viettel

PHÁT HIỆN HÀNH VI VƯỢT ĐÈN ĐỎ CỦA PHƯƠNG TIỆN GIAO THÔNG TRÊN VIDEO

Lại Ngọc Thăng Long – Mentor: TS. Nguyễn Văn Nam
Hà Nội, tháng 6 năm 2021

1. ABSTRACTS

Phát hiện hành vi vượt đèn đỏ của phương tiện giao thông trên video là điều tối quan trọng trong việc phát triển hệ thống giao thông thông minh và hiện đang thu hút lượng lớn sự quan tâm và nghiên cứu từ các doanh nghiệp, công ty cũng như các nhà khoa học. Trong bài báo cáo này, em tìm hiểu và đề xuất phương pháp để phát hiện hành vi vượt đèn đỏ của phương tiện giao thông thông qua chuỗi chuyển động của phương tiện. Phương pháp mà em sử dụng trong bài toán này đó là: LSTM + Attention và sử dụng thêm một số kĩ thuật để tăng metrics đánh giá. Tập dữ liệu sử dụng gồm 1618 object phương tiện và các feature chứa thông tin về phương tiện. Cuối cùng model đạt F1 score: 0.9933 trên tập Test.

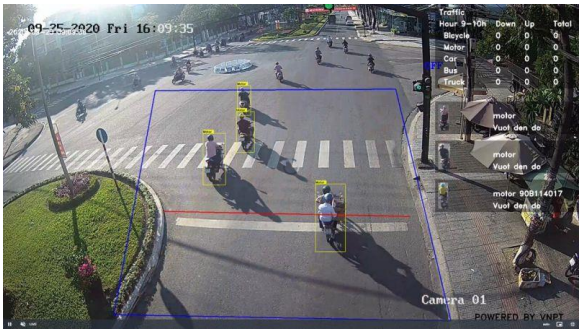
2. INTRODUCTION

Hiện nay, thực trạng giao thông tại Việt Nam đã có nhiều tiến bộ bởi nhận thức tham gia giao thông của người dân cũng như sự vào cuộc của các cơ quan chức năng. Tuy nhiên vẫn còn đó một bộ phận không nhỏ người dân chưa chấp hành đúng khi tham gia giao thông mà vấn đề xảy ra phổ biến đó chính là hành vi vi phạm vượt đèn đỏ. Việc phát hiện và xử lí các phương tiện vi phạm không phải là dễ dàng bởi mật độ giao thông ở các thành phố lớn khá đông đúc và lực lượng chức năng để thường xuyên theo dõi và phát hiện cũng khó đảm bảo 24/24. Vì vậy việc phát hiện và xử lí vi phạm giao thông truyền thống vẫn còn gặp nhiều hạn chế. Cùng với đó, xu hướng phát triển đô thị thông minh, ứng dụng chuyển đổi số, công nghệ hiện đại vào cải thiện các vấn đề hàng ngày cũng được quan tâm và phát triển với tốc độ chóng mặt. Từ đó, việc xây dựng hệ thống giao thông thông minh là hết sức cần thiết và cấp bách, thu hút sự quan tâm và nghiên cứu của các cơ quan, doanh nghiệp.

Hình 1. Tình trạng vượt đèn đỏ khi tham gia giao thông tại VN

Trong phạm vi đề tài này, em xây dựng một module giúp nhận diện hành vi vượt đèn đỏ của phương tiện giao thông – một module nhỏ trong hệ thống giao thông thông minh nhằm giúp cải thiện việc nhận diện và phát hiện phương tiện vi phạm giao thông từ đó giúp có hướng xử lí các phương tiện vi phạm, góp phần thúc đẩy ý thức tham gia giao thông của người dân tốt hơn.

Để phát hiện được phương tiện vượt đèn đỏ, chúng ta cần phát hiện và theo dõi phương tiện. Trong phạm vi project này, đã có dữ liệu về phương tiện được phát hiện và tọa độ qua các frame cắt từ video, với input đầu vào là file text lưu thông tin của phương tiện, nhiệm vụ cần làm là nhận diện hành vi có vượt đèn đỏ hay không từ dữ liệu đầu vào. Điểm khác biệt ở phương pháp đề xuất so với các phương pháp được sử dụng trước đây đó là xây dựng một model có khả năng phát hiện hành vi thông qua chuỗi hành động của đối tượng thay vì sử dụng sơ đồ trạng thái.

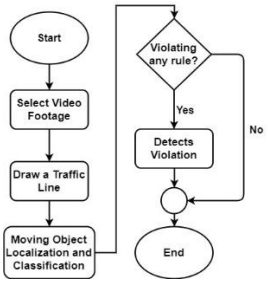


Hình 2. VD về việc áp dụng vào thực tế để phát hiện và xử lí phương tiện vi phạm

Sau khi xây dựng và thử nghiệm thì kết quả cao nhất đạt được khi sử dụng LSTM + Attention trên tập dataset sử dụng feature tọa độ trái trên, phải dưới của bounding box.

3. RELATED WORK

Các phương pháp gần đây thường sử dụng sơ đồ trạng thái: đầu tiên từ video đầu vào sử dụng một model để phát hiện phương tiện (YOLO), sau đó tracking phương tiện và kẻ 1 vạch giao thông cố định để xác định phương tiện có vi phạm trong lúc tín hiệu đèn đang màu đỏ hay không.



Hình 3. Flow diagram phát hiện vi phạm vượt đèn đỏ trong các phương pháp gần đây

Phương pháp này hiện đạt F1 score 95,7% trên bộ test với 90 id.

Phương pháp đề xuất chỉ sử dụng thông tin về chuỗi chuyển động của phương tiện để dự đoán hành vi của phương tiện, phương pháp tỏ ra hiệu quả khi đạt F1 score 99,33% trên bộ test với 162 id và thời gian inference đạt 0.2ms/id.

4. DATASET

Tập dữ liệu thô gồm các folder chứa thông tin về các phương tiện giao thông được phát hiện và theo dõi ở Vũng Tàu, Hà Nam, Thái Nguyên. Mỗi folder chứa 149 frame ảnh và 1 file text lưu thông tin gồm 7 trường:

Column 1: 2 (là frame_id ghi là 002 trong tên file)
Column 2: là id của phương tiện
Column 3: là loại phương tiện (0: xe máy, 1: xe tải, 2: xe đạp, 3: ô tô, 4: đi bộ, 5: xe khách)
Column 4,5: tọa độ x_min, x_max của hình bao đóng phương tiện
Column 6,7: tọa độ y_min, y_max của hình bao đóng phương tiện

VD về dataset:

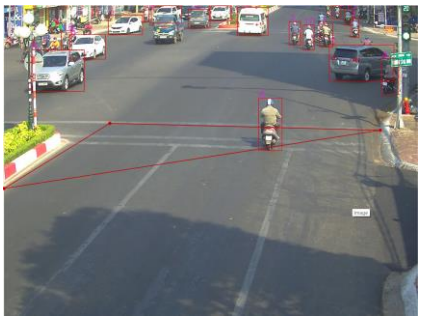
Hình 4. Tọa độ id phương tiện và frame ảnh của id

Preprocessing

- Khử các nhiễu và sai sót trong quá trình detect và tracking, xử lí lại file text (có 1 file text trong folder tập train bị đánh sai thứ tự và trùng nhưng thông tin không đúng theo frame).

- Sử dụng Labelme để vẽ vùng xét các phương tiện xử lí, lọc ra các phương tiện cần xử lí trong vùng xét. Tách từ file info.txt ra thành từng file chứa thông tin của từng id theo các frame.

- Chuẩn hóa MinMaxScale(0,1), kẻ line đại diện cho vạch vượt dừng đèn đỏ và sử dụng tọa độ bottom right của bbox để xác định phương tiện đã vượt qua line hay chưa để gán nhãn: 0: không vượt đèn đỏ, 1: vượt đèn đỏ.



Hình 5. Preprocess data

- Sau khi tiền xử lí được 1618 objects, trong đó mỗi object có len_sequence: 147, lưu ra 2 options:

+ Option 1: mỗi sequence có 4 features (tọa độ trái trên phải dưới bbox)

+ Option 2: mỗi sequence có 2 features (tọa độ x, y của tâm bbox)

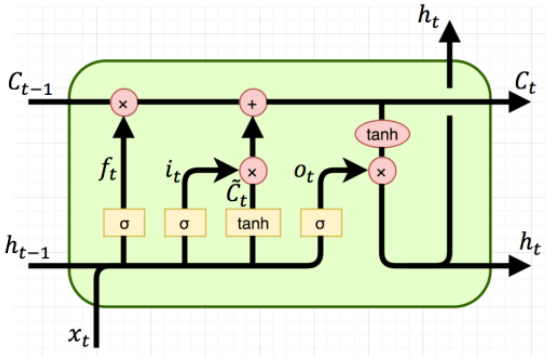
Trong đó gồm: 697 objects có nhãn 0; 921 objects có nhãn 1.

Chia tập train, validation, test: 80% train, 10% validation, 10% test.

5. METHODOLOGY

5.1 LSTM

LSTM là bản mở rộng phát triển từ mạng RNN. LSTM được thiết kế nhằm giải quyết vấn đề phụ thuộc xa của mạng RNN truyền thống. Mạng RNN thường gặp vấn đề vanishing gradient nên các hidden state ở xa sẽ bị mất mát thông tin từ các hidden state ở đầu cách xa nó.



Hình 6. Mô hình LSTM.
Nguồn: https://medium.com/@saurabh.rathor092/simple-rnn-vs-gru-vs-lstm-difference-lies-in-more-flexible-control-5f33e07b1e57

Output: c_t, h_t : c là cell state, h là hidden state

Input: c_{t-1}, h_{t-1}, x_t: x_t là input ở state thứ t của model. c_{t-1}, h_{t-1} là output của layer trước.

f_t,i_t, o_t tương ứng là forget gate, input gate và output gate

Forget gate: loại bỏ những thông tin không cần thiết nhận được khỏi cell state

$$f_t = \sigma(U_f * x_t + W_f * h_{t-1} + b_f)$$

Input gate: chọn lọc những thông tin cần thiết nào được thêm vào cell state

$$i_t = \sigma(U_i * x_t + W_i * h_{t-1} + b_i)$$

Output gate: xác định những thông tin nào từ cell state được sử dụng như đầu ra

$$o_t = \sigma(U_o * x_t + W_o * h_{t-1} + b_o)$$

b_f, b_i, b_o là các hệ số bias. W, U là các ma trận tham số

$$c\sim_t = tanh(U_c * x_t + W_c * h_{t-1} + b_c)$$

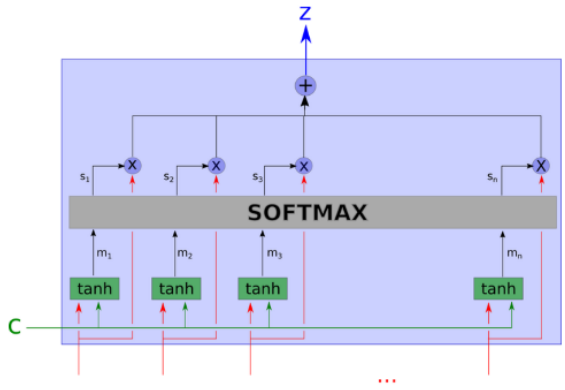
$c_t = f_t * c_{t-1} + i_t * c_{t-1}$: forget gate quyết định xem cần lấy bao nhiêu từ cell state trước và input gate sẽ quyết định lấy bao nhiêu từ input của state và hidden layer của layer trước.

$h_t = o_t * \tanh(c_t)$: output gate quyết định xem cần lấy bao nhiêu từ cell state để trở thành output của hidden state. Ngoài ra h_t cũng được dùng để tính ra output y_t cho state t.

$h_t, c\sim_t$ khá giống với RNN, nên model có short term memory. Trong khi đó c_t giống như một băng chuyền ở trên mô hình RNN vậy, thông tin cần quan trọng và dùng ở sau sẽ được gửi vào và dùng khi cần nên có thể mang thông tin đi xa (long term memory). Do đó mô hình LSTM có cả short term memory và long term memory.

5.2 Attention

Việc encode toàn bộ thông tin từ input vào 1 vector cố định khiến mô hình khi thực hiện trên các chuỗi dài (long sentence) không thực sự tốt, mặc dù sử dụng LSTM để khắc phục điểm yếu của mạng RNN truyền thống với hiện tượng Vanishing Gradient, nhưng như thế vẫn chưa đủ, đặc biệt đối với những chuỗi dài hơn những chuỗi trong training data. Từ đó, kết hợp đầu ra của layer LSTM với cơ chế Attention cho phép mô hình có thể chú trọng vào những phần quan trọng thay vì chỉ sử dụng context layer được tạo ra từ layer cuối cùng của Encoder.



Hình 7. Cơ chế attention. Nguồn: Slide VDT 2021

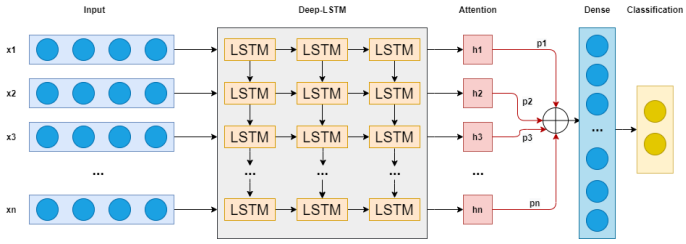
Input đầu vào (chính là output của tầng LSTM) được chia thành n phần, mỗi phần biểu diễn bởi một vector $h_1, h_2, \ldots\ h_n$. Context c. Model sẽ trả về một vector z – "summary" toàn bộ hidden states liên quan đến context c.

$$m_i = \tanh\ (Wh_i + U_c + b)$$

$$p_i = softmax(< u, m_i >)$$

$$z = \sum_i p_i h_i$$

5.3 LSTM + Attention



Hình 8. Kiến trúc mô hình

Input đầu vào có kích thước: Nx147xD với N là số object, 147 là độ dài sequence và D=4 là feature tọa độ trái trên, phải dưới của bbox nếu sử dụng tập dataset A hoặc D=2 là feature tọa độ tâm của bbox nếu sử dụng tập dataset B

Phần Deep-LSTM gồm 3 tầng LSTM chồng lên nhau, trong đó:

- Tầng LSTM thứ nhất nhận input đầu vào là 147xD và output 147x100

- Tầng LSTM thứ hai nhận input đầu vào là 147x100 và output 147x100

- Tầng LSTM thứ ba nhận input đầu vào là 147x100 và output 147x100

Attention nhận output tầng LSTM cuối cùng làm input đầu vào và output vector ***z*** 128 chiều. Cuối cùng vector ***z*** đưa qua layer phân loại 2 lớp, output: 0 nếu không vượt và 1 nếu vượt.

5.4 Binary CrossEntropy Loss

$$BCE = -t_1 \log(q_1) - (1 - t_1)\log(1 - q_1)$$

Giả định với đầu ra gồm 2 lớp C_1, C_2. $t_1[0,1]$ và q_1 lần lượt là groundtruth và giá trị dự đoán của model cho lớp C_1. $t_2 = 1 - t_1$ và $q_2 = 1 - q_1$ tương ứng là groudtruth và giá trị dự đoán của model cho lớp C_2. Mục tiêu của hàm loss là làm cho giá trị dự đoán gần nhất với groundtruth.

6. EXPERIMENTS AND RESULTS

Huấn luyện và thử nghiệm trên môi trường: Google Colab với GPU: Tesla T4

6.1 Evaluation Metrics

Metrics được sử dụng để đánh giá cho bài toán này là F1 score (trung bình điều hòa của Precision và Recall):

$$\boldsymbol{F1\ score} = \frac{\boldsymbol{2 * Precision * Recall}}{\boldsymbol{Precision + Recall}}$$

Trong đó: $\boldsymbol{Precision} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FP}}$, $\boldsymbol{Recall} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FN}}$

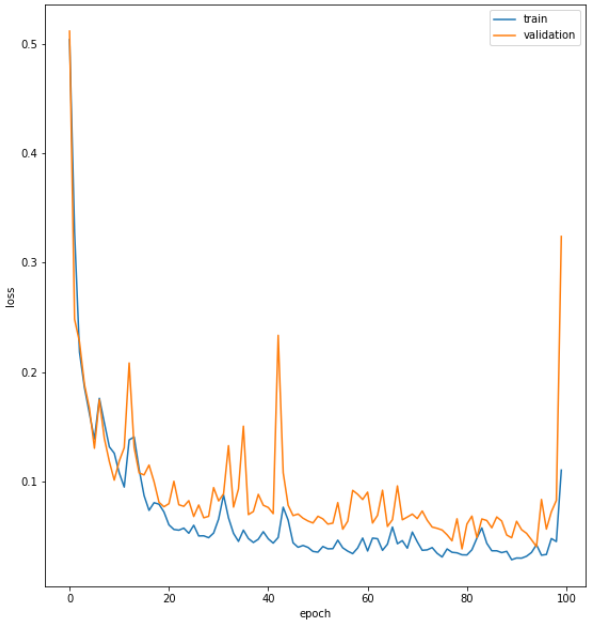
6.2 Experiments

Các phương pháp đều được set cùng bộ hyper parameters để cho kết quả so sánh khách quan nhất.

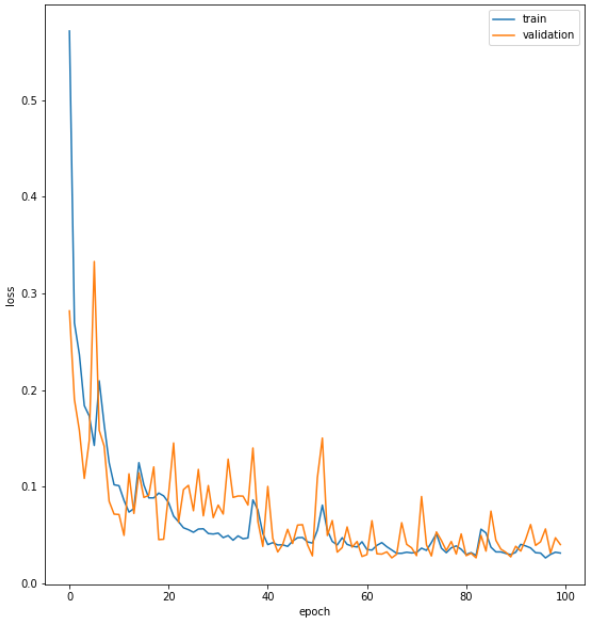
Chiến lược huấn luyện: Train 100 epochs, batch_size: 128, optimizer: Adam với lr = 1e-3. Sử dụng regularization: Dropout (0.2)

Sử dụng model checkpoint để lưu lại kết quả tốt nhất.

6.3 Results



Đồ thị loss trên tập A



Đồ thị loss trên tập B

Bảng 1. Kết quả trên tập train, validation, test

	Loss			Precision			Recall			F1		
	Train	Validation	Test	Train	Validation	Test	Train	Validation	Test	Train	Validation	Test
Tập A	0.0435	0.0684	0.0191	0.9907	1.0000	0.9885	0.9803	0.9867	1.0000	0.9863	0.9962	0.9933
Tập B	0.0398	0.0326	0.0878	0.9898	1.0000	1.0000	0.9850	0.9923	0.9754	0.9873	0.9961	0.9872

Thời gian training: 3,5s

Thời gian inference: 0.2ms/id

6.4 Evaluate

Bảng kết quả cho thấy model LSTM + Attention cho kết quả rất tốt (F1 đạt 0.9933) và việc sử dụng tập dataset A (sử dụng feature tọa độ trái trên phải dưới của bbox) cho kết quả tốt hơn việc sử dụng tập dataset B (sử dụng feature tọa độ tâm của bbox). Điều này có được vì lúc preprocessing sử dụng tọa độ bottom right để gán nhãn phương tiện vượt hay không do đó lúc inference thì sử dụng tọa độ trái trên, phải dưới bbox sẽ phát hiện tốt hơn so với tọa độ tâm bbox.

Kết quả cho thấy model nhận diện hành vi tốt, tuy nhiên còn bị nhầm một số trường hợp như:

(train\Dendo1\thainguyen_23.0.avi_save)

ID: 99 từ frame 124 cho đến frame 147. Label là 0 (vì lúc gán nhãn frame cuối vẫn chưa vượt lane kẻ) nhưng model predict 1. Do độ dài frame chỉ đến 147 mà đến frame 147 thì xe vẫn đang nằm trên vạch (chưa vượt vạch). Do dữ liệu chưa có hết hành trình của xe mà chỉ đến frame 147 nên không bắt được hết hành vi.

Một số trường hợp chưa xử lí:

(\train\Dendo1\thainguyen_40.0.avi_save)

ID 35 đi bộ ngang đường chưa xử lí

7. CONCLUSION AND FUTURE WORK

Mô hình hiện tại cho kết quả khá tốt sau khi huấn luyện, tuy vẫn còn một số lỗi nhưng việc ứng dụng trong thực tế để phát hiện hành vi vượt đèn đỏ của phương tiện giao thông là ứng dụng được.

Hướng phát triển tương lai:

- Thu thập thêm dữ liệu, bao quát hết các trường hợp và xử lí dữ liệu sạch

- Xử lí các trường hợp như được phép rẽ phải, xử lí chiều ngược lại, lấn làn, đi ngang đường, dừng quá vạch dừng,…

- Thử nghiệm phân loại 5 class (vượt, không vượt, ngược chiều, rẽ trái, rẽ phải) thay cho 2 class(vượt, không vượt)

- Thử nghiệm một số model mới như: Transformer,…

8. REFERENCES

[1] Bài giảng VDT 2021.

[2] Dzmitry Bahdanau, KyungHyun, Cho Yoshua Bengio. NEURAL MACHINE TRANSLATION BY JOINTLY LEARNING TO ALIGN AND TRANSLATE, 2015.

[3] Colin Raffel, Daniel P. W. Ellis. FEED-FORWARD NETWORKS WITH ATTENTION CAN SOLVE SOME LONG-TERM MEMORY PROBLEMS, 2016.

[4] Fazle Karima, Somshubra Majumdarb, Houshang Darabia, Samuel Harforda. Multivariate LSTM-FCNs for Time Series Classification, 2019.

[5] https://medium.com/@saurabh.rathor092/simple-rnn-vs-gru-vs-lstm-difference-lies-in-more-flexible-control-5f33e07b1e57

[6] Abu Noman Md. Sakib, Pias Roy. Traffic Signal Violation Detection System using Computer Vision, 2019.