thì được 4 điểm. Để sản xuất ra các điểm TBO thì cần 2 bước. Bước thứ nhất thì sẽ xác định các cặp điểm đối nhau của upper side và lower side của một tứ giác. Nhờ tính hệ số góc trung bình giữa đường thẳng từ đường v1 đến đường v2 và tính hệ số góc trung bình giữa đường thẳng từ đường v3 đến đường v4 thì sẽ thu được điểm P0. Khi mà đường thẳng đi qua P0 thì P0 sẽ dễ dàng được xác định dựa trên TCL thì cũng dễ dàng xác định P1 và P2. Khi đó các cặp điểm tương ứng {Pupper, Plower} cho P0 có thể được xác định từ:

$$\frac{P_0 - P_1}{P_2 - P_1} = \frac{P_{upper} - V_1}{V_2 - V_1} = \frac{P_{lower} - V_4}{V_3 - V_4}$$

Công thức quen thuộc khi dùng định lý Talet.

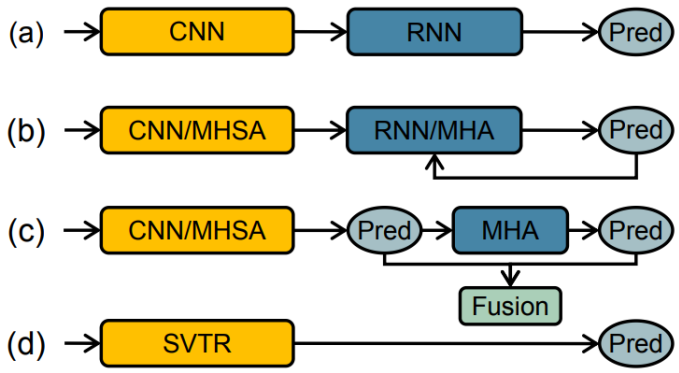
Hàm loss:

$$L_{total} = \lambda_1 L_{tcl} + \lambda_2 L_{tco} + \lambda_3 L_{tvo} + \lambda_4 L_{tbo},$$

3.2 Nhận dạng chữ trong ảnh ngoại cảnh bằng SVTR

Văn bản có thông tin ngữ nghĩa phong phú, đã được sử dụng trong nhiều ứng dụng dựa trên thị giác máy tính như lái xe tự động, phiên dịch du lịch, truy xuất sản phẩm, hỗ trợ người khiếm thị, v.v.

Mặc dù nhận dạng theo trình tự đã tạo ra một số điểm đáng chú ý, nhưng nhận dạng văn bản vẫn là một thách thức lớn do các biến thể đáng kể của văn bản cảnh về màu sắc, phông chữ, bố cục không gian và thậm chí không thể kiểm soát được nền.



Hình trên là một số phương pháp tiếp cận đối với bài toán scene text recognition.

Có thể nói rằng nhận diện văn bản trong ảnh ngoại cảnh có thể được xem xét như một ánh xạ từ hình ảnh đến chuỗi ký tự. Thông thường bộ nhận diện có kiến trúc gồm hai khối như đa phần các kiến trúc trên hình minh hoạ, một mô hình hình ảnh để trích xuất đặc trưng và một mô hình chuỗi để chuyển đổi văn bản. Ví dụ, các mô hình dựa trên CNN-RNN ban đầu sử dụng CNN để trích xuất đặc trưng. Sau đó đặc trưng được biến đôi thành một chuỗi và được mô hình hoá bằng BiLSTM và hàm loss CTC để có được dự đoán như cách tiếp cận (a) trên hình minh hoạ. Chúng nổi bật với hiệu suất và hiện đang được áp dụng trong các ứng phát hiện và nhận diện chữ trong thực tế và thương mại. Tuy nhiên, quá trình biến đổi nhạy cảm đối với các nhiễu văn bản như biến dạng, che phủ, v.v., từ đó giới hiện đi hiệu suất của chúng.

Sau đó, các phương pháp tự học dựa trên encoder-decoder trở nên phổ biến, những phương pháp này chuyển đổi quá trình nhận diện bằng cách encoder trích xuất các đặc trưng từ hình ảnh và sau đó decoder dự đoán chuỗi văn bản theo từng ký tự thành một thủ tục giải mã lặp lại (b). Kết quả là, độ chính xác được cải thiện do thông tin ngữ cảnh được xem xét. Tuy nhiên, tốc độ inference chậm do quá trình chuyển thành từng ký tự cũng như kiến trúc hai giai đoạn (encoder + decoder) có thể phức tạp để quản lý và tối ưu hóa.

Quy trình này đã được mở rộng lên thành phương pháp Vision-language based framework (c) khi mà phương pháp này tích hợp các kiến thức ngôn ngữ vào mô hình và thực hiện dự đoán song song. Các mô hình dựa theo phương pháp này cho thấy độ chính xác cao hơn khi kiến thức ngôn ngữ có thể giúp mô hình hiểu ngữ cảnh và ý nghĩa văn bản, cũng như thực hiện dự đoán song song các ký tự cho phép cải thiện tốc độ. Tuy vậy, việc tích hợp kiến thức ngôn ngữ có thể làm tăng độ

phức tạp của mô hình và làm cho quá trình nhận dạng phức tạp và ít tính tường minh hơn. Ngoài ra, khuyết điểm khác là dung lượng mô hình lớn: Các mô hình vision-language cầu nguồn lực tính toán đáng kể để đạt được hiệu suất tốt, hạn chế hiệu quả của chúng.

Gần đây, đã có những nỗ lực tập trung vào việc phát triển kiến trúc đơn giản hóa để tăng tốc độ. Ví dụ, sử dụng mô hình đào tạo phức tạp nhưng đơn giản cho inference. Giải pháp dựa trên CRNN-RNN đã được xem xét khi nó sử dụng cơ chế attention và graph neural để tổng hợp các đặc trưng tuần tự tương ứng với cùng một ký tự. Trong quá trình inference, nhánh mô hình attention đã được loại bỏ để cân bằng giữa độ chính xác và tốc độ. PREN2D là phương pháp đơn giản hóa hơn nữa việc nhận dạng bằng cách tổng hợp trực tiếp và giải mã đồng thời các đặc trưng của ký tự phụ, dẫn đến xử lý nhanh hơn với độ chính xác thấp hơn một chút so với CRNN-RNN. VisionLAN với cách tiếp cận sử dụng kỹ thuật "học tập theo từng ký tự" để huấn luyện mô hình nhận diện bằng kiến thức ngôn ngữ. Khi inference, mô hình xử lý nhanh hơn, hy sinh một số độ chính xác so với giai đoạn huấn luyện. Các mô hình CNN và ViT có sẵn: Các phương pháp này chỉ đơn giản sử dụng các mô hình pretrained CNN hoặc ViT làm công cụ trích xuất đặc trưng mà không cần bất kỳ bước decoding bổ sung nào. Chúng rất hiệu quả nhưng độ chính xác của chúng thường thấp hơn các mô hình phức tạp hơn.

Trong nhận diện chữ trong ảnh ngoại cảnh có hai đặc trưng quan trọng không thể không nhắc đến:

- Intra-character local patterns (Các mẫu cục bộ trong ký tự): Đây là các đặc điểm giống như nét vẽ chi tiết giúp phân biệt các ký tự riêng lẻ trong văn bản. Hãy tưởng tượng những đường cong và góc cạnh tinh tế của các chữ cái khác nhau giúp phân biệt chúng như thế nào. Những đặc trưng này rất quan trọng để nhận dạng chính xác, đặc biệt là để nhận diện các dấu của các ký tự.
- Inter-character long-term dependence: Điều này đề cập đến kiến thức tương tự ngôn ngữ giúp nắm bắt cách các ký tự liên hệ với nhau và hình thành từ. Nó giống như việc hiểu “ngữ pháp” của văn bản trong cảnh đó. Mô hình hóa sự phụ thuộc này là rất quan trọng để nhận dạng văn bản phức tạp hoặc mơ hồ. Căn bản nói một cách dễ hiểu chính là chúng sẽ xem xét các mối liên hệ giữa các từ.

Điểm yếu lớn nhất của các phương pháp trên chính là chúng không có khả năng nắm bắt hiệu quả đồng thời cả hai loại đối tượng:

- CNN backbones: Mặc dù giỏi nắm bắt các mối tương quan cục bộ (Intra-character local patterns) trong các mảng hình ảnh nhỏ, nhưng lại gặp khó khăn trong việc mô hình hóa mối quan hệ lâu dài (Inter-character long-term dependence) giữa các ký tự trong toàn bộ khung cảnh.
- General-purpose Transformer backbones: Chúng vượt trội trong việc mô hình hóa các phần phụ thuộc toàn cầu (Inter-character long-term dependence) nhưng thường bỏ qua các mẫu cục bộ (Intra-character local patterns) chi tiết trong các ký tự riêng lẻ.

Chính vì thế, chúng ta cần một mô hình có thể hội tủ đủ hai yếu tố trên khi có thể nắm bắt tốt cả hai đặc trưng được đề ra. Từ đó, sự lựa chọn tối ưu để giải quyết vấn đề này chính là SVTR: Scene Text Recognition with a Single Visual Model khi mô hình này có thể làm tốt trong việc nắm bắt

các đặc trưng để cho ra độ chính xác khi nhận dạng cao cũng như thời gian inference nhanh chóng hơn so với các mô hình theo phương pháp trước đây.

Đây là lý do tôi chọn SVTR (Single Visual Text Recognition)

Model	Backbone	Avg Accuracy
Rosetta	Resnet34_vd	79.11%
Rosetta	MobileNetV3	75.80%
CRNN	Resnet34_vd	81.04%
CRNN	MobileNetV3	77.95%
StarNet	Resnet34_vd	82.85%
StarNet	MobileNetV3	79.28%
RARE	Resnet34_vd	83.98%
RARE	MobileNetV3	81.76%

SRN	Resnet50_vd_fpn	86.31%
NRTR	NRTR_MTB	84.21%
SAR	Resnet31	87.20%
SEED	Aster_Resnet	85.35%
SVTR	SVTR-Tiny	89.25%
ViTSTR	ViTSTR	79.82%
ABINet	Resnet45	90.75%
VisionLAN	Resnet45	90.30%
SPIN	ResNet32	90.00%
RobustScanner	ResNet31	87.77%
RFL	ResNetRFL	88.63%
ParseQ	VIT	91.24%
CPPD	SVTR-Base	93.8%

SVTR là một framework có thể đào tạo và huấn luyện từ đầu đến cuối bao gồm năm phần chính:

- Patch-wise image tokenization: Hình ảnh được chia thành các patches 2D nhỏ (các thành phần ký tự), cho phép nắm bắt tốt hơn các đặc điểm cục bộ.
- Mạng backbone Transformers tuỳ chỉnh theo văn bản: Một mạng backbone 3 tần giảm chiều cao ảnh với các hoạt động trộn (mixing), hợp nhất (merging) và kết hợp (combining).
- Các khối local mixing và global mixing: Các khối này nắm bắt cả các đặc trưng nét cục bộ (Intra-character local patterns) trong các patches và sự phụ thuộc lâu dài giữa các ký tự (Inter-character long-term dependence).
- Multi-grained character feature perception: Bằng cách trích xuất các đặc trưng ở các khoảng cách và tỷ lệ khác nhau, SVTR tạo thành một bản trình bày toàn diện về từng ký tự.
- Dự đoán tuyến tính đơn giản (Simple linear prediction): Mô hình dự đoán trực tiếp các ký tự từ các patches được encoded, loại bỏ việc decode chuỗi phức tạp.

Nhiều biến thể kiến trúc: SVTR cung cấp các mô hình khác nhau với công suất khác nhau (SVTR-L, SVTR-B, SVTR-S, SVTR-T) để phù hợp với các nhu cầu khác nhau.

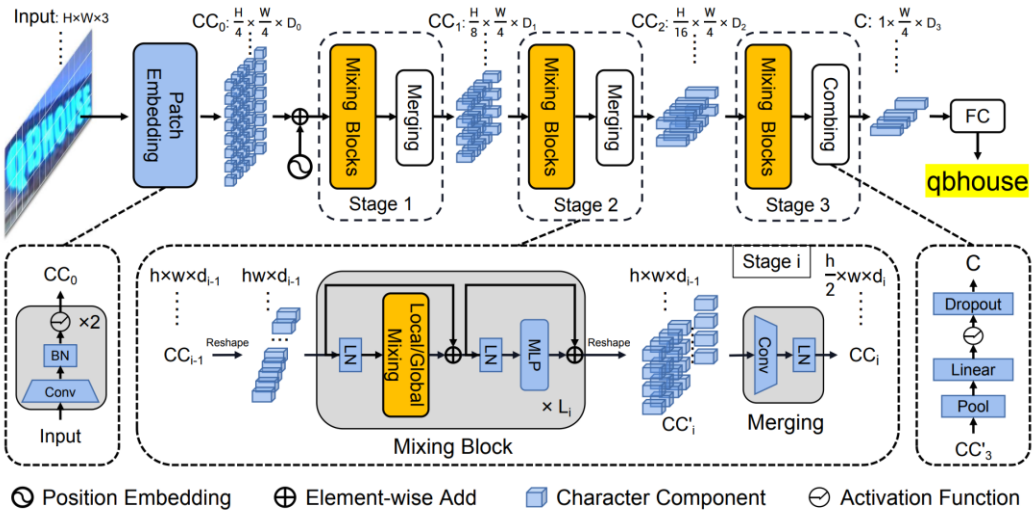


Figure 2: Overall architecture of the proposed SVTR. It is a three-stage height progressively decreased network. In each stage, a series of mixing blocks are carried out and followed by a merging or combining operation. At last, the recognition is conducted by a linear prediction.

3.2.1 Giới thiệu bao quát kiến trúc

Đầu vào và patch embedding:

- Mô hình sẽ nhận vào một tấm ảnh chứa văn bản với kích thước H×W×3.
- Sau đó sẽ chia ảnh thành các lớp patches chồng nhau với kích thước $H/4 \times W/4 \times D_0$, mỗi phần tử đóng vai trò như các kí tự trong các bài toán xử lý ngôn ngữ hay còn gọi là "character components".
- Kiến trúc Pathc Embedding bao gồm 2 lớp convolution có kernel 3×3, stride 2 theo sau đó là lớp Batch Normalization. Theo nghiên cứu tác giả, các chiến thuật Patch Embedding khác nhau sẽ ảnh hưởng đến hiệu quả của mô hình.

Embedding	IC13	IC15
Linear	92.5	72.0
Overlap	93.0	73.9
Ours	**93.5**	**74.8**

So sánh ảnh hưởng của Patch Embedding tới hiệu quả mô hình

Như ảnh trên ta có thể thấy, Patch Embedding được đề xuất trong mô hình giúp out-perform 0.75% và 2.8% so với các phương pháp còn lại.

Ba tầng trích xuất đặc trưng:

- Kiến trúc có ba tần với chiều cao giảm dần (H/4, H/8, H/16) để trích xuất các đặc trưng với các tỷ lệ khác nhau.
- Mỗi tầng sẽ bao gồm:
    - Khối mixing: Chúng nắm bắt cả các tính năng giống như nét cục bộ (Intra-character local patterns) trong các patches và sự phụ thuộc toàn cầu (Inter-character long-

term dependence) giữa các patches gần đó. Có hai loại: khối local mixing và khối global mixing.

- Merging hoặc combining: Điều này kết hợp các đặc trưng từ các patches/tỷ lệ khác nhau để hiểu rộng hơn.

Biểu diễn đặc trưng kí tự multi-grained: Quá trình nhiều giai đoạn sẽ tạo ra feature map "C" có kích thước 1 × W/4 × D_3 biểu thị các đặc trưng ký tự multi-grained (các nét cục bộ và bối cảnh tổng thể).

Dự đoán tuyến tính song song với tính năng khử trùng lặp: Các ký tự được dự đoán trực tiếp từ feature map C mà không cần decode trình tự phức tạp. Tính năng chống trùng lặp sẽ loại bỏ các dự đoán ký tự dư thừa.

3.2.2 Progressive Overlapping Patch Embedding

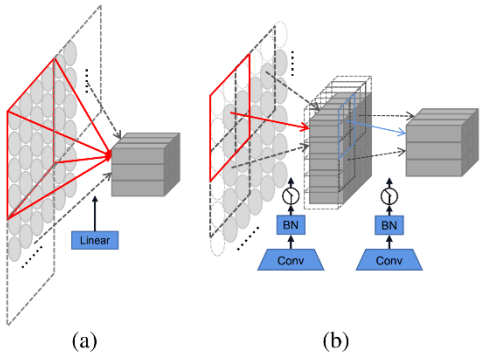


Figure 3: (a) The linear projection in ViT [Dosovitskiy *et al.*, 2021].
(b) Our progressive overlapping patch embedding.

Phần patch embedding đã được giới thiệu ở kiến trúc mô hình bao quát. Tuy nhiên, ở đây chúng ta sẽ đi vào chi tiết một chút.

Có hai hướng tiếp cận với việc phân chia hình ảnh thành các patches nhỏ:

- Phép chiếu tuyến tính rời rạc (4 × 4 Disjoint Linear Projection): Chiếu hình ảnh trực tiếp lên CC_0 bằng lớp tuyến tính 4 × 4. Có thể mất thông tin không gian.
- Lớp tích chập 7 × 7 với stride 4: Trích xuất các patches lớn hơn với nhiều ngữ cảnh hơn nhưng có thể bỏ qua các chi tiết nhỏ hơn.

SVTR's Progressive Overlapping Patch Embedding:

- Hai kết cấu tích chập 3x3 liên tiếp với Stride 2 và batch normalization:
  - Sử dụng các kernel nhỏ hơn để dần dần xây dựng độ sâu của đặc trưng, bảo toàn các chi tiết tốt hơn.
  - Sử dụng batch normalization để ổn định và hội tụ nhanh hơn.
  - Tăng dần kích thước đặc trưng, có lợi cho việc hợp nhất sau này.
  - Tăng nhẹ chi phí tính toán so với phương pháp one-step.

3.2.3 Mixing Blocks

Mixing Blocks gồm có hai loại là:

- Khối local mixing
- Khối global mixing

Động lực của hai module này xuất phát từ 2 ý tưởng: Một mô hình nhận diện chữ tốt ngoài biểu diễn được mối liên hệ giữa các chữ tức là thông tin toàn cục như các phương pháp CRNN + Attention đang làm thì còn phải biểu diễn tốt tương quan giữa các chi tiết trong cùng một kí tự. Khác nhau giữa các nét chấm, phẩy cũng tạo nên khác nhau giữa các chữ cái.

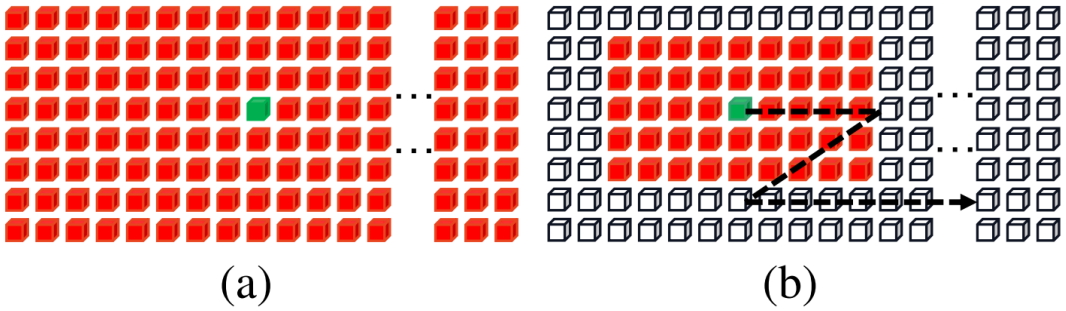


Figure 4: Illustration of (a) global mixing and (b) local mixing.

Global mixing dùng để biểu diễn quan hệ giữa các phần thử non-text và text qua đó biểu diễn phụ thuộc xa giữa các kí tự với nhau. Biểu diễn phụ thuộc xa tác giả sử dụng kiến trúc kiến trúc self-attention kết hợp với một số lớp như LayerNorm và MLP. Biểu diễn phụ thuộc xa tác giả sử dụng kiến trúc kiến trúc self-attention kết hợp với một số lớp như LayerNorm và MLP.

Local mixing dùng để biểu diễn mối quan hệ giữa các nét trong cùng một kí tự. Điều này đặc biệt ý nghĩa đối với các chữ có kí tự phức tạp như tiếng Nhật. Như hình trên bạn có thể thấy chỉ một chút khác nhau về độ dài ngắn giữa các nét cũng đã đủ tạo ra một chữ khác. Local mixing cũng sử dụng cơ chế window slide trượt trên các vùng kích thước 7×11 và tính toán mối liên hệ giữa các phần tử trong vùng cửa sổ đó. Chú ý một chút các phần tử ở đây chính là các phần tử được chia qua lớp Patch Embedding được trình bày bên trên.

Kiến trúc có thể biểu diễn tốt sự khác biệt giữa các chữ đến từng chi tiết nhỏ sẽ giúp cho quá trình phân biệt bằng CTC Loss trở nên tốt hơn sau này.

3.2.4 Merging

Kiến trúc Merging này có chức năng trích xuất đặc trưng trên nhiều tỷ lệ khác nhau để loại bỏ hiện tượng biểu diễn thừa thông tin. Để thực hiện điều này, sau mỗi lớp Mixing Blocks, tác giả sử dụng một lớp tích chập có kích thước kernel 3 × 3, bước nhảy 2 theo chiều cao và 1 theo chiều rộng. Như vậy, với một đầu vào có kích thước $h \times w \times d_{i-1}$ sẽ cho ra đầu ra có kích thước $h/2 \times w \times d_i$. Chiều cao sẽ được giảm đi một nửa tuy nhiên chiều rộng feature map sẽ được giữ nguyên giúp giảm chi phí tính toán và các lớp ở các tầng khác nhau không biểu diễn cùng một thông tin. Điều này có ý nghĩa vì các ảnh cho bài toán nhận dạng chữ có kích thước chiều rộng lớn hơn chiều cao rất nhiều.

3.2.5 Combining and Prediction

Combining:

- Pooling: Giảm kích thước chiều cao xuống 1, cô đọng các thành phần ký tự theo chiều dọc.
- Lớp Fully-Connected: Nén thêm các đặc trưng, tạo chuỗi tính năng 1D.
- Hàm kích hoạt Non-Linear và Dropout: Giới thiệu tính phi tuyến tính và ngăn chặn việc overfitting.
- Mục đích: Tránh tích chập trên các embeddings nhỏ có chiều cao 2, đảm bảo xử lý hiệu quả.

Dự đoán tuyến tính song song (Parallel Linear Prediction):

- Trình phân loại tuyến tính với N nút: Dự đoán trực tiếp các ký tự từ feature map kết hợp.
- Đầu ra: Chuỗi bản ghi có kích thước W/4, với:
    - Ký tự trùng lặp cho các thành phần có cùng ký tự.
    - Biểu tượng trống cho các thành phần không phải văn bản.
- Ngưng tụ (Condensation): Trình tự được tự động cô đọng đến kết quả văn bản cuối cùng.

Kiến trúc Combining được sử dụng ở tầng cuối cùng của mô hình thay thế cho kiến trúc Merging đưa kích thước chiều cao về 1 bằng lớp Poopling. Theo sau đó là lớp fully connected và activation. Việc sử dụng lớp Combining ở cuối thay vì Merging giúp tránh việc sử dụng lớp tích chập đối với các ma trận đặc trưng quá nhỏ gây mất đặc trưng ban đầu.

4

Thực nghiệm

4.1 Input và output

Input: Một hình ảnh ngoại cảnh

Output: Một hình ảnh, nếu có văn bản, sẽ xuất ra kết quả và vị trí của văn bản trên ảnh

4.2 Bộ dataset Vintext

Định dạng của dataset Vintext:

Định dạng x1, y1, x2, y2, x3, y3, x4, y4, TRANSCRIPT.

- Thư mục label: chứa các tệp chú thích cho từng hình ảnh
- Thư mục train_images – chứa 1200 ảnh từ im0001 đến im1200
- Thư mục val_images – chứa 300 ảnh từ im1201 đến 1500
- Thư mục test_images – chứa 500 ảnh từ im1501 đến im2000
- Tập tin general_dict.txt
- Tập tin vn_dictionary.txt


```
343,84,495,113,494,187,343,160,PHÒNG
510,121,641,141,641,213,510,190,KHÁM
276,195,352,200,348,342,272,331,BS.
361,174,483,195,480,356,358,342,ĐOÀN
491,198,582,210,578,368,508,359,VĂN
591,214,697,230,703,384,593,370,HÙNG
671,149,688,153,687,162,671,159,NGỌC
688,153,700,156,700,164,687,161,PHU
670,143,701,150,701,155,671,148,###
671,160,702,165,701,171,671,165,###
```

Nhìn vào tập dữ liệu, chúng ta thấy:

Thư mục label chứa từng tệp chú thích cho mỗi hình ảnh.

Mỗi dòng trong tệp chú thích chứa một hộp văn bản.

Các cặp điểm được sắp xếp ngược chiều kim đồng hồ và bắt đầu từ góc trên dưới và cách nhau bằng dấu phẩy, cuối cùng là văn bản của ô đó.

Chuyển về định dạng PaddleOCR:

Image file name **Image annotation information encoded by json.dumps”**

img_001.jpg [{“transcription”: “text”, “points”: [[310, 104], [416, 141], [418, 216], [312, 179]]}, {…}]

Các điểm sẽ là cặp (x, y) của 4 góc của hộp văn bản theo hướng ngược chiều kim đồng hồ, bắt đầu từ góc dưới cùng bên trái.

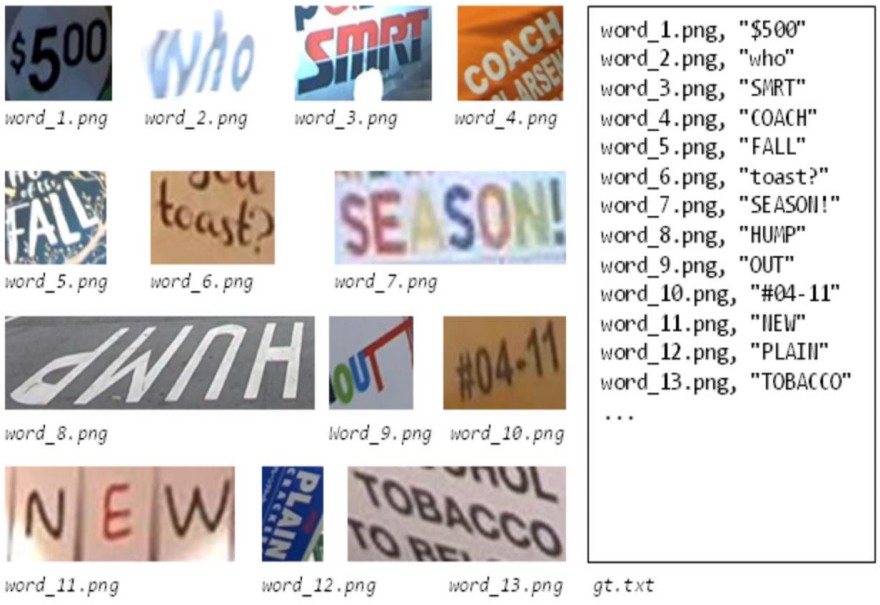
Transcription là văn bản trong hộp văn bản hiện tại. Khi nó chứa "###", điều đó có nghĩa là hộp văn bản này không hợp lệ và sẽ bỏ qua khi huấn luyện mô hình.

```
label = {}
i = i.split(',',8)
label['transcription'] = i[-1]
label['points'] = [[i[0],i[1]],[i[2],i[3]], [i[4],i[5]],[i[6],i[7]]]
text.append(label)
```

Ý tưởng để chuyển đổi sang định dạng PaddleOCR:

- Sử dụng vòng lặp for để đọc tệp chú thích của từng hình ảnh.
- Chạy một vòng lặp for đọc đến hết các dòng trong tệp chú thích để nhận các hộp văn bản.
- Dùng split(',', 8) để tách từng điểm và từng đoạn văn bản rồi lưu vào từ điển (trong lệnh split thêm tham số 8 để tránh trường hợp văn bản có dấu phẩy sẽ bị thiếu dấu phẩy trong văn bản).

Tiền xử lý cho phần nhận dạng:



(a) (b)

- Định dạng dữ liệu như sau, (a) là ảnh gốc, (b) là văn bản tương ứng với mỗi ảnh.
- Chúng ta cần cắt hình ảnh từ các hộp văn bản.
- Sử dụng khoảng cách Euclide để tính max_weight, max_height: là kích thước của phần ảnh được cắt xén chứa văn bản.
- Sử dụng hai hàm 'getPerspectiveTransform' và 'warpPerspective' trong thư viện cv2 để cắt và xoay hình ảnh theo chiều dọc.
- Sử dụng vòng lặp for để đọc từng tệp nhãn, cắt nhỏ hình ảnh chứa văn bản và lưu theo tên tương ứng (đối với văn bản không đọc được "###" chúng tôi sẽ bỏ qua).

4.3 Tăng cường dữ liệu

Chúng tôi thực hiện tăng cường dữ liệu trên tập dữ liệu huấn luyện VinText để so sánh với khi huấn luyện mô hình khi không có tăng cường dữ liệu.

Tăng cường dữ liệu sẽ được chia thành 3 phần khác nhau bao gồm:

- Geometry
- Deterioration
- Color Jitter

Geometry:

- Random rotation:
    - Chọn ngẫu nhiên góc quay trong phạm vi được xác định trước (ví dụ: -15 đến +15 độ).
    - Xoay hình ảnh theo chiều kim đồng hồ hoặc ngược chiều kim đồng hồ theo góc đó.
    - Đảm bảo trung tâm hình ảnh vẫn cố định trong quá trình xoay, ngăn ngừa sự dịch chuyển không gian không cần thiết.
- Random affine:
    - Xoay: Xoay hình ảnh trong phạm vi được chỉ định (ví dụ: -15 đến +15 độ).
    - Chia tỷ lệ: Thay đổi kích thước hình ảnh theo hệ số ngẫu nhiên giữa các giới hạn do người dùng xác định (ví dụ: 0,5 đến 2,0).
    - Cắt: Nghiêng hình ảnh theo chiều ngang hoặc chiều dọc theo một góc ngẫu nhiên (ví dụ: -45 đến +45 độ).
    - Dịch: Dịch chuyển hình ảnh theo chiều ngang và/hoặc chiều dọc theo một lượng ngẫu nhiên (ví dụ: lên tới 30% chiều rộng/chiều cao của hình ảnh).
- Random perspective:
    - Chọn hệ số biến dạng: Điều này kiểm soát cường độ biến dạng phối cảnh (các giá trị điển hình nằm trong khoảng từ 0,1 đến 0,5).
    - Xác định một hình tứ giác: Chọn ngẫu nhiên bốn điểm trong ảnh, tạo thành một hình tứ giác bị biến dạng.
    - Chiếu lên một hình tứ giác đều: Ánh xạ các điểm ảnh trong hình tứ giác bị biến dạng ban đầu lên một hình tứ giác đều (thường là hình chữ nhật).

Lợi ích của việc thay đổi geometry của ảnh:

- Nâng cao khả năng khái quát hóa mô hình cho các góc nhìn chưa được học, hướng và tỷ lệ hình ảnh đa dạng.
- Chuẩn bị mô hình cho các biến thể hình ảnh trong thế giới thực.
- Tăng cường độ bền của mô hình đối với các biến dạng và sai lệch nhỏ.
- Có thể kết hợp với các kỹ thuật tăng cường khác.

Deterioration:

- Gaussian noise:
    - Thêm các giá trị ngẫu nhiên được rút ra từ phân bố Gaussian (trung bình 0, độ lệch chuẩn 20) cho mỗi pixel.
- Motion Blur:
    - Áp dụng hiệu ứng làm mờ tuyến tính dọc theo một góc cụ thể (6 độ trong trường hợp này), trộn các pixel theo hướng đó.

- Rescaling:
    - Thay đổi kích thước hình ảnh thành 1/4 kích thước ban đầu.
- Ứng dụng ngẫu nhiên (Xác suất 0,25):
    - Áp dụng cả ba hiệu ứng cùng nhau với xác suất 25%. Trong 75% trường hợp, hình ảnh gốc không thay đổi.

Lợi ích của việc thay đổi deterioration của ảnh:

- Giới thiệu các biến thể cường độ ngẫu nhiên trên hình ảnh, mô phỏng nhiễu cảm biến hoặc điều kiện ánh sáng yếu.
- Làm mờ hình ảnh theo một hướng cụ thể, bắt chước chuyển động của máy ảnh hoặc chuyển động của vật thể.
- Thu nhỏ hình ảnh, mô phỏng các tình huống trong đó đối tượng quan tâm ở xa hơn hoặc được chụp bằng camera có độ phân giải thấp hơn.

Color Jitter:

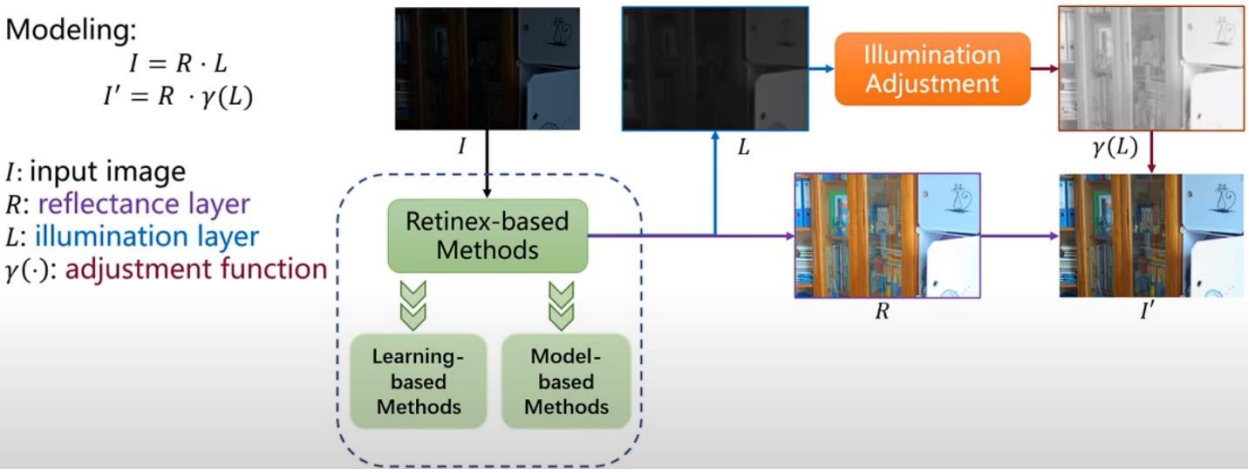
- Độ sáng: Kiểm soát độ sáng và độ tối tổng thể của hình ảnh.
- Độ tương phản: Điều chỉnh sự khác biệt giữa vùng sáng và vùng tối, tăng cường hoặc giảm độ sắc nét của cạnh.
- Độ bão hòa: Xác định cường độ của màu sắc, từ màu xám đến màu sống động.
- Hue Shift: Thay đổi ngẫu nhiên tông màu của hình ảnh, dịch chuyển các giá trị màu xung quanh bánh xe màu.
- Độ sáng, Độ tương phản, Độ bão hòa (0,5): Các giá trị này có thể biểu thị hệ số 0,5 cho các điều chỉnh ngẫu nhiên, nghĩa là thay đổi tới 50% giá trị ban đầu.
- Hue Shift (0,1): Điều này có thể hàm ý mức độ dịch chuyển màu tối đa là 10% không gian màu.
- Tần số (0,25): Chỉ định xác suất 25% áp dụng hiện tượng rung màu cho một hình ảnh nhất định.

Lợi ích của việc thay đổi deterioration của ảnh:

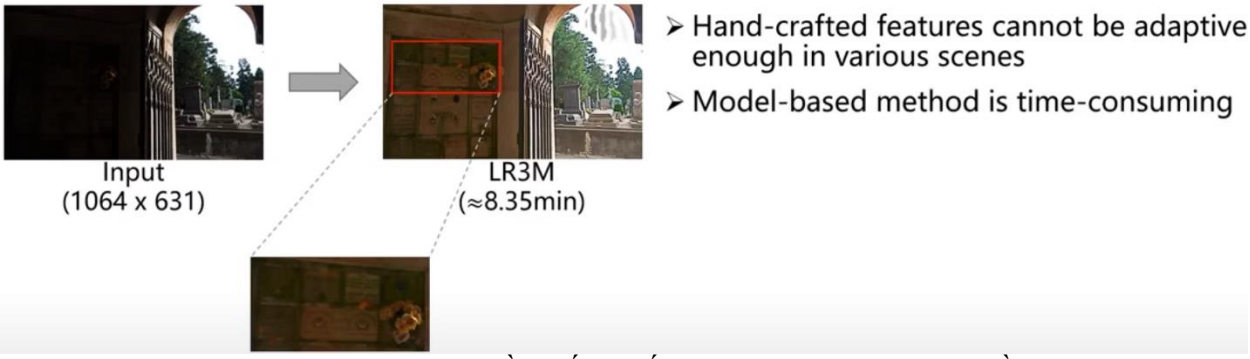
- Giảm sự phụ thuộc vào cách phối màu cụ thể.
- Nâng cao hiệu suất của mô hình trong các tình huống thực tế với độ sáng và màu sắc khác nhau.

4.3 URetinex-Net

Ở đây, nhận thấy mô hình làm không quá tốt trong môi trường tối nên chúng tôi đã đề xuất sử dụng một mô hình pretrained để tăng độ sáng của ảnh. Phương pháp chúng tôi sử dụng chính là URetinex-Net: Retinex-based Deep Unfolding Network for Low-light-Image-Enhancement.

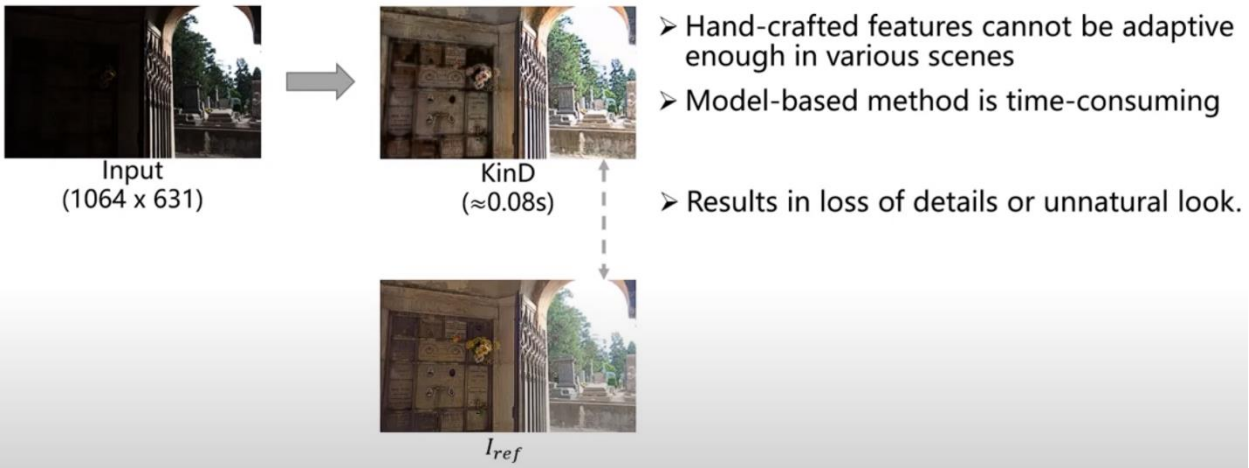


Các phương pháp Retinex-based giả định rằng hình ảnh được chia thành 2 lớp là reflectance và illumination. Bằng việc điều chỉnh lớp illumination, sẽ cho kết quả tăng cường độ sáng.



Với các phương pháp model-based truyền thống, kết quả phân tách dựa vào phần lớn hand-crafted priors và quy trình tối ưu hoá tốn thời gian.

Với phương pháp learning-based, thời gian inference nhanh nhưng thiếu tính tường minh. Thêm vào đó, điểm yếu của kết quả là sự mất chi tiết và nhìn không tự nhiên.



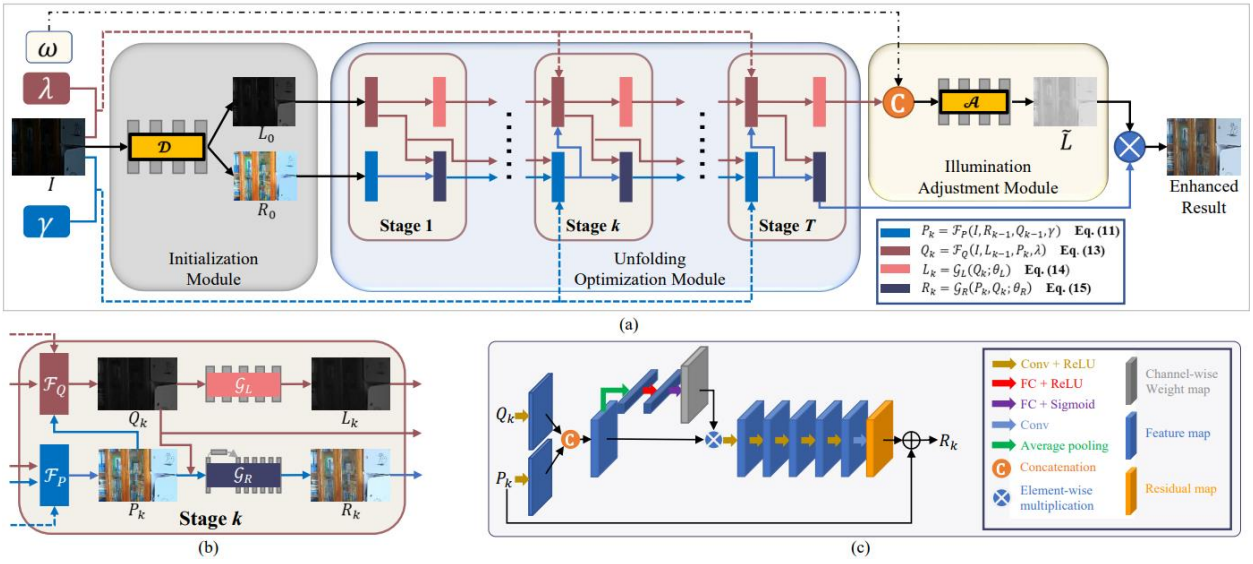
Để giải quyết tình trạng trên thì chúng tôi đề xuất một mạng học sâu mới để cải thiện hình ảnh trong điều kiện ánh sáng yếu gồm 3 module chính:

- Module khởi tạo
- Module tối ưu hoá
- Module điều chỉnh lớp illumination

Chúng tôi triển khai quy trình tối ưu hóa thành một mạng học sâu:

- Kế thừa tính linh hoạt và khả năng diễn giải từ các phương pháp model-based.
- Tận dụng khả năng mô hình mạnh mẽ của các phương pháp learning-based để phù hợp một cách thích ứng với các ưu tiên phụ thuộc vào dữ liệu.

Kiến trúc mô hình:



5

Kết quả

5.1 Kết quả SAST

```
eval model:: 100% 300/300 [13:53<00:00,  2.78s/it]
[2023/11/17 04:46:40] ppocr INFO: metric eval ***************
[2023/11/17 04:46:40] ppocr INFO: precision:0.8767871485943776
[2023/11/17 04:46:40] ppocr INFO: recall:0.7568991818055748
[2023/11/17 04:46:40] ppocr INFO: hmean:0.8124441798154214
[2023/11/17 04:46:40] ppocr INFO: fps:3.584134829447069
```

Một số ảnh ví dụ:

5.2 Kết quả SVTR

Kết quả trên tập validation:

```
eval model:: 100% 29/29 [00:46<00:00,  1.61s/it]
[2023/12/12 15:08:56] ppocr INFO: metric eval ***************
[2023/12/12 15:08:56] ppocr INFO: acc:0.7608033230459788
[2023/12/12 15:08:56] ppocr INFO: norm_edit_dis:0.8655199608777486
[2023/12/12 15:08:56] ppocr INFO: fps:567.426129415102
```

Kết quả trên tập test:

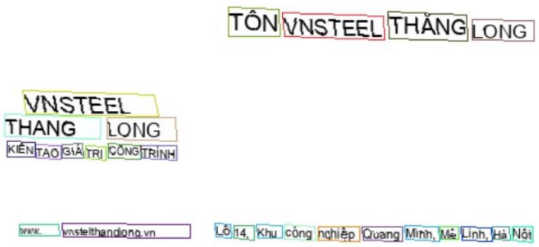
```
eval model:: 100% 40/40 [01:14<00:00,  1.85s/it]
[2023/12/14 23:49:16] ppocr INFO: metric eval ***************
[2023/12/14 23:49:16] ppocr INFO: acc:0.8295657338592448
[2023/12/14 23:49:16] ppocr INFO: norm_edit_dis:0.913554876421521
[2023/12/14 23:49:16] ppocr INFO: fps:712.3465905896311
```

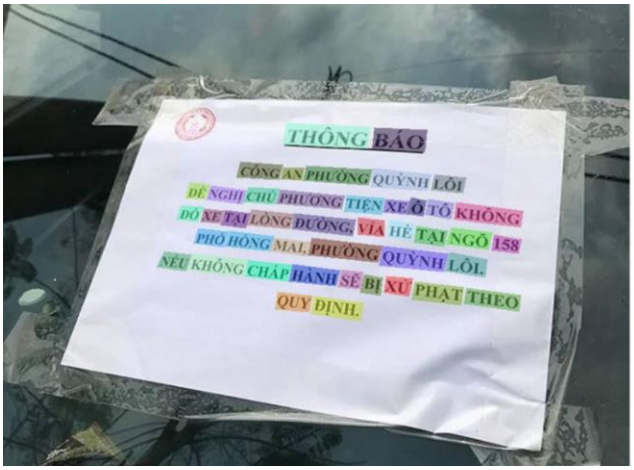
Một số ảnh minh hoạ:


TRƯỜNG TIỂU HỌC
NGUYỄN BỈNH KHIÊM

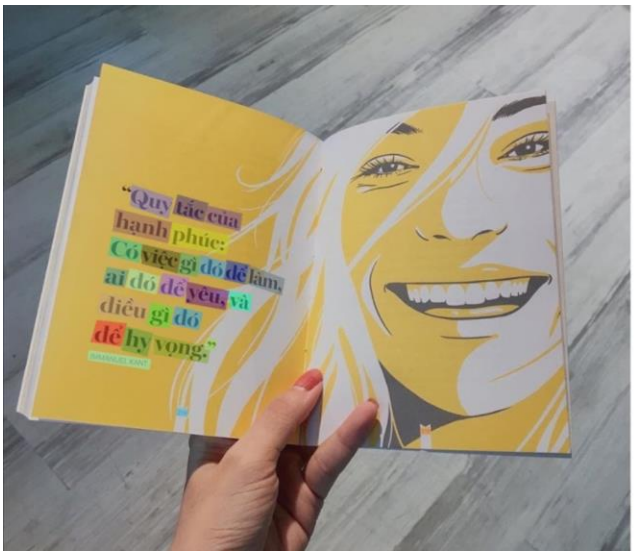

TRƯỜNG TIU HỌC
NGUYN BỈNH KHIÊM


TÔN VNSTEEL THĂNG LONG
VNSTEEL®
THANG LONG
KIẾN TẠO GIÁ TRỊ CÔNG TRÌNH
www. vnsteelthanglong.vn
Lô 14, Khu công nghiệp Quang Minh, Mê Linh, Hà Nội


TÔN VNSTEEL THĂNG LONG
VNSTEEL
THANG LONG
KIẾN TẠO GIÁ TRỊ CÔNG TRÌNH
www. vnsteelthanglong.vn
Lô 14, Khu công nghiệp Quang Minh, Mê Linh, Hà Nội

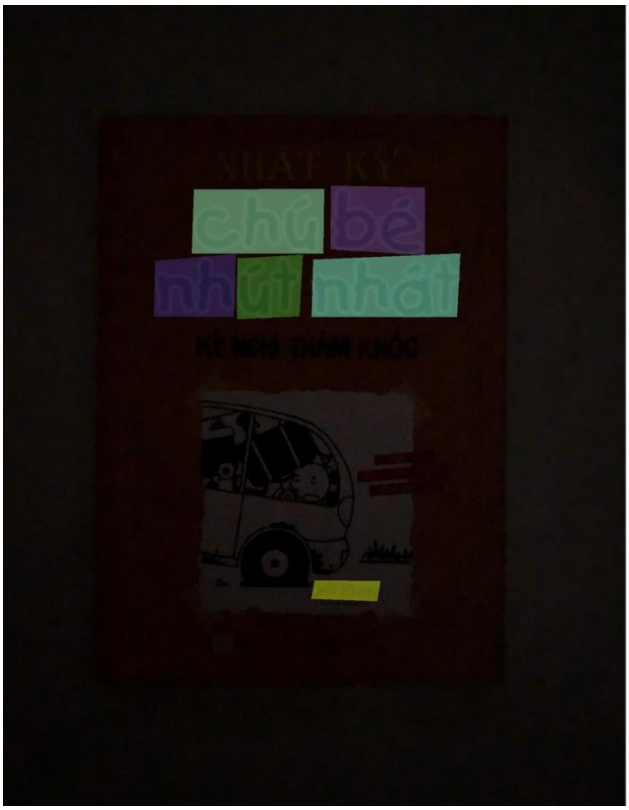

THÔNG BÁO
CÔNG AN PHƯỜNG QUỲNH LÔI
ĐỀ NGHỊ CHỦ PHƯƠNG TIỆN XE Ô TÔ KHÔNG
ĐỖ XE TẠI LÒNG ĐƯỜNG, VỈA HÈ TẠI NGÕ 158
PHỐ HỒNG MAI, PHƯỜNG QUỲNH LÔI.
NẾU KHÔNG CHẤP HÀNH SẼ BỊ XỬ PHẠT THEO
QUY ĐỊNH.

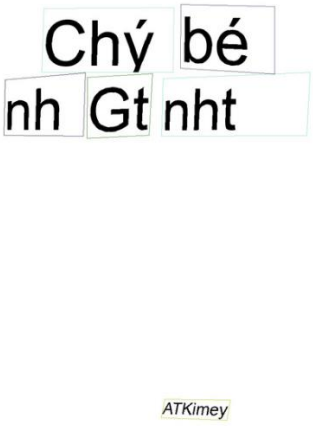

THÔNG BÁO
CÔNG AN PHƯỜNG QUỲNH LÔI
Đ NGHỊ CHỦ PHƯƠNG TIỆN XE Ô TÔ KHÔNG
Đ XE TẠI LÒNG ĐƯỜNG, VỈA HÈ TẠI NG 158
PHỐ HỒNG MAI, PHƯỜNG QUỲNH LÔI.
NẾU KHÔNG CHẤP HÀNH S BỊ XỬ PHẠT THEO
QUY ĐỊNH.


"Quy tắc của
hạnh phúc:
Có việc gì đó để làm,
ai đó để yêu, và
điều gì đó
để hy vọng."
IMMANUEL KANT


NAM DƯỢC
An Thanh


Hea Glang
ThaiNguyen
Bac Giang
Lang Son
Quong Ninh
La CAI Yen Bai
Thai Binh
Hoa Binh PhuTho
Sonla
HaiPhong
inh
H Tinh Quang Binh
Nghe An
HA NOI
Binh
Quang
DA NANG
Quang Tri Quang
Binh Thuan Kon Tham
Ninh Thuan
Duong
Dak Nong
Lan Đang
Dong
Long An
Tay Ninh
kung
SAI GON
Ten
Ber Tre
An Gaong
Tra Vinh
BacLieu
Vinh Long
CaHeu
Soc





Sau khi tăng sáng sử dụng Uretinex-Net:


NAM DƯỢC
VIÊN NGẬM
an Thanh
HỖ TRỢ GIẢM ĐAU RÁT HỌNG - KHÀN TIẾNG - HO
VIÊN NGẬM THẢO DƯỢC KHÔNG ĐƯỜNG, DÙNG ĐƯỢC CHO NGƯỜI TIỂU ĐƯỜNG, ĂN KIÊNG
DÙNG CHO NGƯỜI BỊ HO DO VIÊM HỌNG, KHÀN TIẾNG, MẤT TIẾNG DO HO KÉO DÀI


ONAMDƯỢC
HẢI THƯỢNG LÃN ÔNG
VIÊN NGM
THAnH
aN
HỖ TRỢ GIẢM ĐAU RÁT HONG KHN TIẾNG HO
VIÊN NGM THẢO DƯỢC KHÔNG ĐƯỜNG DÙNG ĐƯỢC CHO NGƯỜI TIỂU ĐƯỜNG AN KIÊNG
DÙNG CHO NGƯỜI BỊ HO DO VIÊM HỌNG KHÀN TIẾNG MT TIẾNG DO HO KÉO DÀI
THC PHM BÃO KHOC


Ha Giang
Bac Kan
Thai Nguyen
Bac Giang
Yen Bai
Hoa Binh
Phong
Nghe An
DA NANG
Binh Thuan
Dak Nong
Binh Duong
SAT GON
Long An
Ben Tre
An Giang
Tra Vinh
Hau Giang
Kien Giang
Ca Mau


Giong
Long lơn
QuangNinh
Thai
VE
Bai
Thel Binh
HOI
Dương Heal
Thành Hoa
Tint
An
Sonst
Quan
Binh
Thur
DA NANKG
Ninh
Thuan
Than
NOI
ăm Dong
Tay anh
SAT GON
Donget
nng Tai
Vinh
VinnIng

NHÀ THUỐC ĐẠT CHUN

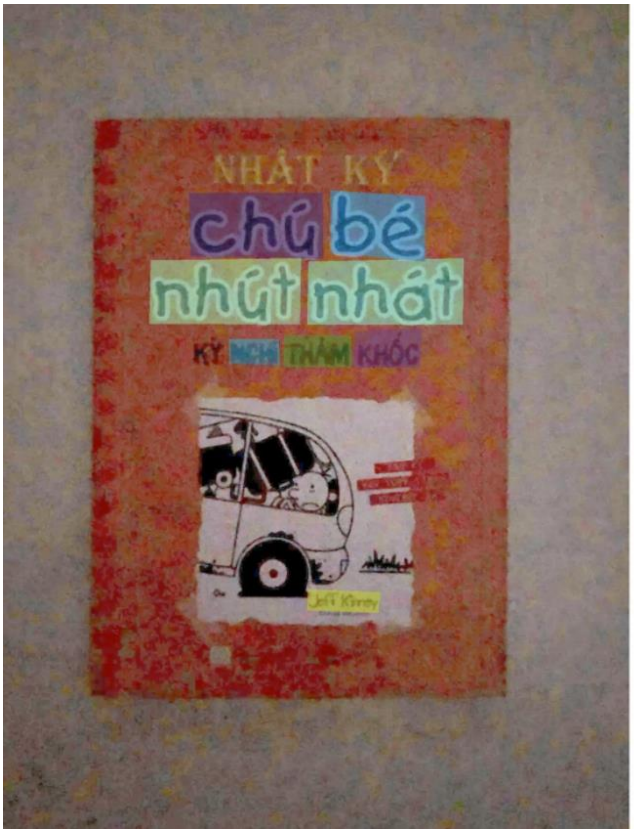
CHUYÊN BÁN GÌ L THUỐC NỘI NGOẠI NHP CÁC LOẠI

Dược Sĩ QUÂN TRỌNG TIẾN

ĐC: 389/52A LÊ VĂN KHƯƠNG KP.5 P. HIỆP THÀNH 12 TP. HCM

GPS: 41.8029877

ĐT: 0964 096 868



Chý bé

nhýt nht

NCHÍ THM KHÓC

HTKirey

Tài liệu tham khảo

[1] A Single-Shot Arbitrarily-Shaped Text Detector based on Context Attended Multi-Task Learning by Wang, Pengfei and Zhang, Chengquan and Qi, Fei and Huang, Zuming and En, Mengyi and Han, Junyu and Liu, Jingtuo and Ding, Errui and Shi, Guangming ACM MM, 2019

[2] SVTR: Scene Text Recognition with a Single Visual Model by Yongkun Du and Zhineng Chen and Caiyan Jia Xiaoting Yin and Tianlun Zheng and Chenxia Li and Yuning Du and Yu-Gang Jiang IJCAI, 2022

[3] [AI-Challenge 2021] Nhận dang chữ tiếng Việt trong ảnh ngoai cảnh – Cao Hung Van

[4] PaddleOCR

[5] Dictionary-guided Scene Text Recognition by Nguyen Nguyen, Thu Nguyen, Vinh Tran, Minh-Triet Tran, Thanh Duc Ngo, Thien Huu Nguyen, Minh Hoai CVPR 2021

[6] URetinex-Net: Retinex-based Deep Unfolding Network for Low-light Image Enhancement by Wu, Wenhui and Weng, Jian and Zhang, Pingping and Wang, Xu and Yang, Wenhan and Jiang, Jianmin CVPR, 2022

Bảng Phân Công

Tên	MSSV	Task
Bùi Quốc Thịnh	20520934	Nhận dạng & Tăng cường dữ liệu
Bùi Viết Đạt	20521162	Phát hiện & Tăng sáng