

# 딥러닝 기반 차량 경로예측 모델의 전파과정 간소화를 통한 트레이드오프 성능 검증 연구

오기성\*, 김희중\*, 임세준<sup>o</sup>

## A Study on Trade-off Evaluation by Simplifying Propagation Process of Vehicle Trajectory Prediction Model Based on Deep Learning

Geesung Oh\*, Heejung Kim\*, Sejoon Lim<sup>o</sup>

### 요약

본 논문에서는 딥러닝 기반의 차량 경로예측 모델의 예측 정확도와 예측시간 사이의 Trade-off 관계에서 성능을 개선하기 위한 전파과정 간소화 방안을 제안하였다. 차량 경로예측 문제에서 딥러닝 기술을 활용하면서 점점 높은 정확도의 예측이 가능해지고 있지만, 반면에 모델의 복잡도가 증가하면서 예측시간이 길어지는 Trade-off 문제를 발생시킨다. 이 문제를 해결하기 위해 기존의 딥러닝 기반 state-of-the-art 차량 경로예측 모델에 여러 가지 전파과정 간소화 방안을 적용하고 실험을 통해 예측 정확도와 예측시간을 측정하였다. PC 및 임베디드 환경에서의 검증을 통하여 주변차량의 Dynamic Motion 특성을 추출하는 과정을 간소화할 경우, 예측 정확도 손실 없이 예측시간을 PC 환경에서는 15.7%, 임베디드 환경에서는 2.1% 단축했다.

**Key Words** : Trajectory Prediction, Deep-learning, Trade-off, Dynamic Motion, Interdependency

### ABSTRACT

In this paper, a method of simplifying the propagation process to improve the performance in the trade-off relationship between prediction accuracy and prediction time of vehicle trajectory prediction model based on deep learning is proposed. In vehicle trajectory prediction tasks, it is possible to predict increasingly high accuracy by using deep learning technology, but on the other hand, a trade-off problem occurred because of the increasing the prediction time of the model according to increased complexity. To solve this problem, various simplifying methods of propagation process were applied to the existing deep learning based state-of-the-art vehicle trajectory prediction model and the prediction accuracy and prediction time were measured through the experiments. In case of simplifying from the process to extract the Dynamic Motion features of neighboring vehicles, the predicted time was reduced by 15.7% in the PC environment and 2.1% in the embedded environment without any loss of the predicted accuracy.

※ 본 연구는 2020년도 산업통상자원부 'AI기반자율주행컴퓨팅모듈개발및서비스실증사업'의 재원으로 산업기술평가관리원(KEIT) 연구비 지원(20005673) 및 산업통상자원부 '산업전문인력역량강화사업'의 재원으로 한국산업기술진흥원(KIAT)의 지원(미래형자동차 R&D 전문인력 양성사업, 과제번호: N0002428)을 받아 수행된 연구입니다.

♦ First Author : Kookmin University Graduate School of Automotive Engineering, gsethan17@kookmin.ac.kr, 학생회원

° Corresponding Author : Kookmin University Dept. of Automobile and IT Convergence, lim@kookmin.ac.kr, 정회원

\* Kookmin University Graduate School of Automotive Engineering, jkdt333@kookmin.ac.kr

논문번호 : 202012-334-C-RE, Received December 29, 2020; Revised February 10, 2021; Accepted February 17, 2021

## I. 서 론

지능형운전자보조시스템(ADAS, Advanced Driver Assistance System)은 첨단 감지 센서를 통해 주행 중 위험 상황을 차량이 인지하고 판단하여, 스스로 제어하거나 운전자가 미리 위험요소를 감지할 수 있도록 알려주는 운전자 보조 시스템이다. 많은 완성차 업체에서 지능형운전자보조시스템을 차량에 적용하고 있음에도 불구하고 2019년 교통사고 건수는 전년도 대비 5.7% 증가하였고 그중 75.5%는 차대차 사고이다.<sup>[1]</sup> 따라서 차대차 사고 위험 상황을 더 정확하게 인지하기 위해 차량 경로예측 연구가 다방면으로 이루어지고 있으며 특히 딥러닝을 활용한 연구가 높은 예측 정확도를 보이며 주목받고 있다. 딥러닝 기반의 차량 경로예측 모델에서는 주로 두 가지 특성, Dynamic Motion(차량 움직임)과 Interdependency(상호의존성)를 추출하여 차량의 경로를 예측하며, 최근 Interdependency 특성 추출 방법을 고도화한 연구들이 좋은 예측 정확도를 보인다.<sup>[2,3]</sup> 하지만 예측 정확도를 높이기 위해 딥러닝 모델은 점점 복잡해지고 이에 따라 예측시간은 점점 증가한다. 이처럼 예측 정확도와 예측시간은 Trade-off 관계에 있지만, 더 높은 정확도를 위해 예측시간 증가 문제는 간과되고 있다. 특히 지능형운전자보조시스템은 위험요소로부터 차량을 안전하게 제어하기 위해서 빠르게 위험 상황을 인지하는 것이 필수적이며, 이러한 일련의 작업은 제한된 임베디드 환경에서 이루어져야 하므로 차량 경로예측은 예측 정확도와 예측시간, 두 가지 성능을 모두 고려해야 한다.

Deo et al.은 Recurrent Neural Network(RNN) 및 Convolutional Neural Network(CNN)를 바탕으로 한 데이터 기반의 차량 경로예측 모델을 제안하였고 여러 실험을 통해 차량의 비선형 움직임 예측에 있어 기존의 칼만 필터 및 가우시안 혼합 모델링보다 우수하다는 것을 입증하여 state-of-the-art 성능을 달성했다.<sup>[2]</sup> 또한 선행연구에서 딥러닝 모델의 전과과정을 간소화하기 위해 Long Short Term Memory(LSTM) 구조를 Fully Connected(FC) 구조로 대체하였고 이를 통해 예측시간이 단축된다는 것이 입증된 바 있다.<sup>[4]</sup> 위 연구결과들을 바탕으로 본 연구에서는 Deo et al.이 제안한 state-of-the-art 차량 경로예측 모델<sup>[2]</sup>을 기반으로 전과과정 간소화를 다양하게 적용한 모델을 제안한다. 예측 정확도 및 예측시간 측정 실험을 통해 제안된 모델 중 최적의 Trade-off 성능을 보이는 전과과정 간소화 방안을 제시한다. 먼저 2장에서는 선행연

구조사를 통해 최근 딥러닝 기반의 차량 경로예측 모델의 세부 단계를 구분하여 특성 및 역할을 파악한다. 3장은 차량 경로예측 문제를 정의하고 2장에서 구분된 딥러닝 모델의 단계별 전과과정 간소화 적용방안을 소개한다. 마지막 4장에서는 3장에서 소개된 전과과정 간소화 적용방안을 바탕으로 구성된 딥러닝 모델을 제안하고 성능을 비교 및 검증한다. Linux PC 환경에서 딥러닝 모델의 학습 및 평가를 진행하고, 학습된 모델은 임베디드 환경에서의 예측시간 성능도 검증한다.

## II. 관련 연구

차량경로 예측은 ADAS를 위한 위험 상황 인지 기술의 핵심으로 오랫동안 많은 연구가 이루어져 왔으며 최근에는 데이터 기반의 경로예측 연구가 물리적 특성 기반 연구의 장기예측에 한계를 극복하면서 더욱 활발하게 진행되고 있다.<sup>[5]</sup> 데이터 기반의 경로예측에는 데이터 필터를 활용한 연구<sup>[6]</sup>도 다양하게 진행되고 있지만, 딥러닝을 활용한 차량 경로예측 연구가 좋은 성능을 발휘하면서 주목받고 있다. 딥러닝 차량 경로예측 모델은 주로 차량의 개별움직임 특성인 Dynamic Motion과 주변차량 간 상호의존성 특성인 Interdependency를 추출하여 예측경로를 생성한다.

### 2.1 Dynamic Motion 특성 추출

스탠포드대학의 딥러닝을 기반으로 사람의 이동경로를 예측하는 연구에서 사람들의 각기 다른 움직임 특성을 학습하고 예측하기 위해 LSTM 구조를 활용하였다.<sup>[7]</sup> LSTM 구조는 시계열 데이터 예측의 강점을 보이는 RNN 구조의 일종으로 Hidden-state에 장기적인 정보를 담당하는 Cell-state 구조를 추가하여 입력데이터가 장기 시계열 데이터일 경우 나타나는 성능 하락을 보완해낸 구조이다.<sup>[8]</sup> 그림 1은 LSTM 구조를 도식화한 그림이며 이러한 특성을 활용하여 기계번역, 손글씨, 날씨, 주가 예측 등과 같은 시계열 데이터 분야에서 다양하게 활용되고 있다. 스탠포드대학의 연구로부터 영감을 받은 많은 차량 경로예측 연구에서 각 차량의 Dynamic Motion 특성을 학습 및 예측하기 위해 LSTM 구조를 적용하였다. 그중에서도 2018년도에 제안된 LSTM Encoder-Decoder 구조는 Dynamic Motion 특성을 추출뿐 아니라 예측경로 생성 부분에도 LSTM 구조를 활용하였다.<sup>[9]</sup> 기존차량의 예측경로를 생성한다면 해당 차량의 Dynamic Motion 특성을 추출하는 LSTM 구조의 Cell-State 벡터를 저

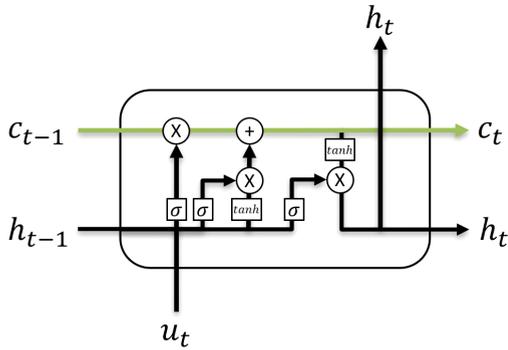


그림 1. LSTM 구조도 및 Cell-State(연두색)  
Fig. 1. LSTM Diagram and Cell-State(light green)

장했다가 경로를 생성하는 LSTM 구조의 Cell-state 초기 벡터로 사용하여 예측 정확도 향상을 보였다.

### 2.2 Interdependency 특성 추출

경로예측에 있어서 개별움직임 특성만큼 중요한 것은 Interdependency 특성이다. 따라서 물리적 특성 기반의 경로예측 연구에서도 역시 Interdependency 모델링을 위한 연구가 많았으나 스탠포드대학에서 Social Pooling<sup>[7]</sup>이라고 하는 데이터 기반 Interdependency 학습 방법을 제안하면서 많은 차량 경로예측 연구의 기초가 되었다. 실제 사람은 주변 사람들의 움직임에 대하여 추론하면서 경로를 조정하므로 경로예측 모델도 기준 사람의 움직임 특성 정보와 인접한 사람들의 움직임 특성 정보를 직접 연결 및 공유하여 경로를 예측하는 전략이다. Convolutional Social Pooling은 차량 경로예측을 위해 Social Pooling과 동일하게 기준차량의 움직임 특성을 주변 차량의 움직임 특성과 연결하지만, 연결 방식에 있어서 직접 연결 방식이 아닌 합성곱 연산 방식을 활용한다. 이를 통해 공간적 특성을 기반으로 주변차량의 Interdependency 특성을 추출하고 기준차량의 움직임 특성과 결합한다. 이러한 방식은 기존의 Social Pooling 대비 효과적인 결합방식이라는 것을 실험을 통해 입증하였다.<sup>[2]</sup> 그 외에도 차량 간의 Interdependency는 단순히 공간적 특성으로만 설명 가능한 것은 아니므로 공간적 특성 외에서 오는 Interdependency 특성을 추출하고자 자연어 처리 목적으로 제안되었던 Multi Head Attention<sup>[9]</sup> 딥러닝 기법을 적용한 차량 경로예측 연구도 최근 소개되었다.<sup>[3]</sup>

이처럼 딥러닝 기반의 차량 경로예측 모델은 발전을 거듭하면서 예측 정확도가 상승하고 있다. 하지만

한편으로는 우리의 목소리 또한 커지고 있다. 예측 정확도 향상을 위해 딥러닝 모델은 점점 더 복잡해지므로 모델의 완성도와 실시간성 제약 사이의 Trade-off 문제를 지적한다.<sup>[5]</sup> 이는 본 연구에서 주목하고 있는 예측 정확도와 예측시간의 Trade-off 관계와 동일한 관점이다.

## III. 본 론

### 3.1 문제정의

도로 위의 기준차량을 임의로 선정하고 기준차량과 주변차량의 과거 이동경로를 입력받아 기준차량의 주행경로를 예측한다. 주변차량은 기준차량을 제외한 관측범위 내 모든 차량을 말한다. 관측범위는 기준차량을 중심으로 전후방  $l$  m 및 인접한 좌우  $w$  개 차선까지이며 원점을 예측시점 ( $T_t$ )의 기준차량 절대 위치 ( $x_0^T, y_0^T$ )로, Y축은 예측시점 ( $T_t$ )의 기준차량이 주행하고 있는 차선의 기울기 방향과 평행하게, X축은 Y축의 직각 방향으로 정의하여 2차원 좌표계로 나타낼 수 있다. 이와 같이 관측범위를 상대 좌표계로 나타냄으로써 차량 내 온보드형태로 적용이 용이하다.<sup>[2]</sup> 그림 2는 차량 경로예측을 위해 관측범위 내 기준 및 주변 차량, 과거 이동경로, 예측 및 실제 주행경로를 도식화한 그림이다.

차량의 경로는 d Hz 마다의 위치 좌표를 통해 나타낸다. 과거 이동경로(X)는 (1)식과 같이 관측범위 내 모든 차량들의 이동경로의 집합이며,  $N$ 은 관측범위 내 모든 차량의 댓수를 나타내고 기준차량은  $X_0$ 로 표

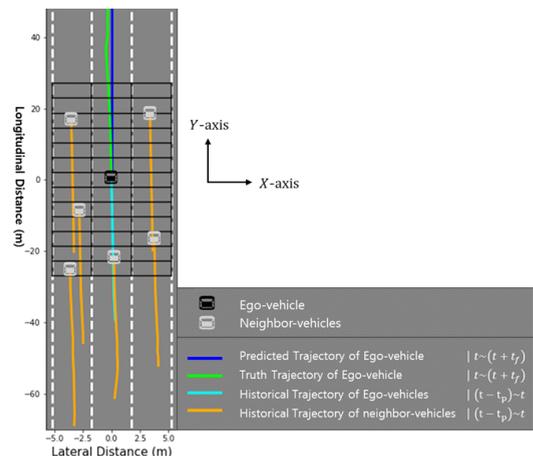


그림 2. 차량 경로예측 문제 개요도  
Fig. 2. Schematic Diagram of Vehicle Trajectory Prediction Task.

현한다. 관측범위 내 각 차량의 과거 이동경로는 (2) 식과 같이 예측시점( $T_t$ )을 기준으로 과거  $T_p$ 초 동안의 상대 좌표계 상의 위치들의 집합 형태로 입력된다. 출력값인 기준차량의 예측경로( $\hat{Y}$ )는 예측시점( $T_t$ ) 이후  $T_f$ 초까지의 상대 좌표계 상의 예측 위치들의 집합 형태로 출력되며 (3)식과 같다.

$$X = [X_0, X_1, \dots, X_N] \quad (1)$$

$$X_t = \left[ \left( x_i^{t-\frac{t}{d}} - x_0^t, y_i^{t-\frac{t}{d}} - y_0^t \right) \right], (t = 0, 1, \dots, d \times T_p) \quad (2)$$

$$\hat{Y} = \left[ \left( x_0^{t+\frac{t}{d}} - x_0^t, y_0^{t+\frac{t}{d}} - y_0^t \right) \right], (t = 1, 2, \dots, d \times T_f) \quad (3)$$

### 3.2 알고리즘

본 연구에서는 차량 경로예측을 위해 딥러닝 기반의 예측 알고리즘을 설계한다. 해당 알고리즘은 크게 세 가지 영역으로 구분되며 Dynamic Motion 추출기, Interdependency 추출기, 경로 생성기 영역으로 구분된다. 그림 3은 전체 알고리즘의 구조도를 보여주는 그림이다.

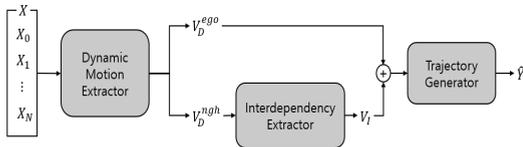


그림 3. 알고리즘 구조도  
Fig. 3. Algorithm Diagram

#### 3.2.1 Dynamic Motion 추출기

본 연구의 딥러닝 알고리즘에서는 기준차량의 주행 경로를 예측하기 위해 가장 먼저 관측범위 내 모든 차량의 개별 Dynamic Motion 특성을 추출한다. Dynamic Motion 추출기에서 각 차량의 개별 Dynamic Motion 특성을 벡터( $V_D$ ) 형태로 출력한다. 출력된 특성 벡터( $V_D$ )는 기준차량의 특성 벡터( $V_D^{ego}$ )와 주변차량의 특성 벡터( $V_D^{gh}$ )로 구분되어 기준차량의 특성 벡터는 바로 경로 생성기로 전달되고 주변차량의 특성 벡터는 Interdependency 추출기를 거쳐 경로 생성기로 전달된다.

##### 3.2.1.1 LSTM 구조 기반 추출기

관련 연구에서 언급되었던 것과 같이 LSTM 구조는 시계열 데이터를 다루는 연구에서 좋은 성능을 보

이며 주목받고 있다. 다양한 차량 경로예측 연구에서도 차량의 Dynamic Motion 특성을 추출하는 역할로 사용된다. 각 차량의 과거 이동경로를 나타내는 위치 좌표( $X$ )는 순차적으로 LSTM 구조에 입력되면서 불필요한 정보는 삭제되고 유용한 정보만이 Cell-state 상에 남아 개별 Dynamic Motion을 효과적으로 학습한다. 이러한 구조는 순차적으로 들어오는 입력데이터( $X$ )에 대한 특성 추출에 유리하지만, 순환식 전과과정은 예측시간 측면에서 불리하게 작용한다.

##### 3.2.1.2 FC 구조 기반 추출기

LSTM 구조가 가지고 있는 순환식 전과과정을 간소화하기 위해 본 연구에서는 FC 구조 기반의 추출기를 구성한다. 이는 기존의 추출기와 동일하게 차량의 개별 Dynamic Motion을 추출하지만 각 차량의 과거 이동경로를 나타내는 위치 좌표( $X$ )가 한 번에 입력된다. 입력된 정보를 바탕으로 일회성 전과과정을 통해 특성 벡터를 빠르게 출력한다. LSTM 구조보다 전과 방식이 간소하므로 추출되는 특성 정보가 경로예측을 위해 부족함이 없다면 더 우수한 성능을 기대할 수 있다. 따라서 4장에서 기존의 LSTM 구조 기반의 Dynamic Motion 추출기를 FC 구조 기반으로 변경해 보고 실험을 통해 예측 정확도와 예측시간의 변화를 확인한다.

##### 3.2.2 Interdependency 추출기

Interdependency 추출기는 Dynamic Motion 추출기를 통해 출력된 주변차량의 Dynamic Motion 특성 벡터( $V_D^{gh}$ )를 입력받아 차량 간의 상호의존성 정보가 담겨있는 Interdependency 특성 벡터( $V_I$ )를 출력한다. 본 연구에서는 Interdependency 특성을 추출하기 위해 Convolutional Social Pooling 구조를 사용한다.<sup>[2]</sup> Convolutional Social Pooling 구조는 입력받은 주변차량의 Dynamic Motion 특성 벡터( $V_D^{gh}$ )를 해당 차량의 위치 정보와 결합한다. 공간적 특성과 결합된 정보를 바탕으로 상호 간의 의존성 정보를 추출하여 Interdependency 특성 벡터( $V_I$ ) 형태로 출력한다. 주변차량의 위치 정보를 결합하기 위해 앞서 정의되었던 관측범위와 Interdependency 추출기의 입력데이터( $V_D^{gh}$ )를 ( $G_x, G_y$ ) 그리드 형태로 나타낸다. 관측범위 내에서 ( $g_x, g_y$ )번째 그리드에 위치한 주변차량의 Dynamic Motion 특성 벡터는 Interdependency 추출기 입력데이터에서도 ( $g_x, g_y$ )번째 그리드에 채워져  $V_D^{gh}(g_x, g_y)$ 와 같은 형태가 된다. 이렇게 채워진 입력

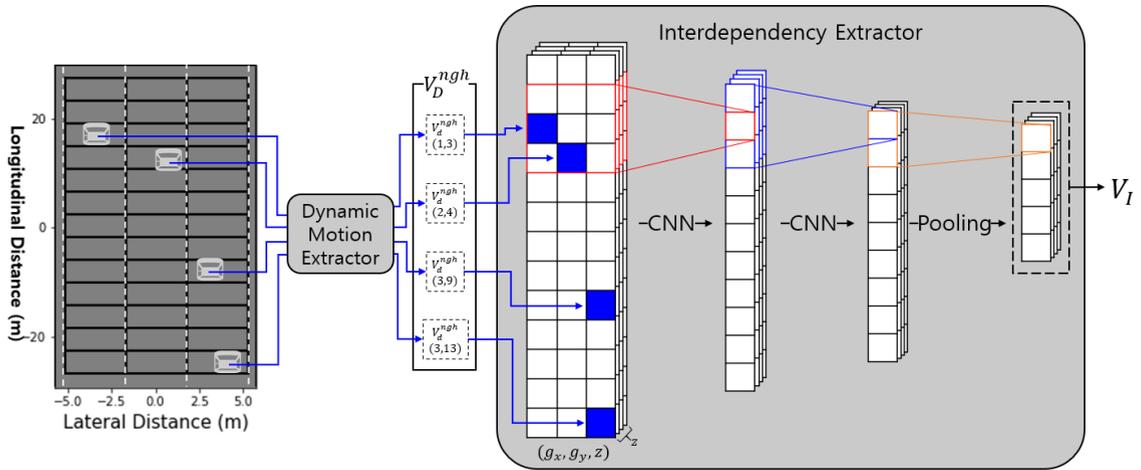


그림 4. Interdependency 추출기 구조도  
 Fig. 4. Interdependency Extractor Diagram

데이터는 그림 4와 같이 합성곱층 및 풀링층을 지나 Interdependency 특성 벡터( $V_I$ )를 출력한다.

### 3.2.3 경로 생성기

경로 생성기에서는 기준차량의 Dynamic Motion 특성 벡터( $V_d^{ego}$ )와 Interdependency 특성 벡터( $V_I$ )를 결합하여 최종 출력값인 기준차량의 예측경로( $\hat{Y}$ )를 출력한다.

#### 3.2.3.1 LSTM 구조 기반 생성기

LSTM 구조를 통해 결합된 기준차량의 Dynamic Motion과 Interdependency 특성 벡터( $V_d^{ego} + V_I$ )를 반복적으로 입력하면서 예측경로를 나타내는 위치 좌표( $\hat{Y}$ )를 순차적으로 출력한다. 이때 출력값은 기준차량의 주행경로를 예측하는 것이기 때문에 기준차량의 Dynamic Motion 특성을 바탕으로 경로를 생성하면 더 정확한 경로예측이 가능하다.<sup>[10]</sup> 따라서 Dynamic Motion 추출기에서 기준차량의 Dynamic Motion 특성을 추출할 때 생성된 Cell-state 값으로 경로 생성기의 Cell-state 값을 초기화하여 사용한다.

#### 3.2.3.2 FC 구조 기반 생성기

FC 구조 기반 생성기에서는 FC 구조 기반의 추출기와 마찬가지로 일회성 전과과정을 통해 예측경로를 생성한다. 한 번의 특성 벡터( $V_d^{ego} + V_I$ ) 입력으로 기준차량의 예측경로( $\hat{Y}$ )를 출력한다. 이 같은 특징을 바탕으로 4장에서는 기존의 state-of-the-art 모델의 LSTM 구조 기반 경로 생성기를 FC 구조로 변경하여

모델을 구현하고 실험을 통해 예측 정확도와 예측시간의 Trade-off 관계에서 성능을 검증한다.

## IV. 실험 및 결과 분석

### 4.1 데이터셋

딥러닝 기반의 차량 경로예측 모델의 학습 및 평가를 위해 미국 연방고속도로관리국에서 제공하는 실도로 차량 경로 데이터 Next Generation Simulation(NGSIM) 데이터셋을 사용한다. NGSIM 데이터셋은 미국 주간고속도로 제80호선<sup>[11]</sup>과 일반 국도 제101호선<sup>[12]</sup>의 일정 구간을 주행하는 차량의 데이터를 카메라를 통해 수집한 뒤 차량의 이동경로를 좌표형태로 기록한 데이터셋이다. 호선별 45분 동안 10 Hz 주기로 관측한 데이터이며 차량 이동경로 외에도 차량의 위치, 크기, 속도 및 선행차량 추종정보까지 포함하고 있어 다양한 차량 관련 연구에 활용된다. 우리는 NGSIM 데이터를 약 85%, 15%의 비율로 나누어 85%는 딥러닝 모델 학습에 활용하고 나머지 15%의 데이터를 활용하여 성능평가를 진행한다.

### 4.2 실험 세팅

Dynamic motion 추출기 및 경로 생성기의 전과과정 간소화를 통해 예측 정확도와 예측시간의 성능 변화를 확인하기 위해 아래와 같이 세 가지 딥러닝 모델을 구성하고 학습 및 성능평가를 진행하였다.

• NON-LSTM 모델

관측범위 내 모든 차량의 Dynamic Motion 특성을 FC 구조 기반의 추출기를 사용하여 추출한다. 경로 생성기 역시 FC 구조 기반의 생성기를 사용하여 전과 과정을 최대한 간소화한다.

• EGO-LSTM 모델

관측범위 내 주변차량의 Dynamic Motion 특성은 FC 구조 기반의 추출기를 사용하여 추출하고 기준차량의 Dynamic Motion 특성은 LSTM 구조 기반의 추출기를 사용하여 추출한다. 경로 생성기도 LSTM 구조 기반의 경로 생성기를 구성하고 Cell-state를 기준차량의 Dynamic Motion 추출기에서 저장된 Cell-state로 초기화하여 기준차량의 예측경로를 출력한다.

• ALL-LSTM 모델

Deo et al.이 제안한 모델<sup>[2]</sup>과 동일한 구조로써 관측범위 내 모든 차량의 Dynamic Motion 특성을 LSTM 구조를 기반으로 추출하고 경로 생성기 역시 LSTM 구조 기반으로 구성하고 Cell-state 초기화를 통한 기준차량의 Dynamic Motion 특성을 바탕으로 경로를 생성한다.

Interdependency 추출기는 세 가지 모델이 모두 동일하게 Convolutional Social Pooling 구조를 사용한다.<sup>[2]</sup>

모든 딥러닝 모델은 동일한 조건에서 실험이 진행된다. 관측범위는 전후방 27.4m(총 54.8m), 인접한 좌우 1개 차선(총 3개 차선)으로 지정하고 관측범위 내 모든 차량의 과거 3초 동안의 이동경로를 입력받아 기준차량의 미래 5초까지의 예측경로를 출력한다. 차량의 경로는 5 Hz 마다의 위치 좌표를 통해 나타낸다.

딥러닝 학습은 Linux PC(i9-9980XE, Nvidia Titan RTX) 환경에서 진행되며 예측경로( $\hat{Y}$ )와 실제경로( $Y$ ) 간의 평균 제곱 차이(Mean Squared Error)를 최소화하는 방향으로 이루어지며 식 4번과 같이 표현된다.

$$MSE = \frac{1}{n} \sum_{i=1}^n (\hat{Y}_i - Y_i)^2 \quad (4)$$

딥러닝 모델의 내부 파라미터를 최적화하기 위해 Adam 알고리즘<sup>[13]</sup>을 사용하였으며 Learning rate는 0.001로 설정하였고 1차, 2차 모멘텀은 각각 0.9,

0.009를 사용했다. 90,000,000번 이상 반복하여 학습하며 모든 학습과 평가는 TensorFlow<sup>[14]</sup> 프레임워크에서 진행된다. 학습이 완료된 차량 경로예측 모델의 예측 정확도 성능은 모델이 출력한 결과( $\hat{Y}$ )와 데이터셋에 존재하는 실제경로( $Y$ )의 평균 제곱근 오차(Root Mean Squared Error)를 기준으로 비교하며 식 5번과 같다.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{Y}_i - Y_i)^2} \quad (5)$$

예측시간의 경우, 모든 차량의 과거 이동경로가 딥러닝 모델로 입력되는 순간부터 기준차량의 예측경로가 출력되기까지의 시간을 ms 단위로 측정했다. 모든 성능평가는 Linux PC 환경에서 이루어지지만 임베디드 환경에서의 예측시간도 측정하기 위해 Nvidia Jetson TX2 Board 환경에서의 검증도 진행한다.

4.3 실험 결과

표 1은 학습이 완료된 딥러닝 모델들의 예측 정확도에 대한 비교 결과를 나타낸다. 해당 실험은 PC 환경에서 진행되었으며 예측 범위인 5초를 1초 단위로 나누어 비교하였다. ADE(Average Distance Error)는 예측시점부터 해당 시간까지의 오차의 평균을 나타낸 것이며 FDE(Final Distance Error)는 해당 시간만을 기준으로 비교했을 때의 오차를 나타낸다. NON-LSTM 모델은 다른 두 모델과 비교했을 때 전반적으로 높은 오차를 보인다. 표 1의 ADE 결과를 그래프로 표현한 그림 5를 보면 NON-LSTM 모델은 EGO-LSTM과 ALL-LSTM 모델 대비 높은 오차를 보이며 특히 1초에서의 ADE 값이 가장 크게 차이는 것을 볼 수 있다. 따라서 모든 차량에 대하여 Dynamic Motion 특성 추출기의 전과과정을 간소화할 경우, 예측 정확도 성능이 떨어지며 특히 예측시점부터 1초 사이, 초반 예측성능이 현저히 낮아진다. 반면 EGO-LSTM 모델과 ALL-LSTM 모델은 그림 5에서 볼 수 있듯이 모든 범위에서 거의 유사한 오차를 보인다. 표 1에서 살펴보면 3, 4초에서의 FDE와 4, 5초에서의 ADE는 EGO-LSTM이 ALL-LSTM 보다 적은 오차를 보이기도 한다. 이를 통해 주변차량의 Dynamic Motion 추출기 전과과정 간소화는 예측 정확도에 영향을 주지 않는 것을 확인했다.

예측시간에 단축을 비교하기 위하여 PC 환경과 TX2 Board 환경에서 실험을 진행하였고 그 결과는 표 2와 같다. ALL-LSTM 모델은 PC 환경에서

표 1. 예측 정확도 비교 결과  
Table 1. Comparison result of predict accuracy.

	Predict Horizon(s)	NON-LSTM(m)	EGO-LSTM(m)	ALL-LSTM(m)
FDE (Final Distance Error)	1	0.635	0.612	0.611
	2	1.371	1.343	1.322
	3	2.291	2.209	2.225
	4	3.477	3.367	3.396
	5	4.964	4.856	4.840
ADE (Average Distance Error)	1	0.847	0.400	0.398
	2	0.975	0.804	0.793
	3	1.366	1.258	1.255
	4	1.910	1.813	1.819
	5	2.594	2.500	2.510

12.908 ms, TX2 보드 환경에서 191.277 ms의 예측 시간을 보였다. 이를 기준으로 주변차량의 Dynamic Motion 추출기 전과과정을 간소화한 EGO-LSTM 모델의 예측시간은 PC 환경에서 약 15.7% 단축한 10.884 ms, TX2 보드에서 2.1% 단축한 187.253 ms이다. 모든 차량의 Dynamic Motion 추출기의 전과과

정을 간소화한 NON-LSTM 모델의 예측시간은 PC 환경에서 약 35.4% 단축한 8.335 ms, TX2 보드에서 2.8% 단축한 185.963 ms이다. NON-LSTM 모델은 가장 큰 예측시간 단축을 보였으나 예측 오차 또한 상승하였다. 반면 EGO-LSTM 모델은 유사한 예측 오차를 보이면서 예측시간이 단축된 것을 확인할 수 있다.

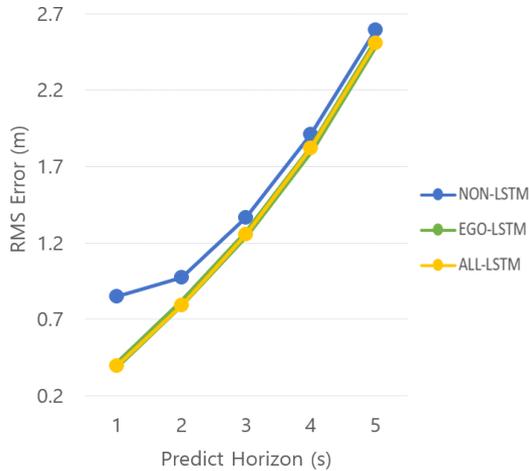


그림 5. Average Distance Error(ADE) 비교 그래프  
Fig. 5. Comparison plot of Average Distance Error(ADE)

표 2. 예측시간 비교 결과  
Table 2. Comparison result of inference time

	PC	TX2 Board
NON-LSTM(ms)	8.335	185.963
EGO-LSTM(ms)	10.884	187.253
ALL-LSTM(ms)	12.908	191.277

## V. 결 론

본 논문에서는 딥러닝 기반의 차량 경로예측 모델의 예측시간을 단축하기 위해 전과과정 간소화 방법에 관하여 연구하였다. 전과과정 간소화를 적용한 여러 모델을 제안하고 예측 정확도 및 예측시간 성능에 대한 검증은 진행하였다. 대용량의 실도로 주행데이터를 활용하여 딥러닝 모델을 학습시키고 PC 환경에서의 검증뿐만 아니라 임베디드 환경에서의 예측시간을 측정하였다. 해당 실험 결과를 통해 딥러닝 기반의 차량 경로예측 모델에서 주변차량 Dynamic Motion 추출기의 구조를 LSTM 기반에서 FC 기반으로 변경하여 전과과정 간소화한 모델은 예측 정확도 손실 없이 예측시간을 단축하는 것을 확인할 수 있었다. 하지만 임베디드 환경에서의 예측시간은 실시간 제어를 하기에는 아직도 부족한 수준이다. 따라서 향후 제한된 임베디드 환경에서도 실시간성을 보장할 수 있는 딥러닝 기반의 차량 경로예측 모델의 구조에 관하여 추가 연구를 진행하고 예측 정확도 및 예측시간의 Trade-off 성능에 대하여 검증할 계획이다.

References

[1] Korea Road Traffic Authority(KoROAD), *2019 Traffic Accident Chart Analysis*, 2020.

[2] N. Deo and M. M. Trivedi, "Convolutional social pooling for vehicle trajectory prediction," in *Proc. IEEE Conf. CVPR Wkshps.*, pp. 1468-1476, Salt Lake City, USA, Jun. 2018.

[3] K. Messaoud, I. Yahiaoui, A. Verroust-Blondet, and F. Nashashibi, "Non-local social pooling for vehicle trajectory prediction," *2019 IEEE Intell. Veh. Symp. (IV)*, pp. 975-980, Paris, France, 2019.

[4] G. Oh, H. Kim, and S. Lim, "A study on deep learning based vehicle trajectory prediction using the interdependencies of vehicle motion in real-time," *KICS Summer Conf.*, pp. 62-63, Korea, Aug. 2020.

[5] S. Lefèvre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *Robomech J.*, vol. 1, no. 1, 2014.

[6] J. Kim and D. S. Han, "A path prediction algorithm of surrounding vehicles based on sensor fusion for safe lane change," *J. KICS*, vol. 45, no. 5, pp. 828-836, May 2020.

[7] A. Alexandre, et al., "Social LSTM: Human trajectory prediction in crowded spaces," in *Proc. IEEE Conf. CVPR*, pp. 961-971, Las Vegas, USA, Jun. 2016.

[8] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, Nov. 1997.

[9] A. Vaswani, et al., "Attention is all you need," *Advances in NIPS*, vol. 30, pp. 5998-6008, 2017.

[10] S. H. Park, et al., "Sequence-to-sequence prediction of vehicle trajectory via LSTM encoder-decoder architecture," *2018 IEEE Intell. Veh. Symp.(IV)*, Changshu, China, Jun. 2018.

[11] J. Colyar and J. Halkias, *US highway i-80 dataset*, Federal Highway Administration (FHWA), Tech. Rep. FHWA-HRT-06-137,

2006.

[12] J. Colyar and J. Halkias, *US highway 101 dataset*, Federal Highway Administration (FHWA), Tech. Rep. FHWA-HRT-07-030, 2007.

[13] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," *3rd Int. Conf. Learning Representation*, pp. 1-15, 2015.

[14] *Google TensorFlow API*, Retrieved May, 12, 2021, from <https://www.tensorflow.org/>

오기성 (Geesung Oh)



2014년 2월 : 국민대학교 자동차공학과 졸업(공학사)  
 2014년 1월~2020년 1월 : 쌍용자동차 기술연구소 주임연구원  
 2020년 3월~현재 : 국민대학교 자동차공학전문대학원 자동차

차IT융합전공 석사과정  
 <관심분야> 컴퓨터공학, 자동차공학, 인공지능  
 [ORCID:0000-0002-0498-5631]

김희중 (Heejung Kim)



2020년 8월 : 국민대학교 자동차IT융합학과 졸업(공학사)  
 2020년 9월~현재 : 국민대학교 자동차공학전문대학원 자동차IT융합전공 석사과정  
 <관심분야> 자동차공학, 인공지능, 컴퓨터공학

[ORCID:0000-0002-2607-0651]

임 세 준 (Sejoon Lim)



2002년 2월 : 서울대학교 전기공학부 졸업(공학사)

2002년 1월~2005년 3월 : 알티캐스트 소프트웨어 엔지니어

2008년 2월 : Massachusetts Institute of Technology, USA, 전기컴퓨터공학부 석사

2012년 9월 : Massachusetts Institute of Technology, USA, 전기컴퓨터공학부 박사

2012년 8월~2015년 2월 : Oracle America, Inc,

2015년 3월~현재 : 국민대학교 자동차IT융합학과 부교수

<관심분야> 컴퓨터공학, 인공지능, 응용SW

[ORCID:0000-0003-1917-699X]