

## 딥러닝 기반 주행 차로 인식 기법을 활용한 차선 변경 검출 기술

\*이경민, \*송혁, \*김제우, \*최병호, \*\*인치호

\*전자부품연구원, \*\*세명대학교

\*{leegis0904, hsong, jwkim, bhchoi}@keti.re.kr, \*\*ich410@semyung.ac.kr

## Lane departure detection method using driving lane recognition based on deep learning

\*Kyung-Min Lee, \*Hyok Song, \*Je Woo Kim, \*Byeongho Choi, \*\*Chi-Ho Lin

\*Korea Electronics Technology Institute, \*\*Semyung University

## 요 약

본 논문에서는 딥러닝 기반의 주행 차로 인식 기법을 활용한 차선 변경 검출 기술을 제안한다. 제안한 방법은 주행 차로, 좌우 차로, 차량 등 3 종의 이미지 데이터를 학습, 검증, 실험 데이터로 나눠 활용하였다. 주행 차로 및 차선 변경 인식을 위하여 변형된 AlexNet 모델을 개발하였다. 실험 결과 주행 차로 69.45%, 좌우 차로 66.9%, 차량 76.4%의 인식을 결과를 보여 기존 패턴인식 방법과 비교하여 우수한 결과를 보였다.

## 1. 서론

최근 고속도로 교통사고 결과를 분석한 결과, 주행차로에서 가장 높은 사고율을 보이고 있으며, 그 다음으로 앞지르기차로 및 틀게이트 부근에서의 사고율이 높았다[1]. 고속도로 교통사고의 다수를 차지하는 사고를 방지하기 위한 능동형 차량 안전 시스템에 대한 개발이 요구되고 있으며 기존 패턴인식 방법을 활용한 상용화된 시스템도 출시되었다. 그러나, 주행차로, 앞지르기차로 및 틀게이트 부근에서의 운전자 부주의로 인한 사고 대처는 기존 시스템을 활용하여 예방하기에는 기술적으로 부족한 상황이다. 영상 처리를 활용한 차선 인식에 대한 연구는 이미 오래전부터 진행되어 왔으며, 고속도로에서 차선을 변경하는 과정을 도출하고 상황정보를 기반으로 bayesian 기법을 활용한 주행 모델을 만들어 차선 변경을 파악하는 패턴인식 기법이 제안되었다[2]. 그러나, 주행 및 차선변경시의 운전자의 행동데이터는 그 종류 및 가능성이 다양하여 모든 상황에서의 데이터를 분류하여 학습하기에는 많은 시간과 비용이 필요하다.

최근 하드웨어의 발전으로 GPU 기반 딥러닝을 이용한 영상 처리가 활용 가능한 수준까지 구현되어 신호처리 분야에서 많은 주목을 받고 있다. 딥러닝 기술은 Neural network 를 이용하여 스스로 분석한 후 답을 내는 방식으로 이미지를 이해하고 이로부터 높은 수준의 추상화된 정보를 추출하여 주행 차로 인식 및 차선 변경을 파악할 수 있다.[3]

본 논문에서는 딥러닝 기반의 차선 변경 및 주행 차로 인식 기술을 제안한다. 제안한 방법은 딥러닝 기법을 이용하여 주행 차로 인식에 기반한 차선 변경 의도를 파악하고자 한다.

2012 년에 krizhevsky 가 제안한 AlexNet 을 활용하여 차선 변경 영상 인식 기법에 적용하였다[4]. AlexNet의 구조는 아래 그림 1 과 같다. AlexNet 은 총 5 개의 convolution layers 와 3 개의 full-connected layers 로 구성되어 있으며, 마지막 FC layer 는 1000 개의 category 로 분류를 위해 활성 함수 softmax 함수를 사용하고 있다. GPU 플랫폼을 활용하여 학습 및 검출에 활용한다. 실험은 구글 이미지 검색, ImageNet, KITTI 데이터베이스를 이용하였으며, 데이터 주행 차로, 좌우 차로, 차량의 3 가지 클래스로 학습, 검증, 실험 데이터셋으로 구분하여 활용하였으며 제안한 방식을 적용해 주행 차로 인식 및 차선 변경을 확인한다.

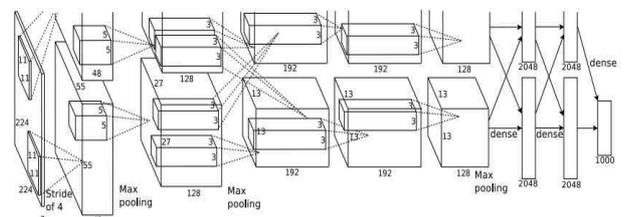


그림 1. AlexNet 의 구조

## 2. 본론

제안하는 방법에서는 구글 이미지 검색, ImageNet, KITTI



그림 2. 주행 차로 학습 데이터

이미지 데이터를 이용하여 아래 그림 2 와 같이 배경을 제거하였으며 주행 차로 부분을 잘라낸 이미지, 주행 차로가 아닌 이미지, 차량 이미지로 학습 데이터와 실험 데이터를 만든다.

기존 AlexNet 네트워크는 FC layer 에서 1000 개로 category 로 분류하는 구조로 제한한 방법에서는 주행 차로, 다른 차로, 차량의 3 종류의 category 로 그림 3. 과 같이 변형한다. 학습 데이터를 변형된 AlexNet 네트워크 구조를 이용하여 학습한다.

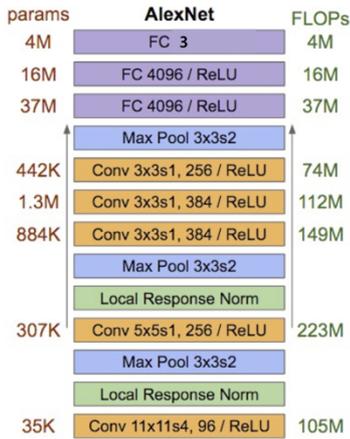


그림 3. 변형 AlexNet 구조

변형된 AlexNet 구조를 학습하여 입력 이미지에서의 클래스와 위치를 그림 4 와 같이 확인하여 주행 차로와 좌우 차로를 구분하여 운전자의 차선 변경 의도를 파악할 수 있다. 식 1 을 통해 그림 4 와 같은 영상 특징을 보여줄 수 있다.

$$M_c(x, y) = \sum_k W_k^c f_k(x, y), \text{ Conv.shape} = (k, x, y)$$

식 1. Class Activation Map



그림 4. 다른 차선의 영상 특징.

### 3. 실험

실험 환경은 i7-5930 CPU3.50GHz, RAM 16GB, GeForce GTX TiTan X 와 같고 학습은 Batch 크기가 128 인 stochastic

gradient descent 을 이용하여 진행하였다. 실험 데이터로는 총 3350 장이 제공된다. 학습 데이터는 1045 장을 이용하였고, 과적합(Overfitting)을 해결하기위해 학습 데이터의 40%의 418 장을 검증 데이터를 나뉘었으며 테스트 데이터로 833 장을 이용하였다. 총 epoch 1000, 학습률(learning rate) 0.01, 검증 데이터와 같이 과적합을 해결하기 위해 드랍아웃(dropout) 0.5 값으로 설정하여 실험하였다.

표 1 은 테스트 데이터 833 장에 대한 각 클래스의 인식률 값을 나타내었다.

표 1. 각 클래스 인식률

	Road	Non-Road	Car
인식률(%)	69.45%	66.9%	76.4%

### Acknowledgement

본 논문은 2017 년도 서울시 도시문제 해결형 기술개발 지원사업 ( 과제번호 2016-시정-04) 의 지원을 받아 수행한 결과입니다.

### 4. 결과

본 논문에서는 딥러닝 기반의 주행 차로 인식 기법을 활용한 차선 변경 검출 기술을 제안하였다. 주행 차로, 좌우 차로, 차량의 3 종의 이미지 데이터를 학습, 검증, 실험 나눠 활용하였으며, 주행 차로 및 차선 변경 인식을 위하여 변형된 AlexNet 모델을 개발하였다. 실험 결과 주행 차로 69.45%, 좌우 차로 66.9%, 차량 76.4%의 인식률 보였다. 데이터셋을 공개된 데이터베이스를 통해 수집하였기 때문에 영상의 환경이 각기 다르고 데이터 수로 인한 낮은 인식률은 데이터의 정제를 통한 데이터와 환경 변화 별로 클래스를 분할하는 것을 이용하면 더 나은 성능을 보일 것이라 판단된다.

[1] TAAS 교통사고분석시스템, 도로교통공단, 2016.  
 [2] D. Polling, M. Mulder, M.M. Van Paassen, and Q.P.Chu, "Inferring the driver's lane change intention using context-based Dynamic Bayesian networks", *IEEE International Conference on*, Vol. 1, pp.853-858. 2005.  
 [3] D. Salvucci and Andrew Liu, "The time course of a lane change: Driver control and eye-movement behavior", *Transportation Research Part F: Traffic Psychology and Behavior*, Vol.5, No.2, pp.123-132, 2002.  
 [4] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks", *Advances in neural information processing systems*, 2012.