

# Split-Attention 백본 네트워크를 활용한 차선 인식에 관한 연구

## A Study on Lane Detection Based on Split-Attention Backbone Network

송인서\* · 이선우\*\* · 권장우\*\*\* · 원종훈\*\*\*\*

\* 주저자 : 인하대학교 전기컴퓨터공학과 석사과정  
 \*\* 공저자 : 인하대학교 전기컴퓨터공학과 박사과정  
 \*\*\* 교신저자 : 인하대학교 컴퓨터공학과 교수  
 \*\*\*\* 공동교신저자: 인하대학교 전기공학과 교수

In seo Song\* · Seon woo Lee\*\* · Jang woo Kwon\*\*\* · Jong hoon Won\*\*\*\*

\* Dept. of Electrical and Computer Eng., Inha Univ.  
 \*\* Dept. of Electrical and Computer Eng., Inha Univ.  
 \*\*\* Dept. of Computer Eng., Inha Univ.  
 \*\*\*\* Dept. of Electronic Eng., Inha Univ.  
 † Corresponding author : Jang woo Kwon, jwkwon@inha.ac.kr, Jong hoon Won, jh.won@inha.ac.kr

Vol.19 No.5(2020)

October, 2020

pp.178~188

pISSN 1738-0774

eISSN 2384-1729

<https://doi.org/10.12815/kits.2020.19.5.178>

2020.19.5.178

Received 5 October 2020

Revised 13 October 2020

Accepted 21 October 2020

© 2020. The Korea Institute of Intelligent Transport Systems. All rights reserved.

### 요약

본 논문에서는 split-attention 네트워크를 백본으로 특징을 추출하는 차선인식 CNN 네트워크를 제안한다. split-attention은 CNN의 특징 추출 과정에서 feature map의 각 channel에 가중치를 부여하는 방법으로, 빠르게 변화하는 자동차의 주행 환경에서 안정적으로 이미지의 특징을 추출할 수 있다. Tusimple 데이터 셋을 활용하여 본 논문에서 제안하는 네트워크를 학습·평가하였으며, 백본 네트워크의 레이어 수에 따른 성능 변화를 비교·분석 하였다. 평가 결과 최대 96.26%의 정확도로 최신 연구에 준하는 결과를 얻었으며, FP의 경우 0.0234(2.34%)로 비교 연구 중 가장 좋은 결과를 보여준다. 따라서, 실제 차량의 주행 환경 등에서도 본 연구에서 제안하는 모델을 사용하여 오인식 없이 안정적인 차선 인식이 가능하다.

핵심어 : 차선 인식, 자율 주행, 합성곱 신경망, 인공 지능, 딥 러닝

### ABSTRACT

This paper proposes a lane recognition CNN network using split-attention network as a backbone to extract feature. Split-attention is a method of assigning weight to each channel of a feature map in the CNN feature extraction process; it can reliably extract the features of an image during the rapidly changing driving environment of a vehicle. The proposed deep neural networks in this paper were trained and evaluated using the Tusimple data set. The change in performance according to the number of layers of the backbone network was compared and analyzed. A result comparable to the latest research was obtained with an accuracy of up to 96.26, and FN showed the best result. Therefore, even in the driving environment of an actual vehicle, stable lane recognition is possible without misrecognition using the model proposed in this study.

Key words : Lane detection, Autonomous driving, Convolutional neural network, AI, Deep learning

## I. 서론

### 1. 개요

차량 주변 환경을 인식하는 컴퓨터 비전(computer vision, CV) 기술은 현대 차량 개발에서 가장 집중적으로 연구되는 분야 중 하나이다. 특히, 차선 인식 기술은 그중에서도 가장 기본이면서 매우 중요한 기술 중 하나이다. 현재 상용화된 차량에는 이미 차선 인식 기술을 기반으로 한 ADAS 시스템(Yoo et al., 2020)이 적용되어 있고, 해당 기술을 기반으로 제한적인 자율 주행 기술이 적용되어 있다. 차선 인식 기술은 최근 인공 신경망(artificial neural network, ANN) 또는 딥 러닝(deep learning, DL)을 기반으로 한 연구에서 전통적인 컴퓨터 비전 기술과 비교하여 높은 정확성을 달성하였지만(Neven et al., 2018), 아직도 차량에서의 차선 인식은 높은 제약과 복잡성으로 인해 여전히 어려운 문제로 남아있다. 첫째로, 자율 주행을 위한 차선 인식은 최소한의 지연시간을 지향하기 때문에, 차량에서 이루어져야 하고, 이는 제한적인 연산 성능으로 이어진다. 차량에서는 주행거리 확보를 위해 고성능 PC에 사용되는 높은 전력을 요구하는 프로세서를 사용할 수 없다. 두 번째로, 시각적 복잡성 때문이다. 차량은 운행되는 장소, 시간에 따라 주변 환경의 밝기, 날씨 등이 급격하게 달라질 수 있으며, 이는 시각적 복잡성으로 이어진다. 그리고 이러한 시각적 복잡성은 많은 학습 데이터를 요구하는 딥 러닝의 특성상 모델 학습을 더 어렵게 만들며, 높은 정확도를 얻기 힘든 원인이 된다.

본 논문에서는, 이미 기존의 연구에서 높은 성능을 보이는 이미지 분할(image segmentation) 방식의 접근을 사용했다(Chen et al., 2018). 이미지 분할 방식은 일반적으로 합성곱 신경망(convolutional neural network, CNN)으로 구성된 특징 추출 알고리즘(feature extraction algorithm)를 이용해서 이미지의 특징(feature)을 추출하고, 추출된 특징으로부터 이미지의 분할 지도(segmentation map)를 구성하는 방식으로, 이미 많은 차선 인식 연구에 사용되었다. 이러한 이미지 분할 방식의 아키텍처는 특징 추출기(feature extractor)의 성능이 신경망의 성능에 직접적인 영향을 준다. 따라서, 이미지 분할 방식은 특징 추출기의 성능이 전체 결과에 직접적인 영향을 준다.

본 논문에서는 특징 추출기로 Resnet의 변형(variant)인 ResNeST 네트워크를 사용하였다(Zhang et al., 2020). Resnet(Zhang et al., 2016)은 기존의 심층 신경망(deep neural network, DNN)에 잔여 입력(residual input)을 추가한 네트워크로, 기존 네트워크의 기울기 소실 문제(gradient vanishing problem)를 해결하여 DNN에서 좋은 성능을 보여준다. ResNeST는 Resnet에 split-attention 개념을 추가한 네트워크로, 특징 추출 과정에서 특징맵(feature map)의 채널(channel)에 가중치(weight)를 부여한다.

## II. Resnet 및 CNN

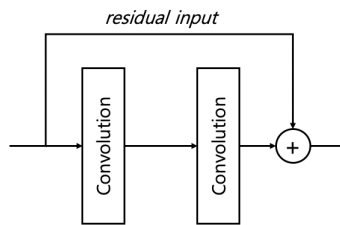
### 1. CNN

이미지 인식 분야에서 사용되는 DNN의 경우, 여러 개의 필터(filter)를 가지는 합성곱 계층을 쌓아서 구성한다. 초기의 CNN의 경우, 적은 수의 레이어를 사용했다. 예를 들어, ILSVRC 2012 대회에서 좋은 성적을 거둔 AlexNet(Krizhevsky et al., 2012)의 경우, 5개의 신경망 레이어와 3개의 완전 연결 레이어(fully connected layer, FC layer)를 사용하였다. AlexNet 이후 연구에서는, 네트워크의 합성곱 레이어 수를 늘려 더 높은 특징 추출 성능을 달성하는 방식으로 연구가 진행되었다(Szegedy et al., 2015). 일정 수까지는 레이어가 늘어날수록 더 높은 정확도를 얻을 수 있었지만, 학습 과정에서 사용되는 경사 하강법(gradient descent)(Ruder, 2016)의 특징상, 레이어 수가

일정 이상을 초과하게 되면 오히려 정확도가 떨어지는 경사 소실 문제(*gradient vanishing problem*)(Hochreiter, 1998)가 발생하게 된다. 이러한 문제를 해결하기 위해 생략 연결(*skip-connection*)을 사용하는 Resnet이 제안되었다.

## 2. Resnet

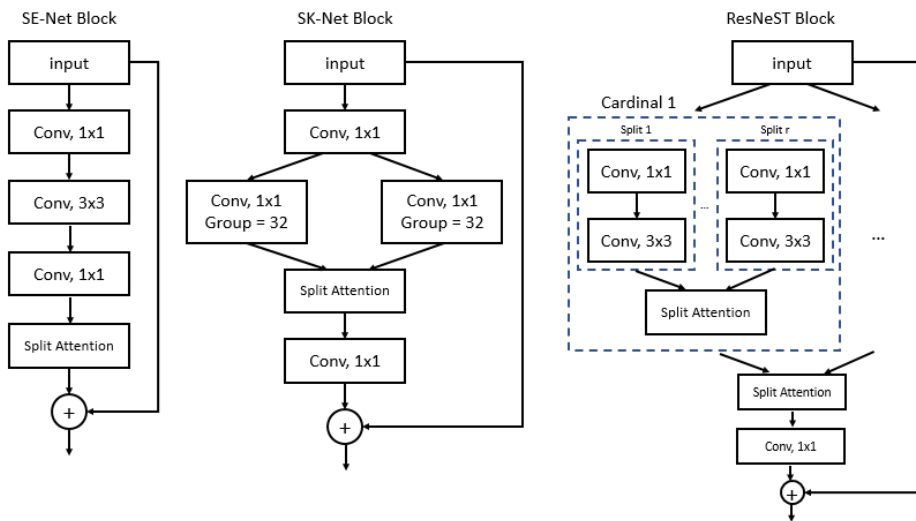
Resnet의 가장 큰 특징은 생략 연결(*skip-connection*)을 사용한다는 것이다. Resnet은 <Fig. 1>와 같이 생략 연결을 사용하는 residual block을 여러 개 연결하여 네트워크를 구성한다. 이러한 생략 연결을 통해, Resnet은 학습 과정에서 경사 소실 문제를 해결하여 더 깊은 네트워크에서도 학습할 수 있게 되었으며, 기존 연구에서 제시된 모델들과 비교했을 때 높은 수준의 특징 추출을 가능하게 한다. 따라서 많은 이미지 인식 인공지능 분야에서 Resnet을 백본 네트워크로 활용한다.



<Fig. 1> Residual block used in Resnet

## 3. Resnet 변형 모델

Resnet 이후에는 생략 연결을 그대로 사용하면서 추가적인 기법을 사용하는 Resnet 변형 모델들(*Resnet variants*)이 연구되었다. SE-Net(Hu et al., 2018)은 백본 네트워크에서 추출한 특징에 채널별 가중치를 부여하여 중요한 특징을 많이 보유하고 있는 채널에서 더 많은 정보를 추출하도록 네트워크를 구성하였다.

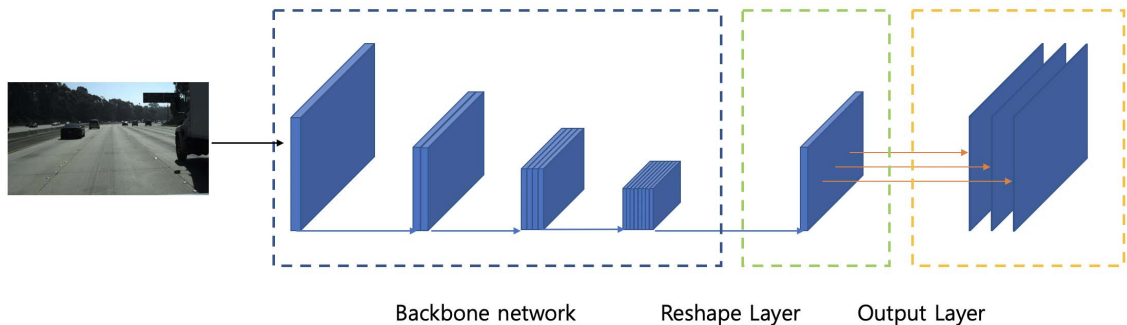


<Fig. 2> Comparison of Blocks used In SE-Net vs. SK-Net vs. ResNeST

SK-Net(Li et al., 2019)의 경우, 특징 추출 과정에서 네트워크에 두 개의 갈림길(branch)을 만들어서 두 개의 서로 다른 합성곱 레이어를 사용하여 특징을 추출하였다. 본 연구에서 백본 네트워크로 사용한 ResNeST의 경우는, 특징 추출 과정에서 특징맵(feature map)의 채널을 cardinality 그룹으로 나누고, 나뉜 그룹 내에서 특징맵을 다시 radix로 나누어 각 특징맵의 채널에 가중치를 적용하는 Split-Attention 방식을 사용한다.

### III. 제안하는 모델

본 논문에서는 백본 네트워크(특징 추출기), Reshape Layer, Output Layer(Ko et al., 2020) 3계층으로 이루어진 차선 인식 모델 및 각 계층에서 사용할 방법에 대해 제안한다. 이번 장에서는 각 계층의 자세한 구조에 대해 설명하며, Output Layer에서 사용하는 loss 함수 및 결과 특징맵(output feature map)을 이용하여 어떻게 차선 정보를 도출하는지 설명한다.



<Fig. 3> Overview of network architecture. Output Layer generates 3 feature maps(confidence, offset, feature)

#### 1. 백본 네트워크(backbone network)

많은 CNN기반 모델에서, 기존의 딥 러닝 네트워크를 특징 추출(feature extraction)을 위한 백본 네트워크로 사용한다. 백본 네트워크 자체는 특별한 목적을 수행하지 않으며, 오직 이미지의 특징을 추출하기 위한 목적으로 사용한다. 예를 들어, Faster-RCNN(Ren et al., 2015), Mask-RCNN(He et al., 2017), Cascade-RCNN(Cai et al., 2018), HTC(Chen et al., 2019) 등의 높은 성능을 보여주는 딥 러닝 네트워크는 Resnet을 특징 추출기로 사용하였다.

따라서 이러한 백본 네트워크를 사용하여 특징 임베딩(feature embedding)을 추출하는 딥 러닝 네트워크는 전체 구조를 바꾸지 않고 특징 추출기만을 변경하여 더 높은 성능을 얻을 수 있다. 본 논문에서는 Zhang et al.(2020)에서 제안한 ResNeST 네트워크를 백본으로 사용하여 특징을 추출한다.

##### 1) ResNeST 네트워크

기존의 실험 결과에서, Resnest를 백본 네트워크로 사용한 아키텍처가 최신(state-of-the-art)아키텍처와 비교해서 이미지 분류(image classification), 물체 인식(object detection), 인스턴스 분할(instance segmentation), 의미론적 분할(semantic segmentation) 등의 CNN 기반 분야에서 더 높은 정확도를 얻을 수 있음을 보여주고 있다. ResNeST 논문에서 소개하고 있는 이미지 분할(image segmentation) 성능 비교는 <Table 1>과 같다(Zhang et al., 2020).

&lt;Table 1&gt; Performance comparison in bounding box detection and image segmentation proposed by ResNeST

Prior Work	Method	Backbone	box mAP%	mask mAP%
		DCV-V2	ResNet50	42.7
	HTC	ResNet50	43.2	38.0
	Mask-RCNN	ResNet101	39.9	36.1
	Cascade-RCNN	ResNet101	44.8	38.0
Result of ResNeST	Mask-RCNN	ResNet50	38.97	36.05
		ResNet101	41.78	37.51
		<i>ResNeST50</i>	42.81	38.14
		<i>ResNeST101</i>	45.75	40.65
	Cascade-RCNN	ResNet50	43.06	37.19
		ResNet101	44.79	38.52
		<i>ResNeST50</i>	46.19	39.55
		<i>ResNeST101</i>	48.30	41.56

## 2) Resnet vs. ResNeST

Resnet은 이름에서 알 수 있듯이, 여러 개의 residual block이 중첩된 구조로 이루어져 있다. Resnet의 residual block은 두 개의 합성곱 레이어와 생략 연결으로 구성된다. ResNeST 네트워크는 Resnet과 기본 골격은 비슷하지만, Residual bottleneck block대신 Split-Attention 블록이 사용된다. Split-Attention 블록을 통해 ResNeST 네트워크는 특징맵의 각 채널에 가중치를 부여하여 더 정확한 특징 추출이 가능하도록 한다.

## 2. Reshape Layer

Reshape Layer는 backbone 네트워크의 출력을 Output Layer의 입력으로 사용하기 위한 계층이다. 백본 네트워크에  $512 \times 256 \times 3(w \times h \times c)$  이미지를 입력으로 넣었을 경우의 결과 텐서(tensor) 크기는  $2048 \times 8 \times 16$ 이다. Output Layer의 입력 텐서 크기는  $128 \times 32 \times 64$ 로, 다음과 같은 방식을 사용하여 백본 네트워크의 출력을 Output Layer의 입력 텐서로 reshape한다.

### 1) Naive Reshape approach

백본 네트워크의 출력 텐서 크기는 위에서 언급한 대로  $2048 \times 8 \times 16$ 이고, Output Layer의 입력 텐서 크기는  $32 \times 64 \times 128$ 은 이다.  $8 \times 16 \times 2048 = 32 \times 64 \times 128$ 으로, 백본의 결과 특징맵을 reshape해서 원하는 입력 텐서를 만들 수 있다. 따라서, 위 방법 같은 경우, 백본의 결과 텐서 16채널이 하나의 입력 채널을 이루게 된다.

### 2) Transpose Convolution approach

두 번째 방법으로,  $1 \times 1$  합성곱과 residual trans-convolution bottleneck block(<Table 2>의 Up-sample Bottleneck)을 사용하였다.  $1 \times 1$  합성곱을 사용하여 특징맵의 채널을 2048에서 128로 줄이고, <Table 2>의 Up-sample Bottleneck block 두 개를 사용하여 feature map의 높이(height), 너비(width)를 각각 4배로 만든 후 Output Layer의 입력으로 사용하였다.

<Table 2> Detail of bottleneck block used in Resize and Output Layer. Residual block is marked with \*

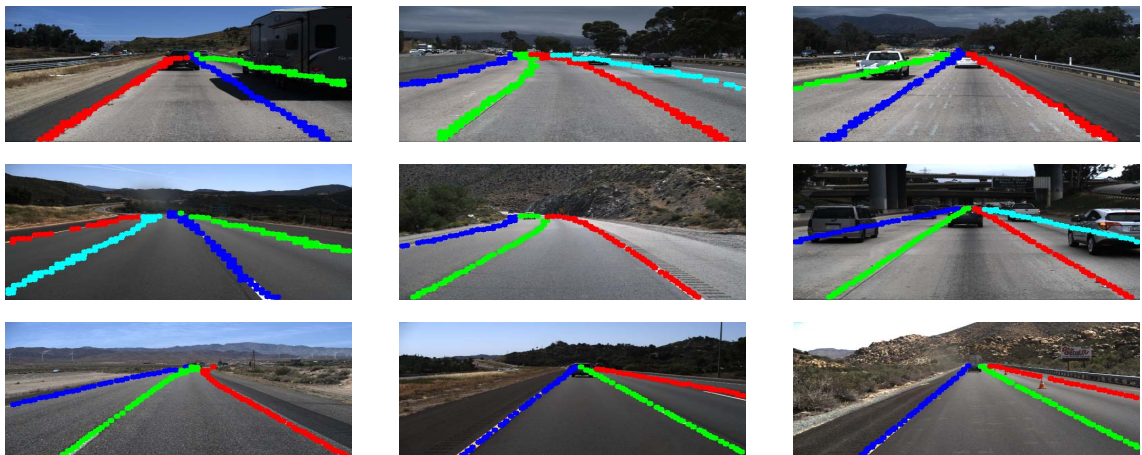
	Operation	Filter	Size/Stride
Same Bottleneck	Conv	32	1/1
	Conv	32	3/1
	Conv	128	1/1
	Conv*	128	1/1
Up-sample Bottleneck	Conv	32	1/1
	Transpose Conv	32	3/2
	Conv	128	1/1
	Transpose Conv*	128	3/2

### 3. Output Layer

Output Layer는 PINet에서 제시한 다중 가지 결과 레이어(multi-branch output layer)를 사용하였다(Ko et al., 2020). 동일한 입력이 각 가지(branch)에 입력으로 사용되고, 각 결과 가지(output branch)는 <Table 2>의 Same Bottleneck 하나로 구성된다. 해당 레이어는 Confidence Branch, Offset Branch, Feature Branch로 이루어지며, 각각 64×32의 너비, 높이 및 1,2,4채널을 가진다.

Confidence Branch는 해당 특징맵의 신뢰도 값(confidence value)을 포함한다. 신뢰도 문턱값(confidence threshold)을 넘는 점을 차선 포인트(lane point)로 선택하고, 문턱값이 더 높을수록 높은 검증 정확도(validation accuracy)를 보여준다.

Offset Branch는 출력으로 두 개의 채널을 가지며, 각 채널은 차선 포인트에 대한 x, y 오프셋(offset) 좌표를 포함한다. Offset Branch의 특징맵은 0~1의 값을 가지며, 해당 값에 리사이즈 비율(resize ratio)을 곱하여 오프셋을 계산한다. Feature Branch는 동일 차선 군집화(clustering)를 위해 사용되며, 동일 차선의 경우 특징 값이 비슷하도록 학습된다. 따라서 특징 값이 비슷한 차선 포인트(lane point)들을 하나의 차선으로 군집화한다. 시각화된 차선 인식 결과는 <Fig. 4>와 같다.



<Fig. 4> Results on Tusimple dataset. Each point is estimated by proposed model, and lane is clustered based on result of feature branch of output layer

## IV. 실험 및 결과

이번 장에서는 Tusimple 데이터 셋을 기준으로 2장에서 제시한 모델의 레이어별 구성에 따른 성능을 평가하고, 기존 연구에서 제시된 모델과 비교한다. 비교의 용이성을 위해, 본 논문과 동일한 Tusimple 데이터 셋을 사용한 연구를 대상으로 비교를 진행한다.

모델의 학습은 아래 서술된 Tusimple 데이터셋을 사용하였으며, 학습률(learning rate) 2e-4의 Adam optimizer(Kingma and Ba, 2014)를 사용하였다. epoch은 1000으로 설정되었고, 10 epoch마다 모델의 정확도(accuracy) 및 FP, FN을 평가하였다. 안정적인 결과를 도출하기 위해 이미지 뒤집기, 영상 이동, 회전, Gaussian noise 추가, 그리고 그림자 추가 등의 데이터 증강(data augmentation) 기법을 사용하였다. 모델의 학습 및 평가는 NVIDIA RTX 2080ti를 사용하였다.

### 1. Tusimple benchmark

본 논문에서는 성능 평가를 위해 Tusimple(Tusimple, 2020)에서 제공하는 벤치마크(benchmark)를 사용하였다. Tusimple 벤치마크는 데이터 셋(data set) 및 평가식(evaluation metric)으로 이루어진다. Tusimple 데이터 셋은 고속도로 환경에서 촬영된 영상 클립으로 구성된다. 각 클립은 20개의 프레임(이미지)으로 구성되어 있고, 각 클립의 20번째 프레임의 차선 정보가 어노테이션 되어있다. 따라서 본 논문에서는 ground truth(또는 annotation data)가 존재하는 3,626장의 이미지를 학습 데이터로 사용하고, 2,787장의 이미지를 테스트 데이터로 활용하였다. 모델의 성능 평가는 정확도(accuracy, Acc) 및 거짓 양성(false positive, FP), 거짓 음성(false negative, FN)에 대해 이루어졌으며, tusimple에서 제공하는 평가식은 다음과 같다.

$$accuracy = \frac{\sum_{clip} C_{clip}}{\sum_{clip} S_{clip}} \dots\dots\dots (1)$$

(1)에서  $\sum_{clip} C_{clip}$ 은 정확하게 예측된 차선 포인트이고,  $\sum_{clip} S_{clip}$ 은 클립의 전체 차선 포인트를 의미한다

$$FP = \frac{F_{pred}}{N_{pred}} \dots\dots\dots (2)$$

$$FN = \frac{M_{pred}}{N_{gt}} \dots\dots\dots (3)$$

(2), (3)에서,  $F_{pred}$ ,  $N_{pred}$ ,  $M_{pred}$ ,  $N_{gt}$ 는 각각 잘못 예측된 차선 포인트 수, 예측된 차선 포인트 수, 제대로 예측되지 못한 실제 차선 포인트 수, 실제 차선 포인트 수를 의미한다.

### 2. Ablation Study

#### 1) Radix and Cardinality

ResNeST 논문의 AblationStudy에서는 radix와 cardinality 변경에 따른 성능 테스트를 진행하였고, 그 결과는 <Table 3>와 같다. 해당 표에서 s는 radix를, x는 cardinality를, 그리고 d는 network width를 의미한다.(0s 는 ResNeST 모델이 ResNet-D(He et al., 2019)를 기반으로 하였기 때문에 기존 ResNet-D의 residual block을 의미

한다). 해당 결과가 본 논문에서 제안한 모델에도 적용되는지 검증하기 위해, ResNeST-18 및 ResNeST-50을 백본으로 하여 실험을 진행하였으며, Reshape Layer의 경우 위에서 소개한 Naive Reshape 방식을 사용하였다. 각 모델에 대해 radix:2, cardinality:1 또는 radix:4, cardinality:2로 실험을 진행하였으며, 결과는 <Table 3>와 같다. ResNeST-18 모델의 경우 정확도 0.13% 차이를 보이고, ResNeST-50 모델의 경우 정확도가 0.02% 차이로 근소하지만 4s2x 변형 모델(variant)이 더 높은 성능을 보여준다. 그리고 정확도가 높을수록 FP, FN는 낮은 경향성을 보인다.

<Table 3> Result of ablation study by ResNeST

Variant	#P	Acc(%)
0s1x64d	25.6M	79.41
1s1x64d	26.3M	80.35
2s1x64d	27.5M	80.64
4s1x64d	31.9M	80.90
2s2x40d	26.9M	81.00

<Table 4>Result of ablation study: Radix and Cardinality

Backbone	Variant	Acc(%)	FP	FN
ResNeST-18	2s1x	95.81	0.0324	0.0438
ResNeST-18	4s2x	95.94	0.0300	0.0416
ResNeST-50	2s1x	96.05	0.0291	0.0400
ResNeST-50	4s2x	96.07	0.0256	0.0377

<Table 5> Result of ablation study: Reshape vs. Transpose-Conv.

Backbone	Variant	Acc(%)	FP	FN
ResNeST-18	2s1x	96.21	0.0275	0.0398
ResNeST-18	4s2x	96.24	0.0234	0.0361
ResNeST-50	2s1x	96.24	0.0259	0.0386
ResNeST-50	4s2x	96.23	0.0254	0.0386
ResNeST-101	2s1x	96.24	0.0293	0.4000
ResNeST-101	4s2x	96.26	0.0245	0.0369

## 2) Reshape Layer : Reshape vs. Transpose-Conv.

3장에서 백본 네트워크의 결과 특징(output feature)를 변형(reshape)하는 방법으로 Naive Reshape Layer 와 Transpose Convolution Layer 두 가지 방법을 언급했다. 본 실험에서는 Reshape Layer를 Transpose Convolution Layer로 변경하여 실험을 진행하였으며, 위에서 진행한 실험과 마찬가지로 백본 및 radix, cardinality를 변경하며 실험을 진행했다. 결과(<Table 5>)를 보면, naive reshape 방법과 비교 했을 때, ResNeST-18, 2s1x의 경우 정확도가 1.03%으로 가장 많이 증가하였고, ResNeST-50, 4s2x의 경우 정확도가 0.16% 증가하여 가장 조금 증가하였다. 하지만 몇 번의 합성곱만으로 성능을 향상할 수 있었기 때문에, Transpose Convolution 방식이 더 효과적임을 알 수 있다.



아래의 결과만 비교해 보면, ResNeST-18, 2s1x 변형 모델이 96.21%로 정확도가 가장 낮고, ResNeST-101, 4s2x 변형 모델이 정확도가 가장 높다. 파라미터 수는 네트워크의 속도에 직접적인 영향을 주게 되므로, ResNeST-18 모델을 사용하는 것이 실시간 환경에서 가장 바람직하다.

### 3. 실험 결과

Tusimple 데이터 셋을 사용한 최신 모델 중 SCNN(Parashar et al., 2017), LaneNet(Wang et al., 2018), PointLaneNet(Chen et al., 2019), ENet-SAD(Hou et al., 2019), PINet(Ko et al., 2020)과 본 논문에서 제안하는 모델의 성능 비교는 <Table 7>와 같다. 가장 높은 정확도를 보여주는 모델은 ENet-SAD로, 96.64%의 정확도를 보여준다. 본 논문에서 제안하는 모델의 최고 실험 정확도는 96.26%으로 ENet-SAD와 0.38%의 정확도 차이로 근소하게 낮은 성능을 보여준다.

<Table 6> Overall result of proposed CNN model. ResNeST-101 with 4s2x variant shows best performance

Backbone	Variant	R-layer	Acc(%)	FP	FN
ResNeST-18	2s1x	reshape	95.81	0.0324	0.0438
ResNeST-18	4s2x	reshape	95.94	0.0300	0.0416
ResNeST-18	2s1x	trans-conv	96.21	0.0275	0.0398
ResNeST-18	4s2x	trans-conv	96.24	<b>0.0234</b>	<b>0.0361</b>
ResNeST-50	2s1x	reshape	96.05	0.0291	0.0400
ResNeST-50	4s2x	reshape	96.07	0.0256	0.0377
ResNeST-50	2s1x	trans-conv	96.24	0.0259	0.0386
ResNeST-50	4s2x	trans-conv	96.23	0.0254	0.0386
ResNest-101	2s1x	trans-conv	96.24	0.0293	0.4000
ResNest-101	4s2x	trans-conv	<b>96.26</b>	0.0245	0.0369

본 논문에서 제안하는 모델에서 가장 낮은 FP를 보여주는 모델은 ResNeST-18을 백본으로 사용하는 4s2x 모델으로, 0.0234의 FP결과를 보여준다. 이 수치는 비교한 모델 중 가장 좋은 성능을 보여준다. 차선 인식과 같이 위험성이 큰 분야에서, 잘못된 인식된 차선 정보는 사고 등으로 이어질 수 있으므로, FN 값은 Acc와 마찬가지로 매우 중요한 성능 지표 중 하나이다.

<Table 7> Result comparison based on Acc, FP, FN. We got best result on FP

Model	Acc	FP	FN
SCNN	96.53	0.0617	<b>0.0180</b>
LaneNet	96.38	0.0780	0.0244
PointLaneNet	96.34	0.0467	0.0518
ENet-SAD	<b>96.64</b>	0.0602	0.0205
PINet(32×16)	95.75	0.0266	0.0362
PINet(64×32)	96.62	0.0308	0.0272
ResNeST18-4s2x with trans-conv.	96.24	<b>0.0234</b>	0.0361
ResNeST101-4s2x with trans-conv.	96.26	0.2454	0.0369

## V. 결 론

본 연구에서는 기존에 최신 연구에서 사용되던 Resnet 및 다른 Resnet 변형 모델들보다 성능이 우수한 ResNeST 네트워크를 백본 네트워크로 사용하여 차선인식 모델을 구성하였다. ResNeST는 Resnet에 channel-wise-attention 기법이 도입된 네트워크로, 본 논문에서 제시하는 실험 결과, Acc 96.24%, FP 0.0234로 기존 Resnet보다 이미지 인식 분야에서 더 안정적인 성능을 보여준다. 본 연구에서 제안한 모델은 다른 최신 연구에 준하는 96.24%의 정확성을 보여주며, 비교한 모델 중 가장 높은 정확도를 보여주는 ENet-SAD와 0.38%로 근소한 차이로, 이에 준하는 성능을 보여준다. 또한 FN의 경우, 본 연구에서 제안하는 모델이 가장 좋은 실험 결과를 얻었다. 차선 인식과 같이 주변 환경에 민감하고 위험성이 큰 문제에서는, 오인식된 정보가 안전사고 등으로 이어질 수 있기 때문에, FN은 아주 중요한 성능 지표이다. 따라서 본 논문에서 제안하는 모델은 실제 차량에서 응용될 때(ADAS, 자율 주행 등) 안정적으로 동작할 수 있음을 시사한다.

그리고 본 연구에서는 백본 네트워크의 블록 수에 따른 성능을 비교하였는데, ResNeST-18 네트워크의 경우 블록 수가 18개로, ResNeST-50(블록 50개), ResNeST-101(블록 101개) 등의 네트워크와 비교했을 때, 블록 수가 50% 미만으로, 실제 자율 주행 차량 환경 등, 제한적인 연산 성능(computing power)이 주어지는 실시간 환경에서도 문제없이 동작할 수 있다. 또한, 가장 높은 정확도를 보여주는 ResNeST-101 네트워크와 비교했을 때, 0.02%의 Acc차이로, 아주 근소하게 낮은 정확도를 보여준다. 따라서, 본 연구에서 제안하는 모델은 실시간 환경에서도 높은 정확도의 모델을 적용할 수 있다.

## ACKNOWLEDGEMENTS

이 논문은 2020년도 정부(산업통상자원부)의 재원으로 한국산업기술진흥원의 지원을 받아 수행된 연구임 (N0002428, 2020년 산업전문인력역량강화사업)

## REFERENCES

- Cai Z. and Vasconcelos N.(2018), "Cascade r-cnn: Delving into high quality object detection," *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.6154-6162.
- Chen K., Pang J., Wang J., Xiong Y., Li X., Sun S. and Loy C. C.(2019), "Hybrid task cascade for instance segmentation," *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.4974-4983.
- Chen P., Lo S., Hang H., Chan S. and Lin J.(2018), "Efficient Road Lane Marking Detection with Deep Learning," *2018 IEEE 23rd International Conference on Digital Signal Processing(DSP)*, China, pp.1-5, doi:10.1109/ICDSP.2018.8631673.
- Chen Z., Liu Q. and Lian C.(2019), "PointLaneNet: Efficient end-to-end CNNs for Accurate Real-Time Lane Detection," *In 2019 IEEE Intelligent Vehicles Symposium(IV)*, pp.2563-2568.
- He K., Gkioxari G., Dollár P. and Girshick R.(2017), "Mask r-cnn," *In Proceedings of the IEEE International Conference on Computer Vision*, pp.2961-2969.
- He K., Zhang X., Ren S. and Sun J.(2016), "Deep residual learning for image recognition," *In*

- Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.770-778.
- He T., Zhang Z., Zhang H., Zhang Z., Xie J. and Li M.(2019), “Bag of tricks for image classification with convolutional neural networks,” *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.558-567.
- Hochreiter S.(1998), “The vanishing gradient problem during learning recurrent neural nets and problem solutions,” *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 6, no. 2, pp.107-116.
- Hou Y., Ma Z., Liu C. and Loy C. C.(2019), “Learning lightweight lane detection cnns by self attention distillation,” *In Proceedings of the IEEE International Conference on Computer Vision*, pp.1013-1021.
- Hu J., Shen L. and Sun G.(2018), “Squeeze-and-excitation networks,” *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.7132-7141.
- Kingma D. P. and Ba J.(2014), *Adam: A method for stochastic optimization*, arXiv preprint arXiv:1412.6980.
- Ko Y., Jun J., Ko D. and Jeon M.(2020), *Key Points Estimation and Point Instance Segmentation Approach for Lane Detection*, arXiv preprint arXiv:2002.06604.
- Krizhevsky A., Sutskever I. and Hinton G. E.(2012), “Imagenet classification with deep convolutional neural networks,” *In Advances in Neural Information Processing Systems*, pp.1097-1105.
- Li X., Wang W., Hu X. and Yang J.(2019), “Selective kernel networks,” *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.510-519.
- Neven D., Brabandere B. D., Georgoulis S., Proesmans M. and Gool L. V.(2018), “Towards End-to-End Lane Detection: An Instance Segmentation Approach,” *2018 IEEE Intelligent Vehicles Symposium(IV)*, Changshu, pp.286-291, doi:10.1109/IVS.2018.8500547.
- Parashar A., Rhu M., Mukkara A., Puglielli A., Venkatesan R., Khailany B. and Dally W. J.(2017), “Scnn: An accelerator for compressed-sparse convolutional neural networks,” *ACM SIGARCH Computer Architecture News*, vol. 45, no. 2, pp.27-40.
- Ren S., He K., Girshick R. and Sun J.(2015), “Faster r-cnn: Towards real-time object detection with region proposal networks,” *In Advances in Neural Information Processing Systems*, pp.91-99.
- Ruder S.(2016), *An overview of gradient descent optimization algorithms*, arXiv preprint arXiv:1609.04747.
- Szegedy C., Liu W., Jia Y., Sermanet P., Reed S., Anguelov D. and Rabinovich A.(2015), “Going deeper with convolutions,” *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.1-9.
- Tusimple(2017.7.18), [https://github.com/TuSimple/tusimple-benchmark/tree/master/doc/lane\\_detection](https://github.com/TuSimple/tusimple-benchmark/tree/master/doc/lane_detection)
- Wang Z., Ren W. and Qiu Q.(2018), *Lanenet: Real-time lane detection networks for autonomous driving*, arXiv preprint arXiv:1807.01726.
- Yoo H., Yang U. and Sohn K.(2013), “Gradient-Enhancing Conversion for Illumination-Robust Lane Detection,” *In IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp.1083-1094, doi:10.1109/TITS.2013.2252427.
- Zhang H., Wu C., Zhang Z., Zhu Y., Zhang Z., Lin H. and Li M.(2020), *Resnest: Split-attention networks*, arXiv preprint arXiv:2004.08955.