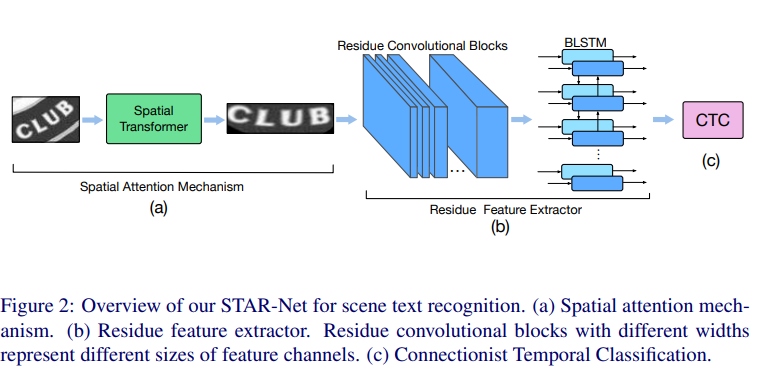
**STAR-Net: A SpaTial Attention Residue Network for Scene Text Recognition**

**1 Introduction**

End-to-end tranable STAR-Net là một novel deep neural network được tích hợp spatial attention mechanism và residue learning cho nhận điện văn bản bối cảnh. Theo kết quả thực nghiệm cho thấy nó vượt trội hơn sao với các phương pháp hiện đại khác đối với những hình ảnh văn bảng có ràng buộc lỏng lẻo và bị biến dạng đáng kể.

**2 Methodology**

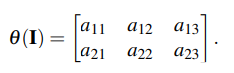
Có 3 thành phần chính của STAR-Net đó là biến đổi không gian (spatial transformer), residue feature extractor và connectionist temporal classification.



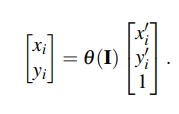
**2.1 Spatial Transformer**

Spatial Transformer chịu trách nhiệm giới thiệu cơ chế chú ý không gian(spatial attention mechanism) bằng cách chuyển các vùng văn bản giới hạn lỏng lẻo sang vùng văn bản chặt chẽ hơn và được chỉnh sửa. Spatial Transformer bao gồm 3 thành phần localisation network, sampler and interpolator. Localisation network dùng để chỉnh định những biến dạng thông qua hình ảnh văn bản nguyên mẫu và xuất ra thông số biến đổi tương ứng. Dựa vào những thông số đó, sampler sẽ định vị các điểm lấy mẫu trên input image, xác định vùng văn bản để khử nhiễu. Cuối cùng, interpolator tạo ra output image bằng cách nội suy các giá trị cường độ của 4 pixel gần nhất đối với mỗi điểm lấy mẫu.

**Localisation Network** The Localisation Network sử dụng ảnh trắng đen nguyên mẫu với rộng W và cao H như là input và trực tiếp xuất ra thông số của một affine transformation.



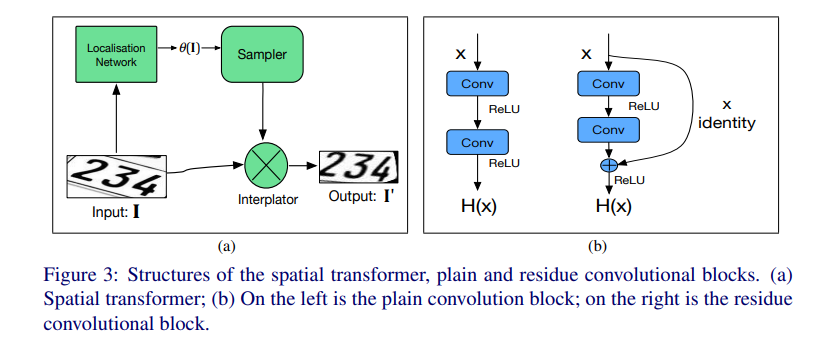
Sampler The Sampler nhắm vào vị trí của từng điểm lấy mẫu trên ảnh đầu vào cho mỗi pixel cho ảnh đầu ra. Mỗi điểm lấy mẫu ( ) trên ảnh đầu vào cho mỗi pixel ( ) được tính toán bằng hàm θ(I).



**Interpolation** The Interpolation tạo ra một giá trị cường độ cho mỗi pixel ( ) trên input image từ giá trị cường độ của mỗi 4 pixel trên input image mà gần với điểm lấy mẫu nhất.

**2.2 Residue Feature Extractor**

Để khai thác triệt để tiềm năng của các lớp và xây dựng một bộ mã hóa mạnh mẽ và sâu sắc, chúng tôi sử dụng các khối phức hợp (convolutional blocks) để trích xuất đặc điểm từ ảnh, và sử dụng Long Short-Term Memory (LSTM). Bản đồ mã hóa có miền không gian là Cs × Hs ×Ws trong đó Cs, Hs, Ws đại diện cho kênh, cao và rộng. Chúng tôi chuyển không gian 3 chiều trên thành các bản đồ đặc điểm tuần tự bằng cách cắt dọc theo chiều rộng của nó thành không gian 2 chiều, với mỗi không gian bao gồm Cs × Hs, và map mỗi lát cắt thành vector , trong đó t = [1,2,...,Ws]



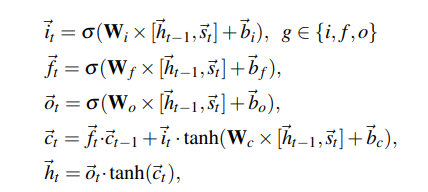
**Residue Convolutional Block** The residue block sử dụng trong bộ mã hóa bao gồm 2 lớp liên kết, 2 ReLU (Rectified Linear Units) và một kết nối tắt giữa input và output của lớp liên kết thứ 2. Với x đại diện cho input đến block, H(x) là hàm phức hợp mà chúng tôi muốn ước lượng. Một khối cấu trúc dùng để tìm các thông số phù hợp (Wpb) cho lớp liên kết đến ước lượng (Hx), tức là

H(x) = PB(x,Wpb),

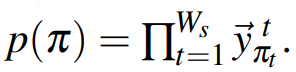
trong khi Residue Convolutional Block, với một kết nối tắt đặc biệt, mục đích để tìm tham số phù hợp (Wrb) các lớp chập để tính hàm dư H(x) - x, tức là

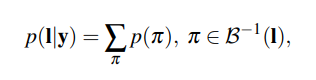
H(x) = RB(x,Wrb) +x

**Long Short-Term Memory** Long Short-Term Memory (LSTM) là một loại lớp lặp lại có khả năng học các phụ thuộc dài của đặc điểm đầu vào tuần tự. Thành phần đơn giản của LSTM là memory block. Mỗi memory block bao gồm một ô nhớ c, cổng đầu vào i, forget gate f and an output gate o, tương ứng, có thể biểu điển theo công thức như sau



**2.3 Connectionist Temporal Classification**

L đại diện cho tập 36-class( 26 ký tự và 10 chữ số) ký tự không phân biệt chữ thường là L’ = L∪ {blank}. Bước cuối cùng của mã hóa, chúng tôi sử dụng một softmax layer để xuất ra một bản đồ xác suất y, điều kiện dựa trên các tính năng tuần tự st. Mỗi là một xác suất phân phối L’ và là xác suất trên nhãn m tại thời gian t. Xác suất của mỗi tuần tự π với độ dài Ws được định nghĩa bởi  Với nhãn I và độ dài T (T ≤ Ws ), xác suất được tính toán như sau



trong đó đại diện cho tập các tuần tự độ dài Ws. Hàm Octc của CTC được định nghĩa là tổng của các negative log likelihood

