



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

2021년 8월

석사학위 논문

잔차 학습 신경망을 이용한 도로
네트워크 속도 예측

조선대학교 대학원

토목공학과

전승배

잔차 학습 신경망을 이용한 도로
네트워크 속도 예측

Traffic Speed Prediction of Road Networks
Using Deep Residual Learning Networks

2021년 8월 27일

조선대학교 대학원

토목공학과

전승배

잔차 학습 신경망을 이용한 도로 네트워크 속도 예측

지도교수 정 명 훈

이 논문을 공학석사학위신청 논문으로 제출함

2021년 4월

조선대학교 대학원

토 목 공 학 과

전 승 배

전승배의 공학석사학위 논문을 인준함

위원장 조선대학교 교수 박 상 준 (인)

위 원 조선대학교 교수 장 우 식 (인)

위 원 조선대학교 교수 정 명 훈 (인)

2021년 5월

조선대학교 대학원

PUBLICATIONS

본 논문은 석사과정 동안 이루어진 연구를 기반으로 쓰였습니다. 일부 아이디어, 알고리즘 및 그림과 같은 내용은 다음 논문들을 토대로 합니다.

해외 논문(SCIE)

[1] Jeon, S. B., Jeong, M. H., Lee, T. Y., Lee, J. H., & Cho, J. M. (2020). Bus Travel Speed Prediction Using Long Short-term Memory Neural Network. *Sensors and Materials*, 32(12), 4441-4447.

국내 논문(KCI)

[2] 전승배, 오행열, & 정명훈. (2020). 머신러닝을 이용한 해양수질 등급 추정. *대한공간정보학회지*, 28(4), 145-152.

국내외 학회 논문

[3] 전승배, 정명훈, 이태영, 오행열, 김건, & 조정호. (2020). LSTM 순환 신경망을 이용한 DTG 데이터의 시계열 예측. *한국측량학회 학술대회자료집*, 106-107.

[4] 전승배, 이태영, & 정명훈. (2019). Projection method 를 이용한 택시 이동 데이터 흐름에 대한 outlier 탐지에 관한 연구. *한국측량학회 학술대회자료집*, 76-77.

[5] 전승배, 이태영, & 정명훈. (2018). Minimum generalized variance를 이용한 비정상적인 택시 이동경로 탐지. *대한공간정보학회 학술대회*, 112-113.

목 차

ABSTRACT

제 1장 서론	1
1.1 연구 배경 및 목적	1
1.2 논문의 구성	3
제 2장 문헌 검토	4
2.1 모수적 방식	4
2.1.1 Autoregressive integrated moving average	4
2.1.2 Kalman filter	5
2.2 비 모수적 방식	5
2.2.1 순환 신경망	7
2.2.2 합성곱 신경망	9
제 3장 데이터 및 아키텍처	13
3.1 데이터 처리 및 분석 아키텍처	13
3.2 택시 이동데이터	15
3.3 도로 링크 데이터	16
3.4 공간 조인	17
3.5 이미지 데이터	18

3.6 도로 네트워크 선택	21
제 4장 모델 구조 및 하이퍼 파라미터	23
4.1 모델의 구조	23
4.1.1 잔차 학습	23
4.1.2 Batch normalization	25
4.1.3 모델 아키텍처	26
4.2 모델 하이퍼 파라미터 및 학습 과정	29
4.2.1 모델 하이퍼 파라미터	29
4.2.2 학습 데이터 및 과정	31
제 5장 분석 결과	34
5.1 도로 네트워크 사용 빈도 상위 5% 예측 결과	34
5.2 결과 비교	40
제 6장 결 론	42
참 고 문 헌	43
부 록	48

표 목 차

표 1.1 2012년 도로 교통혼잡비용의 지역별 비교 (Jo et al., 2014)	2
표 4.1 비교 분석을 위한 Ma et al. 's 모델 구조	27
표 4.2 Model 1: ResNet-50	27
표 5.1 Ma et al. 's 모델의 학습 결과	34
표 5.2 모델 1의 학습 결과	36
표 5.3 모델 2의 학습 결과	38

그림 목 차

그림 1.1 기존 방식과 합성곱 신경망의 차이	2
그림 2.1 단일 계층 퍼셉트론의 구조	6
그림 2.2 XOR 문제	6
그림 2.3 심층 신경망의 구조	7
그림 2.4 순환 신경망(Rumelhart et al., 1986)	8
그림 2.5 ILSVRC 우승 결과(Khan et al., 2020)	10
그림 2.6 필터와 합성곱	11
그림 2.7 Max pooling과 average pooling	11
그림 3.1 시스템 아키텍처	13
그림 3.2 포인트 데이터 시각화 (a) 축척 1/100,000 (b) 축척 1/3,000	15
그림 3.3 도로 링크 데이터 시각화	16
그림 3.4 위치에 따른 속성값 조인: 버퍼	17
그림 3.5 버퍼 길이에 따른 데이터양의 비교	18
그림 3.6 합성곱 신경망의 입력 데이터 형태	18
그림 3.7 이미지 데이터 생성 과정	19
그림 3.8 도로 네트워크 통행 속도의 시간 단위 이미지 데이터 ...	20
그림 3.9 도로 네트워크 통행 속도의 일 단위 이미지 데이터	20
그림 3.10 전체 도로의 누적 사용 횟수	21
그림 3.11 이용 빈도수 상위 5% 도로 (a) 이용 빈도수 상위 5%에 대한 히스토그램 (b) 이용 빈도수 상위 5%에 대한 도로 네트워크	22

그림 4.1	잔차 학습: a building block(He et al., 2016) (a) Identity block (b) Convolution block	24
그림 4.2	Identity block과 convolution block으로 구성된 ResNet-50	24
그림 4.3	ResNet-34를 위한 building block(왼쪽 이미지)과 ResNet-50/101/152를 위한 “bottleneck” building block (오른쪽 이미지)(He et al., 2016)	28
그림 4.4	Adam 최적화 함수의 pseudocode (Kingma and Ba, 2014)	30
그림 4.5	모델 학습을 위한 데이터 개요	32
그림 4.6	ResNet을 이용한 통행 속도 예측 아키텍처	32
그림 5.1	Ma et al.’s 모델의 epoch과 loss의 관계	34
그림 5.2	Ma et al.’s 모델의 속도 예측 결과 중 일부	35
그림 5.3	모델 1의 epoch과 loss의 관계	36
그림 5.4	모델 1의 속도 예측 결과 중 일부	37
그림 5.5	모델 2의 epoch과 loss의 관계	38
그림 5.6	모델 2의 속도 예측 결과 중 일부	39
그림 5.7	각 모델의 예측 결과값 분포	41

ABSTRACT

Traffic Speed Prediction of Road Networks Using Deep Residual Learning Networks

Jeon, Seung Bae

Department of Civil Engineering

Graduate School of Chosun University

Due to recent advancements in Internet of Things (IoT) and 5G technology, new and unprecedented opportunities have risen with the usage of Artificial Intelligence techniques (AI) for big data analytics. These developments further accelerate the digitization of human society in terms of smart cities and self-driving cars. In this cyberinfrastructure, the geospatial information has been proven to play a pivotal role in enabling researchers to develop novel theories and scientific knowledge in the domain of Intelligent Transport Systems (ITS). For the last couple of decades, researchers in the field of ITS have been trying to optimize the prediction of traffic data of road networks with variations in space and time. Since road networks have an inherent complexity due to their nonlinear characteristics, the prediction of traffic data remains a difficult task. Among the preceding works, the application of Convolutional Neural Networks (CNN) for prediction has been utilized to reflect the characteristics of traffic data on a road network with regards to time and space. However, previous studies were unable to incorporate the vanishing gradient as the depth of the model increased. This study aims to address this limitation with the use of a ResNet model constructed with transfer learning. Traffic data prediction for a road network with a complex

structure was performed by combining the road link and Mobileye network sensor data. The ResNet models used in the study are ResNet-50 and ResNet-152. After fine-tuning, ResNet-152, the model achieved a higher improvement in accuracy by 3.8% compared with previous models with a relatively shallow structure. The model proposed in this study can be useful for real-time traffic data prediction of complex road networks. The current study performed traffic data prediction by considering only space and time. In future studies, it is necessary to include other factors that may affect the road network.

제 1 장 서 론

1.1 연구 배경 및 목적

도로 네트워크의 환경에서 도로의 통행 속도에 대한 정밀한 정확도를 가진 예측은 지능형 교통 체계(intelligent transport systems, ITS), 도시 계획 및 관리, 공공복지 증진 및 안전과 같은 분야에서 중요한 역할을 한다(Weisbrod et al., 2003; Vlahogianni et al., 2014). 한국교통연구원(Jo et al., 2014)에 따르면, 2012년 교통혼잡비용이 약 30조 3,146억 원으로 추정되었으며, 표 1.1에서 확인할 수 있는 것과 같이 상대적으로 많은 인구수로 인해 교통의 수요가 많고 도로 네트워크의 구조가 복잡한 수도권 지역이 높은 비율을 차지하는 것을 확인할 수 있다. 교통 혼잡은 일반적으로 통행 속도를 기준으로 판별되며, 통행 속도는 도로 환경에서 다양한 요소에 의해 영향을 받는다. 때문에, 통행 속도는 도로 네트워크에서 중요한 지표가 된다. 또한, 네트워크 규모의 정확한 통행 속도의 예측은 전반적인 도로 교통의 상태를 예측할 수 있게 하며, 교통 혼잡을 예측 및 완화할 수 있는 기반을 마련할 수 있다(Zhao et al., 2019).

지능형 교통 체계 분야는 통행 속도에 대한 정밀한 예측을 위해 지속적인 연구를 수행해왔다(Ma et al., 2015; Jia et al., 2016). 하지만, 복잡한 구조를 가진 도로 네트워크에서의 통행 속도의 예측은 여전히 어려운 과제이다. 시간에 종속되는 통행 속도의 특성을 반영하기 위해 적용되었던 시계열 분석 기법인 Autoregressive Integrated Moving Average(ARIMA)와 순환 신경망(Recurrent Neural Network, RNN)은 복잡한 구조를 가진 도로 네트워크의 환경에서 통행 속도를 예측하기 위해 정의되어야 하는 모델의 수가 네트워크 규모에 따라 증가하는 단점이 있다. 또한, 시간만이 아니라 공간에도 종속되어 서로에게 영향을 미치는 공간적 상관관계를 고려하지 못한다는 한계도 존재한다. 이것을 보완하기 위해 제안된 것이 합성곱 신경망(Convolutional Neural Network, CNN)의 적용이다. 통행 속도를 시간과 공간으로 표현되는 이미지로 변환한 뒤 모델 정의를 위한 학습에 이용하는 것이다. 그림 1.1은 도로 네트워크 환경에서의 통행 속도를 예측하기 위해 기존의 시계열 분석 방법을 적용한 경우와 합성곱 신경망을 적용한 경우를 비교하여 나타낸다.

표 1.1 2012년 도로 교통혼잡비용의 지역별 비교(Jo et al., 2014)

구분	지역	교통혼잡비용(억 원)	비율(%)
수도권	서울·인천·경기	174,294	57.5
강원	강원	3,143	1.0
충북	충북	7,018	2.3
대전·충남권	대전·충남	19,959	6.6
대구·경북권	대구·경북	19,350	6.4
부산·경남권	부산·울산·경남	55,203	18.2
전북	전북	5,681	1.9
광주·전남권	광주·전남	15,540	5.1
제주	제주	2,958	1.0
계		303,146	100

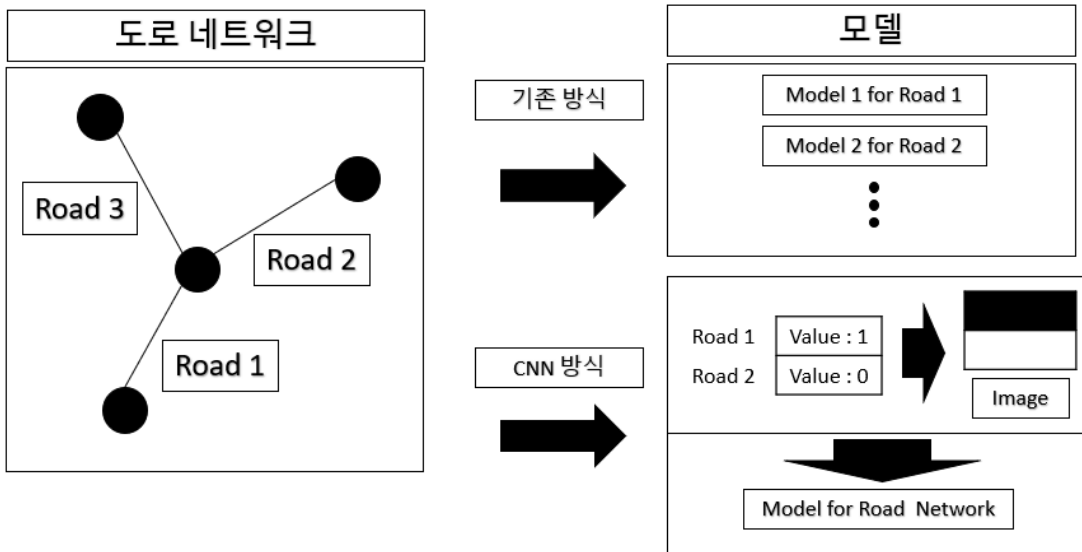


그림 1.1 기존 방식과 합성곱 신경망의 차이

기존의 방식은 도로 네트워크의 통행 속도를 예측하기 위해 네트워크를 구성하는 도로마다 모델을 정의해야 한다. 이로 인해 네트워크의 규모가 큰 경우, 예측을 수행하는 것 자체가 불가능할 수 있다. 반면에 합성곱 신경망을 적용하는 경우, 도로 네트워크를

구성하는 도로가 이미지의 한 축으로 변환되기 때문에 개별로 모델을 정의할 필요가 없어진다. 뿐만 아니라 도로 네트워크 전체를 하나의 이미지로 변환하여 학습을 진행하기 때문에 공간적 상관관계도 고려하여 예측을 수행할 수 있다. 이러한 아이디어는 Ma et al.(2017)에 의해 최초로 도입되었다. 하지만, Ma et al.'s 모델은 아주 단순한 구조를 가지고 있는 합성곱 신경망이다.

본 연구는 모델의 깊이가 깊어짐에 따라, 학습이 제대로 이뤄지지 않는 문제를 완화하고자 제시되어 더 깊은 모델의 정의를 가능하게 만든 합성곱 신경망의 일종인 잔차 학습(residual learning)을 기반으로 구성된 residual neural network(ResNet)을 전이 학습(transfer learning)하여 복잡한 구조를 가진 도로 네트워크에서의 통행 속도를 예측하였다. 연구의 목적은 비교적 단순한 구조를 가진 합성곱 신경망, Ma et al.'s 모델과 본 연구에서 제시된 모델을 비교 분석하여, 더욱 향상된 정확도를 가진 모델을 제시하는 것이다.

1.2 논문의 구성

본 연구는 기존에 통행 속도를 예측하기 위해 제시되었던 연구들의 문헌을 검토하며, 이들 중 비교적 단순한 구조를 가진 Ma et al.'s 모델과 ResNet을 전이 학습한 모델을 비교 분석하여 결과의 차이를 제시하고자 한다. 본 논문은 총 6장으로 구성된다. 위 1장의 서론에는 도로 네트워크에서의 통행 속도를 예측하는 것에 대한 중요성에 관해 서술하였다. 이후 2장의 문헌 검토에서 선행 연구들의 조사와 이론적인 고찰을 통해 본 연구의 중요성과 필요성에 대해서 논의하고, 3장에선 연구에 사용된 데이터에 관해 설명하며, 데이터의 전처리 과정과 분석 아키텍처 및 연구 범위를 명시하고 처리된 데이터를 시각화하여 제시하였다. 4장은 데이터에 적용된 모델의 배경과 구성요소들에 관해 설명하고 이후 모델의 구조와 하이퍼 파라미터(hyperparameter)를 제시한다. 5장은 데이터에 대한 모델 적용의 결과로, 각 모델의 정확도를 Mean Absolute Error(MAE)를 척도로 비교 분석하며, 예측된 결과값과 참값(ground truth)을 비교하여 시각화하였다. 마지막으로 6장의 결론은 본 연구가 가지는 한계점에 관해 서술하였으며, 향후 연구 과제에 대해 제시한다.

제 2장 문헌 검토

본 연구는 도로 네트워크 환경에서의 통행 속도 예측 분야에서, 시뮬레이션 방식과 모델링 방식 중 모델링 방식에 속한다. 이는 모수적(parametric) 방식과 비 모수적(non-parametric) 방식이 있으며, 두 방식을 결합한 하이브리드(hybrid) 방식도 존재한다(Mou et al., 2019). 본 연구에서는 모수적 방식과 비 모수적 방식에 관한 선행 연구들을 조사하였다.

2.1 모수적 방식

모수적 방식은 미리 정의된 통계 모델을 기반으로 수행된다. 이러한 모델에 대한 방정식들은 예측을 수행함에 있어 다양한 매개변수를 요구한다. 그러나 복잡하고 동적인 특성이 있는 교통 데이터와 같은 상황에서 매개변수를 정확하게 추정하기는 쉽지 않다. 또한, 모수적 방식이 적용될 데이터는 분포나 성질에 관한 여러 가지 가정을 만족해야 한다. 통행 속도의 예측을 위해 수행된 모수적 방식 중 대표적인 방법에는 ARIMA와 Kalman filter가 있으며, 이후 각각의 방법이 가지는 한계에 관해 검토하고자 한다.

2.1.1 Autoregression integrated moving average

Box et al.(1970)에 의해서 제안된 ARIMA 모델은 자기 회귀(Autoregression, AR)모델과 이동평균 모델(Moving Average, MV)의 누적된 합(integrated)을 의미한다. 자기 회귀모델은 이전의 상태가 이후의 상태에 영향을 준다는 생각에서 기인한 모형이며, 이동평균 모델은 이전의 변동을 이용하여 현재의 상태를 추론하는 모형이다. 이러한 ARIMA 모델은 통행 속도를 예측하기 위해 널리 사용된 방식이다(Ahmed and Cook, 1979; Williams et al., 1998). 하지만 ARIMA 모델은 데이터가 정상성(stationarity)을 만족해야만 적용될 수 있다. 정상성이란 데이터의 분산이 시점에 의존하지 않는 것을 의미한다. 실제 데이터는 보편적으로 정상성을 만족하지 않으므로 ARIMA는 차분을 통해 이를 보완하여 알고리즘을 적용해야만 한다. 또한, ARIMA와 같은 모수적 방식은 이상값에 대해 상대적으로 민감하며, 네트워크의 규모가 증가함에 따라 정의되어야 하는 모델의 수가 증가하는 단점이 있다.

2.1.2 Kalman filter

Kalman filter는 과거의 측정값과 새로운 측정값을 사용하여 측정값에 포함된 잡음(noise)을 제거하고 최적의 값을 추적하는데 사용하는 재귀 추정 알고리즘이다. 측정에 대해 확률적인 오차와 특정 시점에서의 상태가 이전 시점에서의 상태와 선형적일 때 적용할 수 있다. 이전 상태의 정보만을 활용하기 때문에 연산 환경의 메모리 측면에서 유리하며 연산 과정이 빠르므로 실시간 예측과 같은 문제에 적합하다. 이러한 장점이 있는 Kalman filter는 교통 데이터를 예측하기 위해 적용되었다(Okutani and Stephanedes, 1984). 하지만 Kalman filter는 각 시간 간격에서 이동 시간이 급격하게 변하면 시간 지연 문제가 발생할 수 있는 단점이 존재한다(Chen et al., 2004). 또한, 교통 네트워크와 같은 복잡한 구조를 가진 공간에서의 데이터는 변동이 선형적이지 않을 가능성이 크므로, 알고리즘이 요구하는 데이터에 대한 기본 가정을 만족하지 않을 가능성이 크다. Kalman filter는 다른 모수적 방식인 ARIMA와 마찬가지로, 도로 네트워크에서의 예측을 위해 정의되어야 하는 모델의 수와 같은 측면에서 같은 한계점이 존재한다.

2.2 비 모수적 방식

비 모수적 방식은 모수적 방식과 다르게 데이터의 정규성이 검증되지 않아도 사용될 수 있으며, 모수적 방식보다 상대적으로 이상값에 대해 덜 민감하다. 또한, 실세계의 데이터는 대부분 비선형적인 특성이 있으므로 이러한 특징을 다룰 수 있는 비 모수적 방식, 특히 인공 신경망(artificial neural network, ANN)의 적용에 관한 관심이 증가해 왔다(Dougherty, 1995). 인공 신경망은 기계 학습(machine learning, ML) 중 하나이며 퍼셉트론(perceptron)에서 유래된다. 퍼셉트론은 Rosenblatt(1957)가 제안한 초기의 인공 신경망이며, 다수의 입력을 기반으로 하나의 결과를 도출한다.

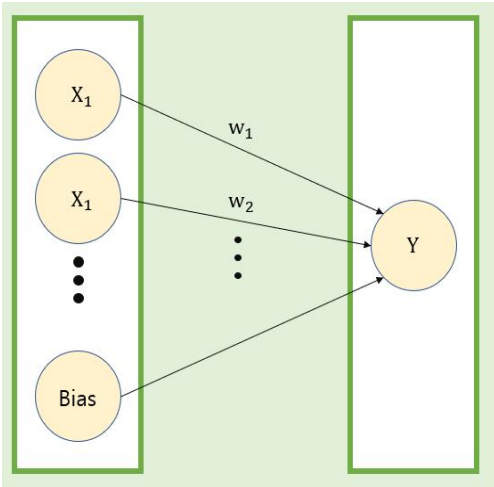


그림 2.1. 단일 계층 퍼셉트론의 구조

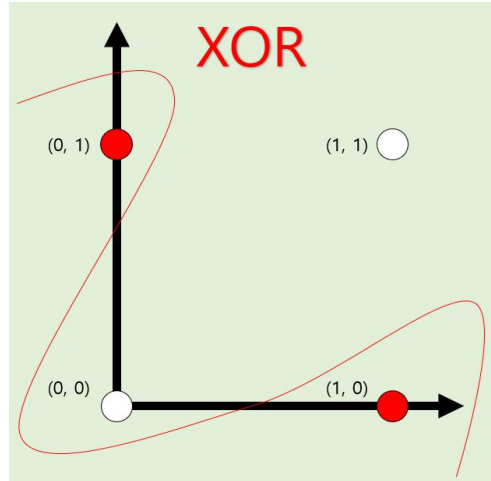


그림 2.2. XOR 문제

그림 2.1은 단층 퍼셉트론의 구조를 나타낸다. 그림 2.1의 x 는 입력값을 나타내며 w 는 각 입력 노드에 적용되는 가중치를 나타낸다. Y 는 입력값에 가중치가 적용된 각 노드에 의해 결정된다. 하지만 이러한 구조의 단층 퍼셉트론은 그림 2.2와 같은 XOR, 비선형적 분류문제를 해결하지 못한다. XOR이란 배타적 논리합(exclusive OR)을 의미하며, 주어진 두 명제 중 한 개만 참일 경우를 판단하는 논리 연산이다. 이러한 XOR 문제는 AND, NAND, OR를 조합하여 해결할 수 있다. 즉, 층을 더 쌓는 것이다. 다층 퍼셉트론과 단층 퍼셉트론의 차이는 층을 더 쌓음에 따라 추가된 은닉층(hidden layer)의 유무이다. 입력층(input layer)과 출력층(output layer)만 존재하는 단층 퍼셉트론과 달리, 다층 퍼셉트론은 중간에 은닉층이 존재한다. 그림 2.3과 같이 은닉층이 2개 이상인 신경망을 심층 신경망(deep neural network, DNN)이라 한다.

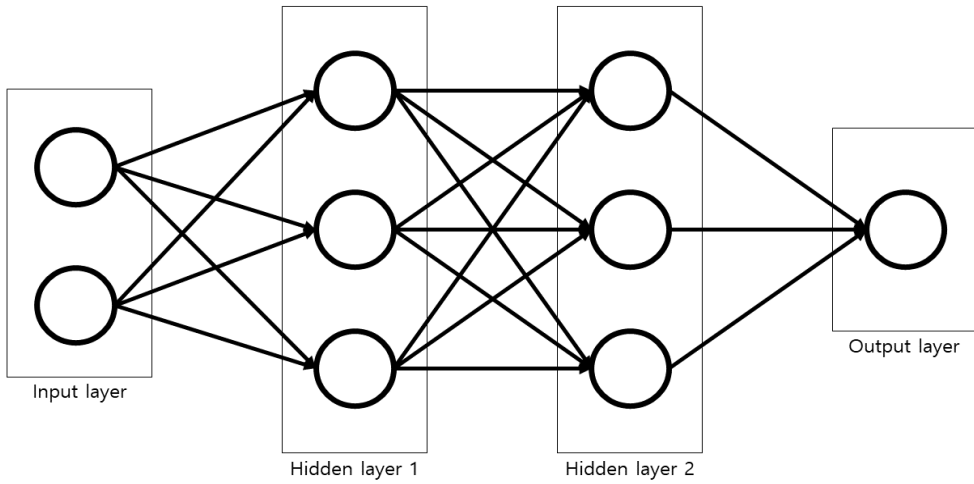


그림 2.3 심층 신경망의 구조

또한, 초기의 퍼셉트론은 가중치를 수동으로 찾아야만 했다. 하지만 입력층의 값이 많아짐에 따라 기계가 가중치를 스스로 찾아내도록 해야만 했는데, 이것이 기계 학습에서 말하는 학습 단계에 해당한다. 이 과정에서 손실 함수(loss function)와 최적화 함수(optimizer)를 사용한다. 학습을 시키는 신경망이 심층 신경망인 경우, 이를 딥 러닝(deep learning)이라고 한다. 이러한 구조를 가진 딥 러닝의 발전과 함께 등장한 순환 신경망, 합성곱 신경망은 교통 데이터를 예측하는 데 사용되어왔다.

2.2.1 순환 신경망

순환 신경망은 입력과 출력을 시퀀스(sequence) 단위로 처리할 수 있다. 이 때문에 번역하고자 하는 문장, 일련의 시계열 데이터와 같은 시퀀스 형태의 데이터를 처리하는 것에 사용되어왔다. 순환 신경망과 feed forward 신경망의 가장 큰 차이점은 출력층 방향으로만 진행되는 feed forward 방식과 달리, 순환 신경망은 순환적인 구조를 갖는다는 것이다. 순환 신경망은 신경망 내부에 상태를 저장할 수 있는 공간인 메모리가 존재한다. 이로 인해 시퀀스 형태의 입력과 출력을 처리할 수 있는 것이다. 이러한 특성을 가진 순환 신경망은 교통 데이터의 예측에 적용된 바 있다(Van Lint et al., 2002). 그림 2.4는 순환 신경망의 구조를 나타낸다.

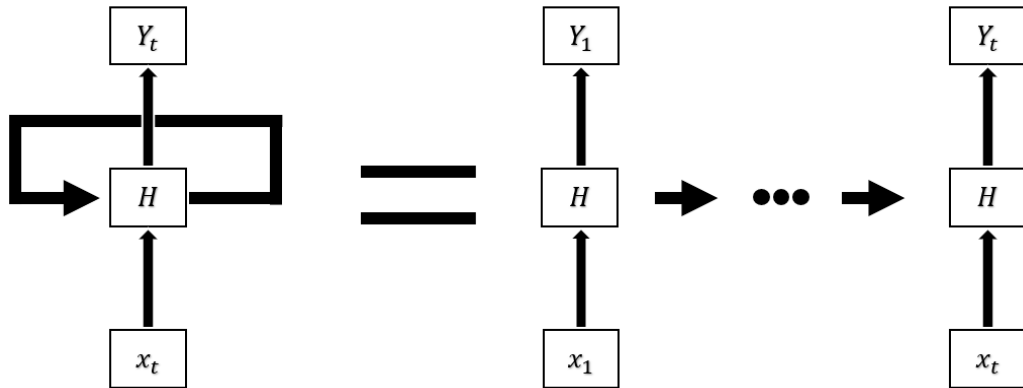


그림 2.4 순환 신경망(Rumelhart et al., 1986)

하지만 순환 신경망은 시차(time-lag)가 커지면 기울기가 소실되는 문제가 있으며, 이는 모델의 학습을 저해한다. 순환 신경망이 가지고 있는 시차의 크기에 의한 기울기 소실 문제를 완화하고자 제시된 것이 Long Short-Term Memory(LSTM)이다 (Hochreiter and Schmidhuber, 1997a).

LSTM은 input, output, forget 게이트가 존재하며 이들은 셀 상태(cell state)를 어떻게 할 것인지를 결정한다. forget 게이트는 셀 상태로부터 얼마만큼의 정보를 잊을 것인지를 정하는 게이트이다. 시그모이드(sigmoid) 레이어에 의해 결정되며 값이 1에 가까울수록 셀 상태를 보존한다. input 게이트는 입력된 값 중 어떤 것을 셀 상태에 저장할 것인지를 결정한다. 시그모이드 레이어에 의해 어떤 값을 업데이트할 것인지 정해지면 하이퍼볼릭 탄젠트(hyperbolic tangent, tanh) 레이어에 의해 후보 값들이 생성된다. 이후 output 게이트를 통해 셀 상태를 바탕으로 결정된 값을 출력한다 (Hochreiter and Schmidhuber, 1997b). 이러한 구조를 가진 LSTM은 교통 데이터의 속도를 예측하기 위해 적용된 바 있다(Jeon et al., 2020; Zhao et al., 2017). LSTM은 긴 시차에도 기울기가 소실될 확률이 낮다는 장점이 있지만 구조가 복잡하고 학습하는데 상당한 시간이 소요된다. 이 때문에 LSTM의 구조를 단순화한 gated recurrent neural network(GRU)가 제시되었다(Chung et al., 2014). GRU는 LSTM보다 적은 수의 매개변수를 가지고 있어 학습 속도가 LSTM보다 빠르며, 거의 동등한 기능을 한다 (Dai et al 2019; Jeong et al., 2021). 하지만 LSTM과 마찬가지로 GRU와 같은 순환

신경망은 복잡한 구조를 가진 네트워크에서의 통행 속도를 예측하는 것에 한계가 있다. 예를 들어 100개의 구간으로 정의된 네트워크의 정보를 예측하기 위해선 각각 100개의 모델이 필요할 것이다. 이러한 한계를 극복하기 위해 제시된 방식이 합성곱 신경망의 적용이다.

2.2.2 합성곱 신경망

Hubel and Wiesel(1959)은 고양이 실험을 통해 시각 피질의 수많은 뉴런은 국부 수용 영역(local receptive field)이 있으며, 시야 중 일부 시각적 자극에만 반응하는 다는 사실을 확인하였다. 전체 시야는 이러한 수용 영역들이 서로 겹쳐져 이뤄진다는 것과 어떤 뉴런은 직선 이미지에만 반응하는 반면, 특정 각도를 가진 선에만 반응하는 뉴런, 넓은 수용 영역을 가져 낮은 수준의 패턴이 조합되어 복잡한 패턴에 반응하는 뉴런이 존재한다는 사실을 확인하였다. 이 실험은 높은 수준의 뉴런이 낮은 수준을 가진 뉴런의 출력에 기반을 둔다는 아이디어를 얻게 하였다. Fukushima(1979)는 퍼셉트론을 여러 층으로 쌓고 위의 고양이 실험에서 얻은 아이디어를 접목하여 시각적 패턴을 인지하기 위한 첫 합성곱 신경망, neocognitron을 제시하였다. 하지만 neocognitron은 수동으로 가중치를 설정함으로써 성능을 높였는데 이는 수만 개의 변수를 인간이 분석해야 하는 문제가 있었다. 이러한 한계점을 극복한 LeNet(LeCun et al., 1989)은 neocognitron에 학습 데이터와 타겟 데이터를 기반으로 신경망의 가중치를 자동으로 업데이트하는 오차 역전파(backpropagation)(Rumelhart et al., 1985)를 적용한 알고리즘이다. 이후 하드웨어, GPU 병렬처리의 발전으로, LeNet에 병렬처리 방식을 적용하고 활성화 함수(activation function)의 수정을 통해 재구성된 AlexNet을 이용한 팀이 2012년에 개최된 ImageNet Large Scale Visual Recognition Challenge(ILSVRC)에서 우승하면서 딥 러닝의 시대가 도래했다(Krizhevsky et al., 2012). 또한, Simonyan and Zisserman(2014)은 합성곱 신경망의 깊이가 모델의 성과와 관련하여 중요한 요소 중 하나라는 사실을 밝혔고, 2015년 ImageNet 데이터(Russakovsky et al., 2015)에 대한 주요 성과는 모두 깊은 모델을 사용하였다. 하지만 모델의 깊이를 더 깊게 정의한 모델은 오히려 기울기가 소실되어 학습이 제대로 되지 않는다는 문제가 발견되었고, 이를 완화하고자 ResNet이 제안되었다(He et al., 2016). 그림 2.5는 지난 ILSVRC에서 우승한 역대 알고리즘들을 나타낸다.

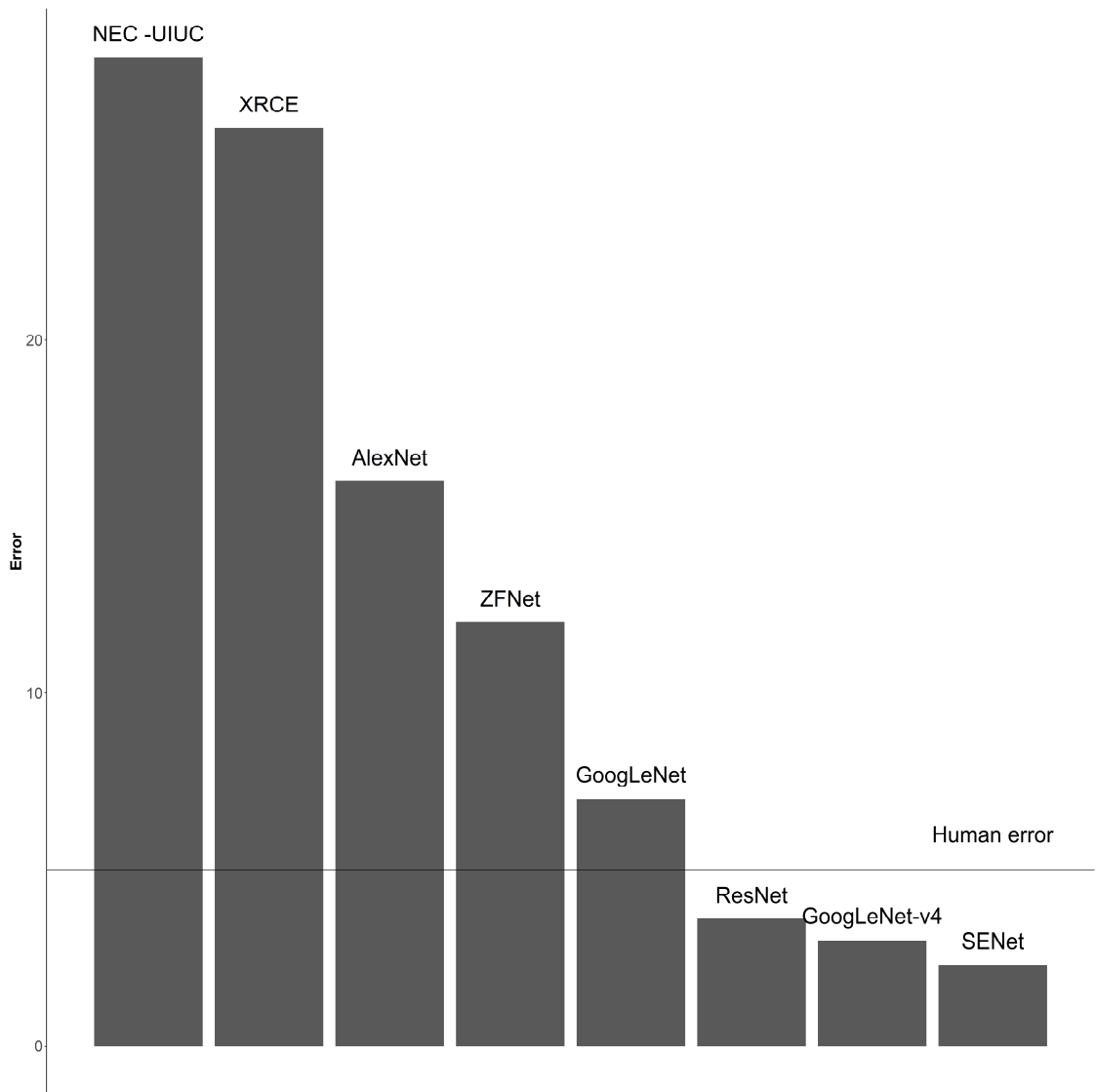


그림 2.5 ILSVRC 우승 결과(Khan et al., 2020)

그림 2.5에서 확인할 수 있는 것과 같이 ResNet은 인간과 가장 비슷한 에러율을 보이는 신경망이다. 또한, 최신 알고리즘 중 가장 큰 폭으로 성능의 향상을 보였다. 본 연구는 이러한 ResNet을 이미지로 변환된 대한민국 대구광역시 지역의 택시 통행 속도 데이터를 전이 학습하여 예측 결과를 제시하고자 한다.

합성곱 신경망의 핵심은 데이터 전체를 보고 판단하는 것이 아닌 부분적으로 특징을 찾아내어 결과를 출력하는 것이다. 이는 합성곱(convolution) 레이어와 pooling 레이어로 구성된다. 그림 2.6은 필터와 합성곱의 동작을 나타낸다.

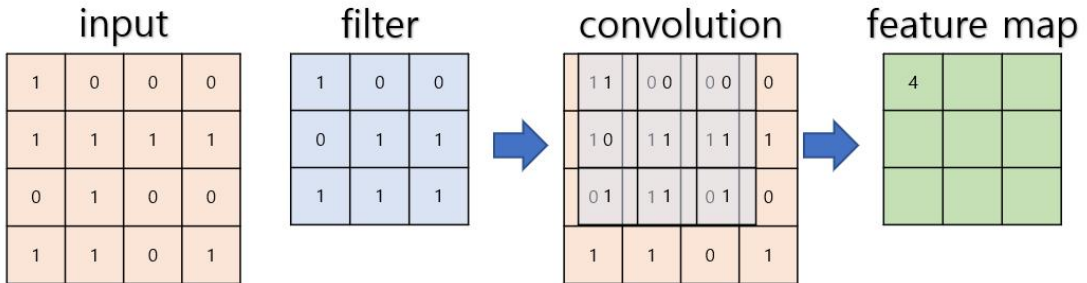


그림 2.6 필터와 합성곱

그림 2.6의 feature map의 각 원소는 입력값과 필터의 합성곱에 의해 채워지며, 필터의 이동 단위는 stride라고 한다. pooling 레이어는 합성곱 레이어의 출력값을 입력받아 feature map의 크기를 줄이거나 특정 데이터를 강조하는 데 사용된다. 처리방법으로는 max pooling과 average pooling, min pooling이 있다. 그림 2.7은 max pooling과 average pooling을 나타낸다.

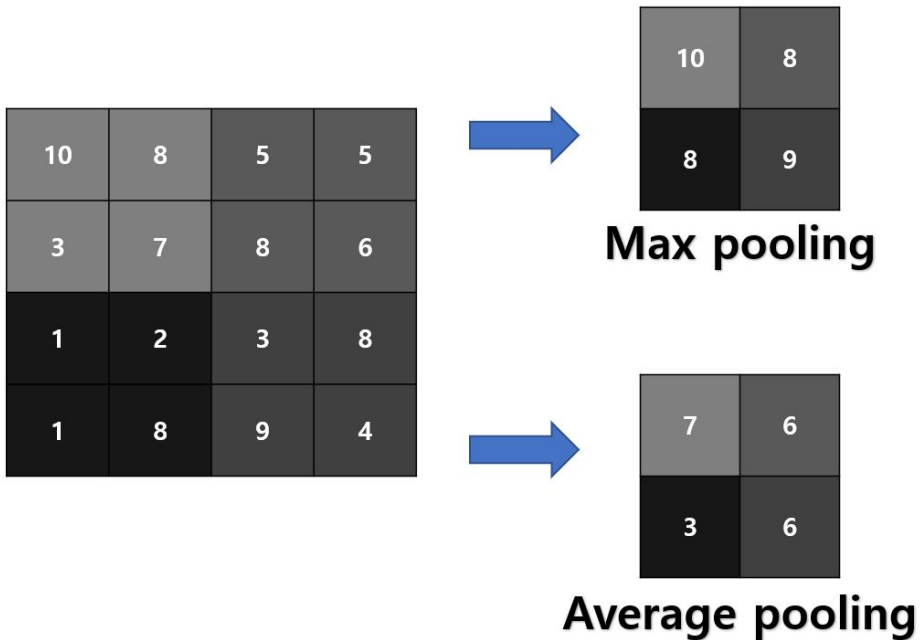


그림 2.7 Max pooling과 average pooling

이미지로 변환된 통행 속도에 합성곱 신경망을 적용하여 이미지마다 특징을 추출하여 예측하고자 하는 도로 네트워크와 시간으로 구성된 이미지로 매핑을 수행하는 방식은 도로 네트워크의 공간적 특성과 시간적 특성을 동시에 고려할 수 있다는 장점이 있다. 또한, 복잡한 구조의 도로 네트워크에서 각각의 도로 구간마다 모델을 정의하지 않아도 되며, 이는 기존의 시계열 분석 방식의 한계를 극복한다. 합성곱 신경망을 이미지로 변환된 통행 속도에 적용하는 연구는 Ma et al.(2017)를 시작으로 하여, 다양한 방식으로 적용되어왔다(Kim et al., 2018; Ren et al., 2018). 본 연구는 기존에 통행 속도 예측을 위해 사용된 Ma et al.'s 모델과 모델의 깊이가 깊어짐에 따라 기울기가 소실되는 문제를 완화하기 위해 제안된 ResNet을 전이 학습하여 대구광역시 도로 네트워크의 평균 속도에 대한 예측을 수행한 후 비교 분석하였다.

제 3장 데이터 및 아키텍처

3.1 데이터 처리 및 분석 아키텍처

본 연구에서 사용된 데이터는 주식회사 이모션(emotion)에서 제공한 택시 이동데이터와 대한민국의 국토교통부에서 제공하는 도로 링크 데이터의 조합으로 구성되었다. 이모션의 택시 포인트 데이터는 주식회사 모빌아이(Mbileye)의 센서를 통해 수집되며, 이는 Controller Area Network(CAN) message로 File Transfer Protocol(FTP) 서버에 적재된다. CAN 통신은 차량 내에서 네트워크 호스트 없이 장치들이 서로 통신하기 위해 설계된 표준 통신 규격이다. 적재된 데이터의 처리를 위해 빅 데이터(big data)를 다루기 위한 클라우드 컴퓨팅(cloud computing) 플랫폼인 아파치 스파크(apache spark)를 아마존 웹 서비스(Amazon Web Service, AWS)의 Elastic MapReduce(EMR)에서 사용하였으며, 결측값(missing value)과 이상값(outlier)의 처리와 CAN message의 파싱(parsing), 데이터의 재구성(reshape)을 아파치 스파크에서 진행하였다. 이후 Elastic Computing(EC2)의 R 환경에서 국토교통부의 도로 링크 데이터를 포인트 데이터와 공간 조인(spatial join)하였고, 모델의 학습을 위한 이미지 데이터로 재구성하였다. 모델의 구현과 학습은 파이썬(python)에서 이루어졌으며, 연구의 전반적인 아키텍처는 그림 3.1과 같다.

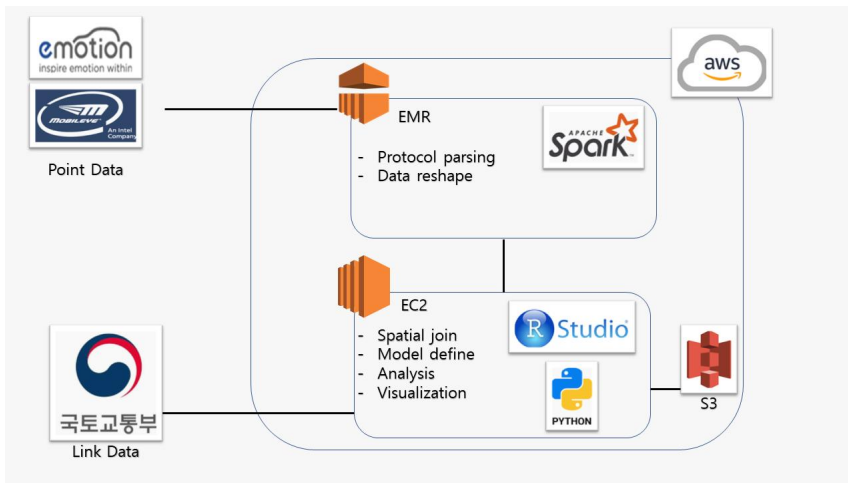


그림 3.1 시스템 아키텍처

본 연구는 도로 네트워크에서의 통행 속도를 예측하기 위해 통행 속도를 시간과 공간으로 표현된 행렬로 재구성하여 이미지로 변환하였다. 하지만 항상 모든 도로 위에서 항상 택시가 운행하지는 않으므로, 행렬을 구성하는 값들은 결측값을 포함한다. 이러한 결측값으로 인해 모델 학습을 위해 생성된 이미지는 결측값에 따라 서로 다른 크기를 갖게 되고, 학습을 위해 적용될 수 없다. 때문에, 본 연구는 이러한 결측값을 처리하는 과정을 포함하였다.

이미지로 변환한 데이터의 결측값은 완전 무작위 결측(missing completely at random)이며, 전체 데이터 중 0.7%를 차지한다. 결측값의 처리는 단순 대치법(simple imputation)의 표준오차의 과소추정, 계산의 난해성의 문제를 보완한 다중 대치법(multiple imputation)을 이용하였다. 다중 대치법은 대치, 분석, 결합의 세 단계를 거친다. 다중 대치법의 대치 단계는 데이터를 생성하는 단계이다. 특정 알고리즘에 따라 결측값을 대체 값으로 변환한 n 개의 데이터 세트를 생성한다. 본 연구에선 데이터의 특성상 Markov Chain Monte Carlo(MCMC)를 사용하였다(Schafer, 1999). MCMC는 Markov chain의 구성에 기반을 둔 확률 분포로부터 원하는 분포의 표본을 추출하는 알고리즘이다. 분석 단계에선 n 개의 완전한 데이터 세트를 분석하고 모수 추정치와 표준오차를 계산한다. 마지막 결합 단계에서는 각 데이터 세트의 결과를 Rubin's rule에 의해 결합한다. 식 1은 결합한 추정치의 분산을 나타낸다(Rubin, 1987).

$$\bar{W} = \frac{1}{m} \sum_{i=1}^m \hat{W}_i, \quad B = \frac{1}{m-1} \sum_{i=1}^m (\hat{Q}_i - \bar{Q})^2, \quad T = \bar{W} + (1 + \frac{1}{m})B \quad (1)$$

여기서 W 는 대치 내 분산(within imputation variance), 대치 간 분산(between imputation variance) B , \bar{Q} 는 각 추정치의 평균을 나타낸다.

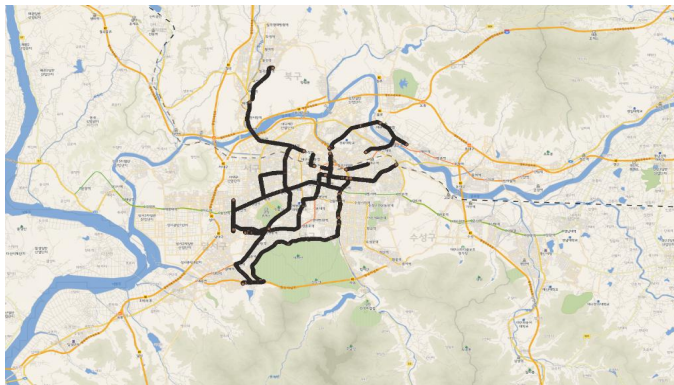
또한, 시퀀스 내 이상값의 처리는 박스 플롯 법칙(box plot rule)(식 2)에 따라 제거하였으며, 데이터 전체 기간 중 각 시퀀스의 표준 편차는 최소 8.447에서 최대 19.267의 값을 가진다(Spear, 1952).

$$D_i > Q_3 + 1.5(Q_3 - Q_1) \vee D_i < Q_1 - 1.5(Q_3 - Q_1) \quad (2)$$

여기서, D_i 는 i 번째 데이터의 값, Q_3 은 제3 사분위 수(upper quartile), Q_1 은 제1 사분위 수(lower quartile)를 나타낸다.

3.2 택시 이동데이터

주식회사 이모션에서 제공된 데이터는 대구 지역의 약 450대의 택시에 설치된 센서로 취득된 포인트 데이터이다. 각 센서는 60~100 millisecond마다 데이터를 CAN message 형식으로 저장한다. 센서의 데이터는 여러 가지 정보를 포함하고 있으나 본 연구에서는 공간정보와 속도, 시간만을 사용하였다. CAN message는 모빌아이에서 제공된 mobileye system standard can output protocol을 참고하여 파싱하였다. 포인트 데이터의 시간적 범위는 2020년 10월 2일 00시 00분부터 2021년 1월 2일 23시 59분까지 약 3달(92일)이다. 이는 약 500GB(44억 rows) 정도이다. 그림 3.2는 처리된 포인트 데이터의 일부와 확대된 모습이다.



(a) 축척 1/100,000



(b) 축척 1/3,000

그림 3.2 포인트 데이터 시각화

3.3 도로 링크 데이터

도로의 구간별로 교통 데이터를 예측하기 위해 포인트 데이터와 도로 링크 데이터를 공간 조인하여 포인트 데이터에 도로의 ID를 부여하였다. 도로 링크 데이터는 대구 지역의 링크 데이터(17,945 객체)만을 사용하였으며, 통행량별로 구분하여 모델 정의를 위한 네트워크의 규모를 결정하였다. 연구에 사용된 링크 데이터는 그림 3.3과 같다.

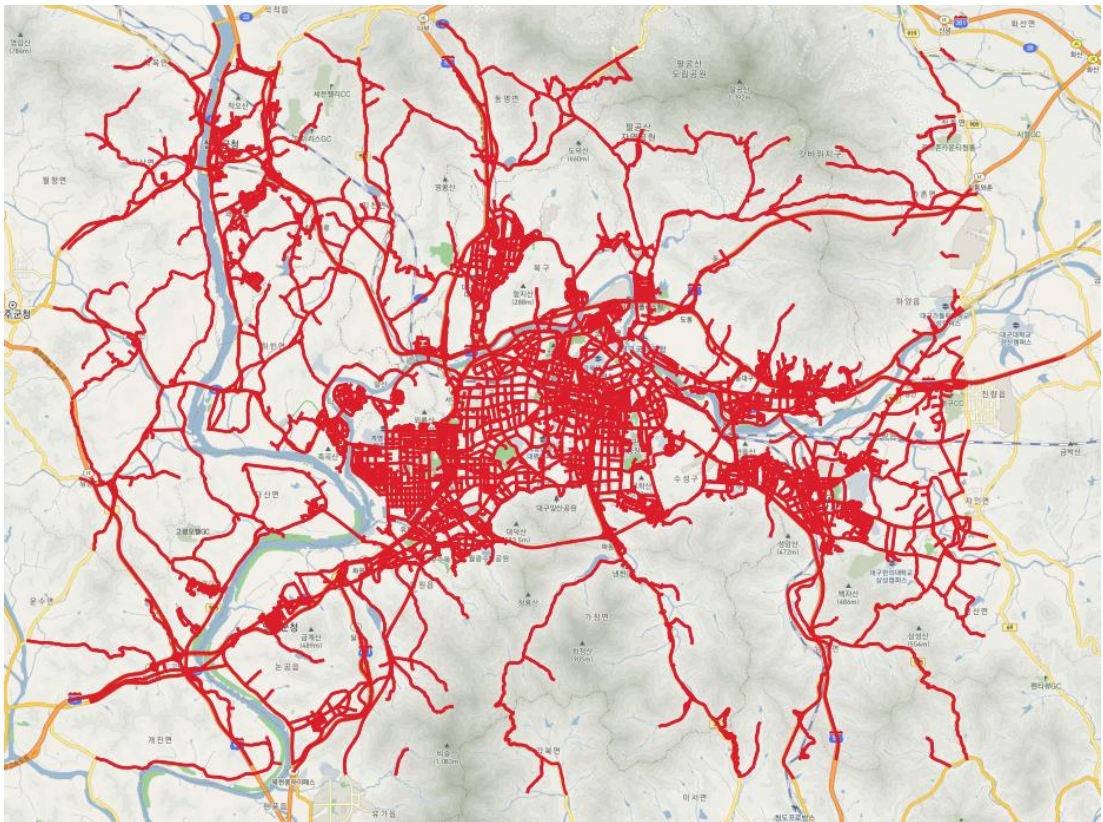


그림 3.3. 도로 링크 데이터 시각화

3.4 공간 조인

포인트 데이터와 링크 데이터의 조합을 위해 공간 조인을 수행하였다. 모바일의 포인트 데이터와 국토교통부의 링크 데이터는 둘 다 높은 정확도의 데이터지만, 국토교통부의 도로 링크 데이터가 도로의 폭을 반영한 폴리곤 형태의 데이터가 아니라 라인 데이터이므로, 공간 조인을 할 때 포인트 데이터에 버퍼(buffer)를 주어 접하는 링크 데이터와 속성을 조합하는 방식을 채택하였다. 그림 3.4는 연구에서 사용된 공간 조인 방식을 설명한다.

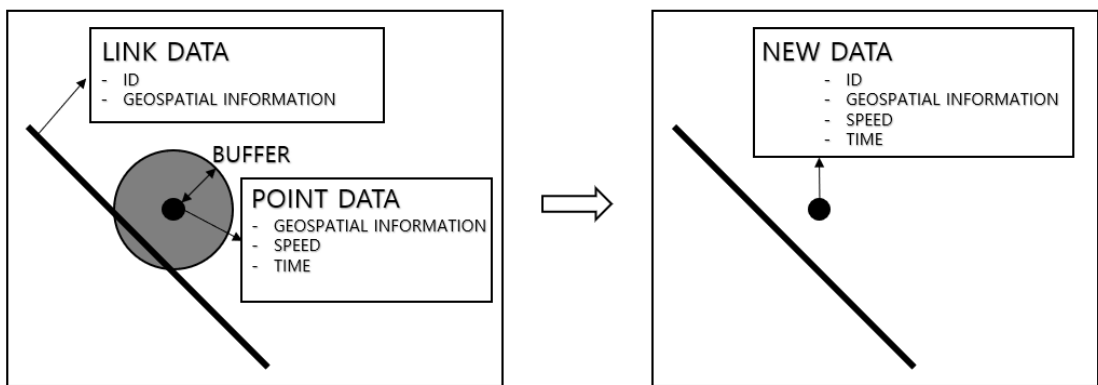


그림 3.4 위치에 따른 속성값 조인: 버퍼

그림 3.4의 포인트 데이터는 택시에 설치된 센서로 취득된 데이터를 의미한다. 포인트 데이터에 버퍼를 주어 접하게 되는 링크 데이터의 ID를 포인트 데이터에 부여함으로써 각각의 포인트 데이터는 도로에 속하게 된다. 이러한 방식의 데이터 결합은 버퍼의 크기에 따라 데이터의 소실 정도가 결정된다. 포인트 데이터에 주워진 버퍼의 범위 내에 접하는 링크가 없는 경우 포인트 데이터는 도로의 ID를 부여받지 못하게 되어 제거된다. 본 연구에서는 포인트 데이터 중 몇몇 표본을 추출한 뒤 버퍼의 길이를 다르게 줌에 따라 최대한 포인트 데이터를 보존하는 방향으로 버퍼의 길이를 결정하였다. 그림 3.5는 무작위로 뽑은 데이터 집합(set)에 대해 버퍼 크기에 따라 공간 조인을 수행한 후의 데이터와 원본 데이터의 양을 비교하여 나타낸다. 버퍼의 길이가 1 미터(group a)에서 10 미터(group d)로 길어짐에 따라 공간 조인 후 남는 데이터의 양을 확인할 수 있다.

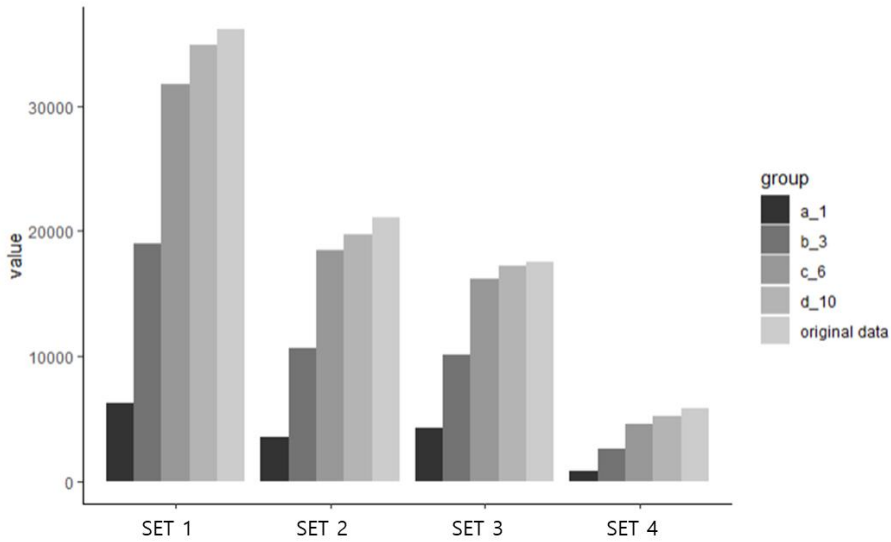


그림 3.5 버퍼 길이에 따른 데이터양의 비교

3.5 이미지 데이터

포인트 데이터와 링크 데이터의 조합으로 생성한 데이터를 이용하여 도로 네트워크의 교통 데이터 예측을 수행하기 위해선, 합성곱 신경망을 적용할 수 있는 형태로 데이터를 재구성해야 한다. 합성곱 신경망의 일반적인 입력값은 이미지의 형태이며 이는 그림 3.6과 같다.

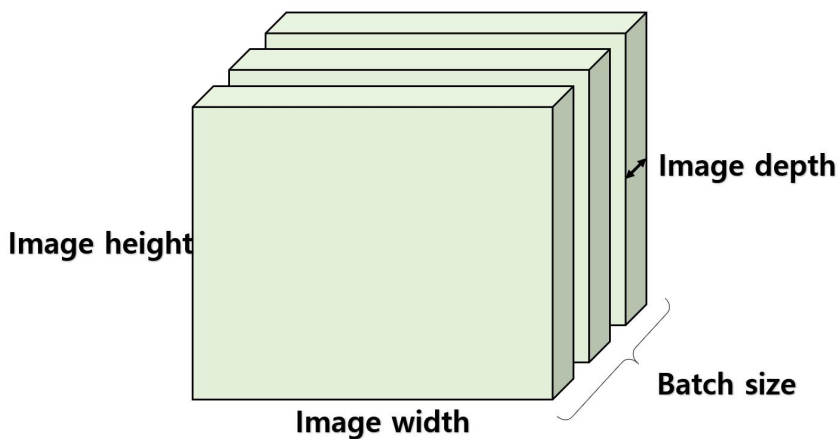


그림 3.6. 합성곱 신경망의 입력 데이터 형태

본 연구에서는 교통 데이터를 시간과 공간으로 표현되는 이미지로 변환하여 모델 학습에 사용하였으며 이미지의 형태는 고려하고자 하는 도로 네트워크의 규모와 시간적 범위, 예측하고자 하는 시간의 단위에 따라 다르게 정의된다. 그림 3.7은 모델 학습을 위한 이미지 데이터의 정의 방식을 나타낸다.

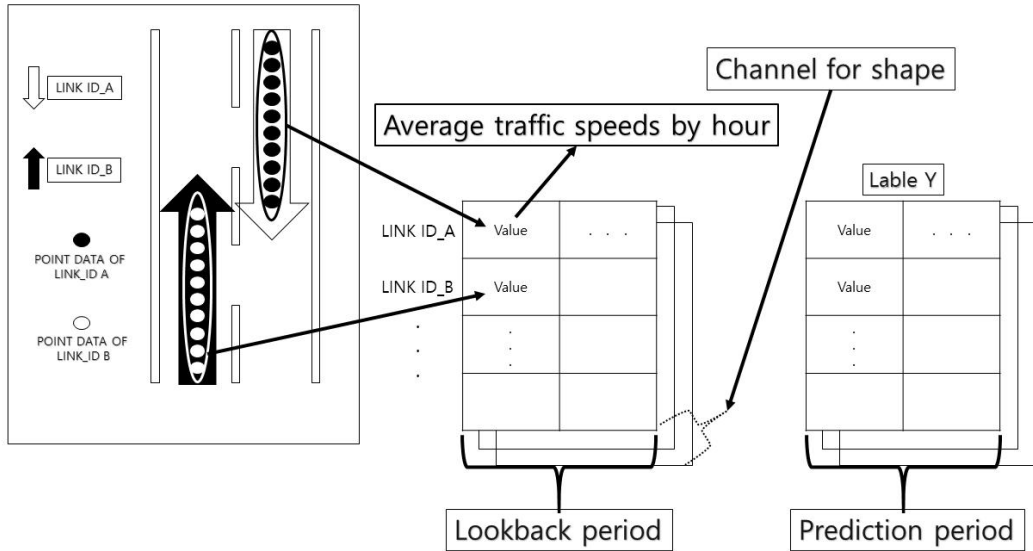


그림 3.7 이미지 데이터 생성 과정

도로 네트워크를 구성하는 각각의 도로 링크의 ID는 이미지 행렬의 y축을 이루고 x축은 시간을 나타낸다. 타겟 혹은 레이블로 불리는 행렬의 x축은 예측하고자 하는 시간의 범위로 정의한다. 예를 들어 1시간 후 네트워크에서의 통행 속도를 예측할 때, 하루 동안의 교통 상황을 고려하여 예측하고자 한다면, 왼쪽의 lookback period는 24가 될 것이고 오른쪽의 prediction period는 1이 될 것이다. 또한, 행렬의 각 셀값은 ID를 부여받은 시퀀스의 평균을 나타낸다. n 대의 택시가 x 시간에 y 의 도로에서 운행했다면, 행렬 (x, y)의 셀값은 n 대 택시의 평균 속도이다. 이러한 방식으로 시간과 공간으로 표현된 교통 데이터의 행렬은 이미지로 변환될 수 있으며, 합성곱 신경망을 적용할 수 있게 된다. 이미지의 채널(channel)은 전이 학습 시 초기화되는 가중치에 따라 1 또는 3으로 설정하였으며 그림 3.8과 그림 3.9는 변환된 데이터의 이미지를 나타낸다.

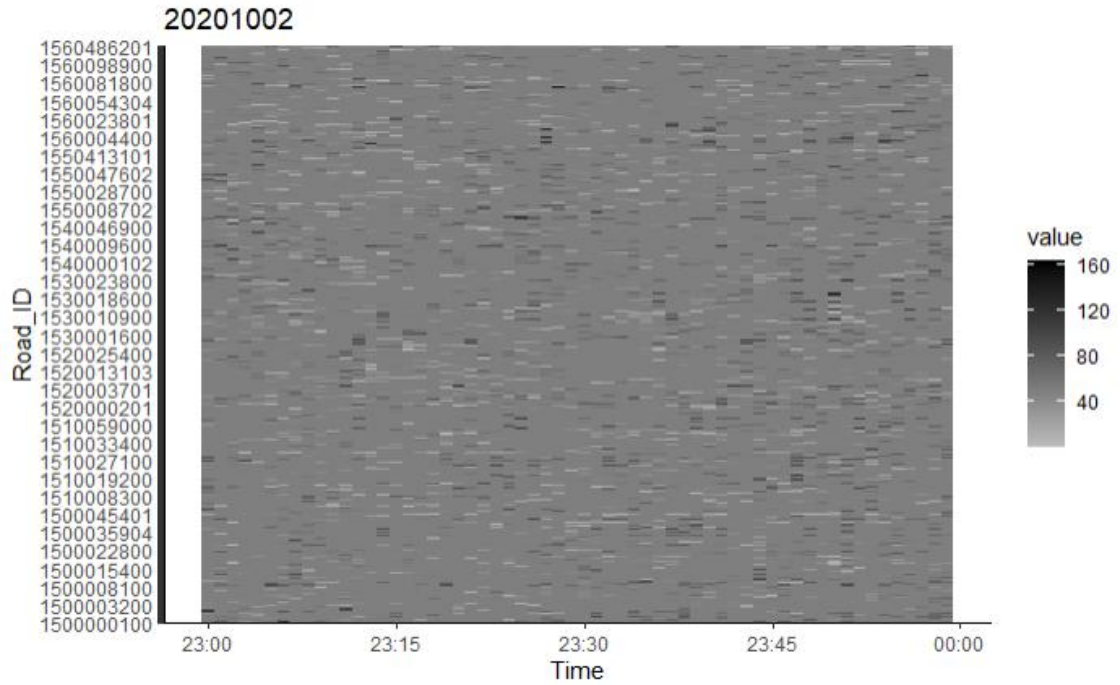


그림 3.8 도로 네트워크 통행 속도의 시간 단위 이미지 데이터

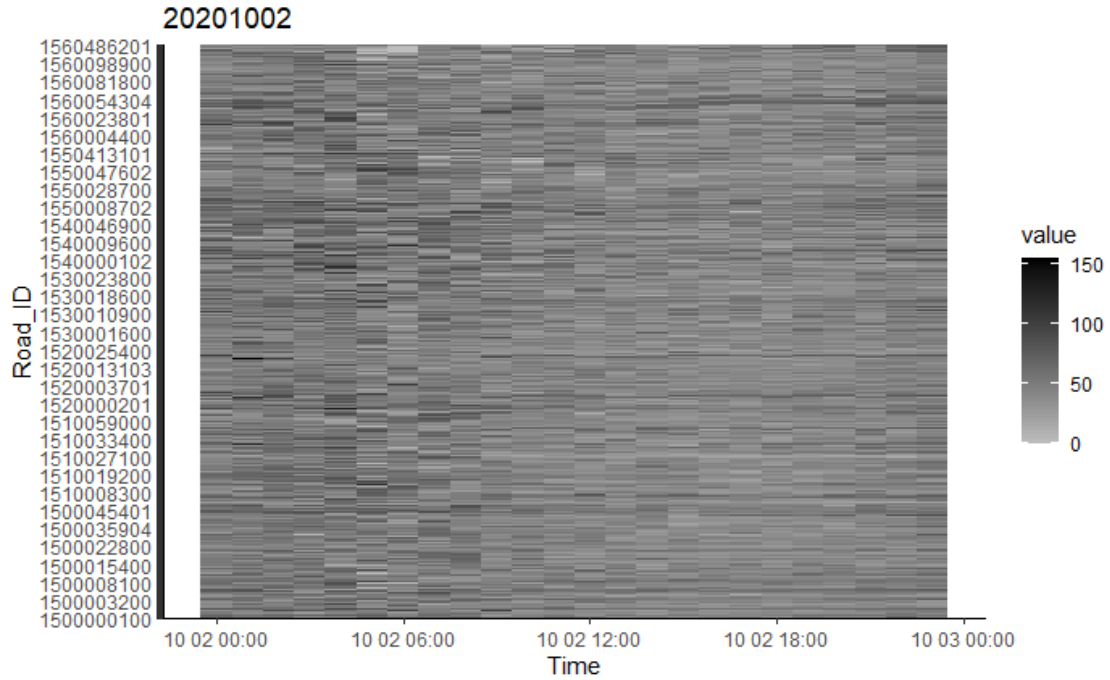


그림 3.9 도로 네트워크 통행 속도의 일 단위 이미지 데이터

3.6 도로 네트워크 선택

국토교통부에서 제공된 데이터 중 대구 지역을 위주로 추출한 도로 링크 데이터의 객체 수는 17,945개이다. 하지만 이모션에서 제공된 이동 객체의 포인트 데이터와 공간 조인을 한 결과 도로 링크 데이터보다 수집된 포인트 데이터의 범위가 상대적으로 좁아 도로 대부분은 운행 기록이 없는 것으로 확인되었다. 그림 3.10은 도로마다 중복되지 않은 이동 객체의 일별 통행량의 누적된 값을 히스토그램으로 표현한 것이다.

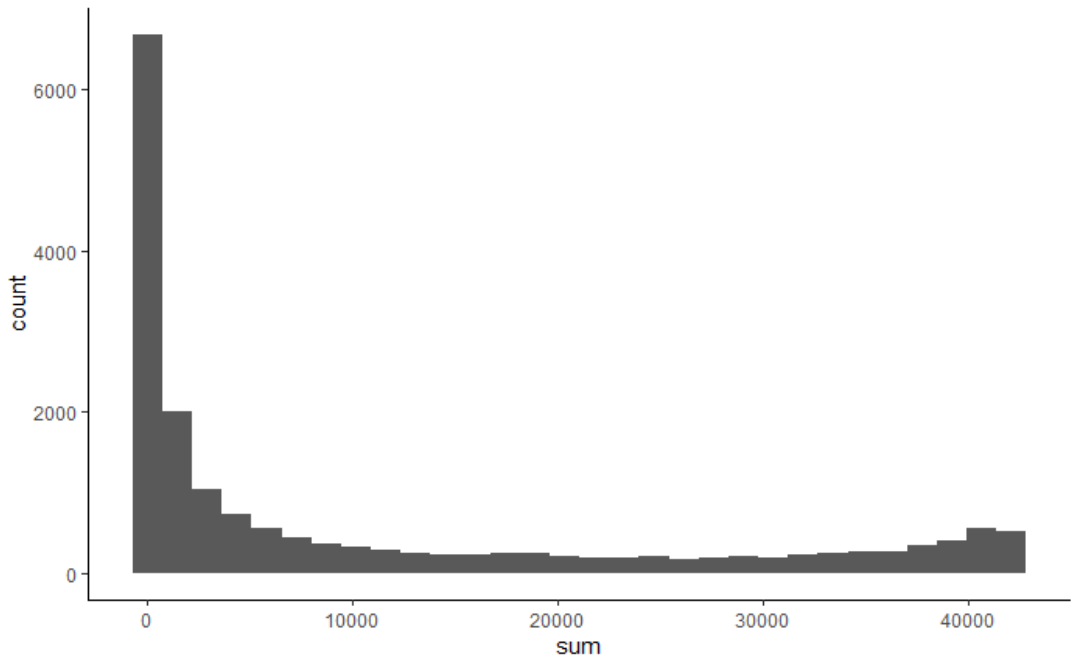
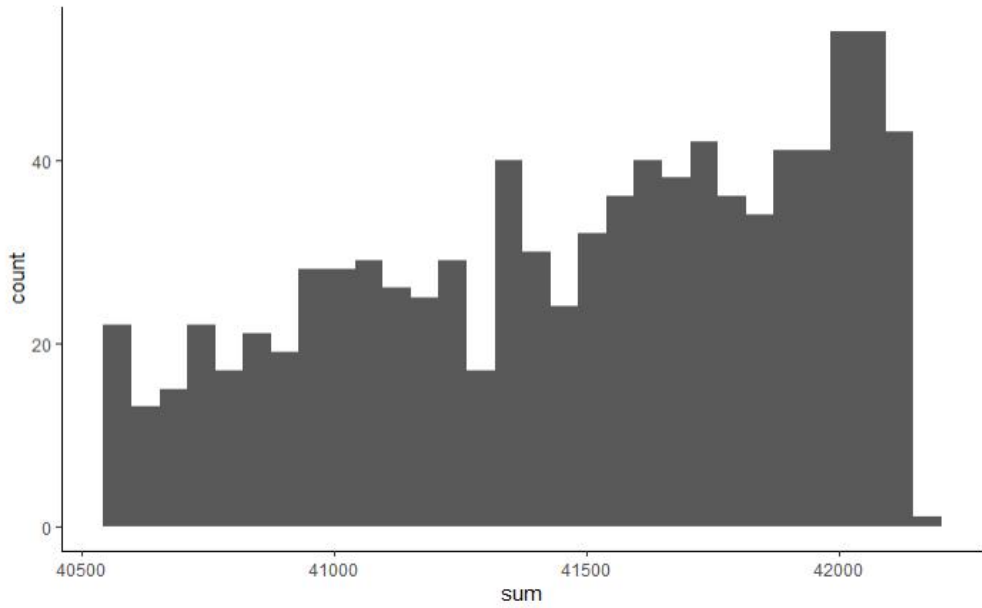
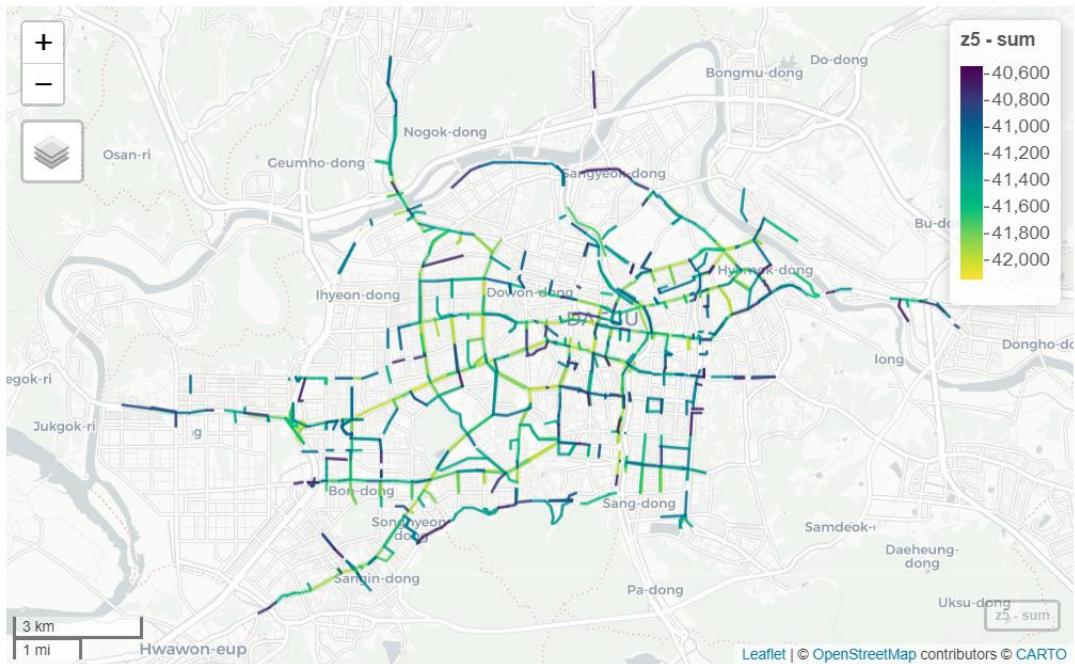


그림 3.10 전체 도로의 누적 사용 횟수

모든 도로를 예측 모델의 학습 데이터로 사용하는 경우, 불필요한 계산이 많아지므로 본 연구는 17,945개의 도로 중 이용 빈도수의 상위 5%의 도로만을 추출하여 연구를 진행하였다. 그림 3.11은 각각 퍼센트에 대한 히스토그램과 도로의 모습이며, 897개의 객체이다.



(a) 이용 빈도수 상위 5%에 대한 히스토그램



(b) 이용 빈도수 상위 5%에 대한 도로 네트워크

그림 3.11 이용 빈도수 상위 5% 도로

제 4장 모델 구조 및 하이퍼 파라미터

4.1 모델의 구조

4.1.1 잔차 학습

최근 딥 러닝은 여러 분야에서 많이 활용되고 있다(Jeon et al., 2020; Than et al., 2020). 하지만 딥 러닝은 기울기 소실 또는 증폭(exploding)과 같은 문제가 존재하며 이는 모델의 성능에 치명적인 영향을 끼친다. 이 때문에 기울기 소실 문제에 대한 해결을 위해 많은 방법이 제시되었다. ResNet은 모델의 깊이가 깊어짐에 따라 발생하는 기울기 소실 문제를 완화하기 위해 제시되었다. 모델의 깊이는 모델의 성능에 있어서 중요한 요소 중 하나이다(Szegedy et al., 2015). 하지만 모델의 깊이가 깊어지게 되면 오차 역전파를 진행함에 따라 미분의 횟수가 많아져 앞의 레이어일 수록 미분값이 작아져 그만큼 출력값에 영향을 미치는 가중치도 작아진다. 이는 과적합(overfitting)과는 별개의 문제이다. ResNet은 2015년 Simonyan and Zisserman에 의해 제시된 Visual Geometry Group(VGG)과 비슷한 구조로, shortcut을 설치함으로써 더 깊은 구조를 가능하게 만든 것이다. 기존의 신경망은 입력값을 타겟값으로 매핑하는 함수 $H(x)$ 를 찾는 것이 목표였다. 즉, $H(x) - y$ 를 최소화하는 방향이다. 하지만 ResNet은 기존의 모델과는 다른 방식을 추구한다. ResNet은 잔차를 학습하는 것이 기존의 방식보다 최적화하기 더 쉽다는 가정에서 비롯된다. 즉, 네트워크의 출력값이 x 가 되도록 $F(x) = H(x) - x$ 를 최소화하는 방향이다. 이러한 $F(x)$ 를 잔차라고 하며, 이 잔차를 학습하는 것이 잔차 학습이다. 또한, 잔차 학습은 출력값($H(x) = F(x) + x$)을 미분하였을 때 $F'(x) + 1$ 이 되게 함으로써 최소한 1의 기울기를 가지게 한다.

또한, ResNet은 convolution block과 함께 구성된다. identity block은 $F(x)$ 에 x 를 그대로 더하는 것이고 convolution block은 x 역시 convolution 연산을 거친 후에 더해주는 것이다. ResNet은 이 두 가지 block을 쌓아서 구성한다. 그림 4.1은 identity block과 convolution block을 나타내며, 그림 4.2는 block으로 표현된 ResNet-50의 기본 구조를 나타낸다.

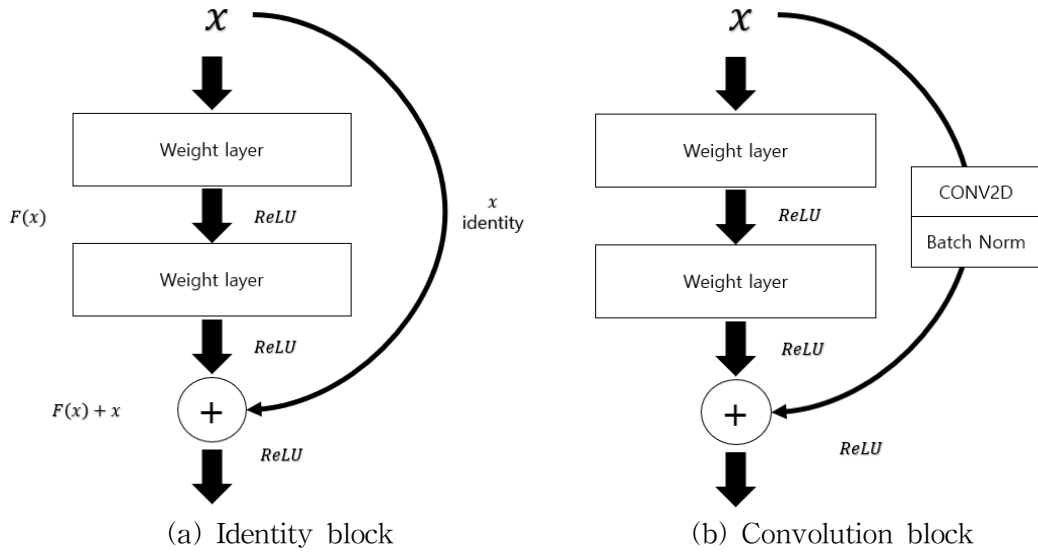


그림 4.1 잔차 학습: a building block(He et al., 2016)

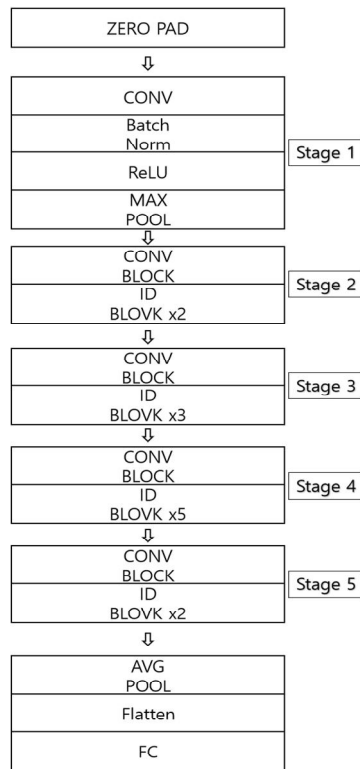


그림 4.2. Identity block과 convolution block으로 구성된 ResNet-50

4.1.2 Batch normalization

ResNet은 또한 batch normalization 레이어를 사용한다. batch normalization은 활성화 함수 중 시그모이드를 ReLU 및 하이퍼볼릭 탄젠트로 바꾸거나 작은 학습률(small learning rate)을 주는 방식은 간접적이며, 직접 학습 과정 자체의 안정화를 위해 제안되었다. 이는 기울기 소실/증폭의 원인이 internal covariate shift에 있다는 가설을 기반으로 제안되었다. internal covariate shift는 네트워크의 각 층이나 네트워크마다 입력값의 분포가 달라지는 현상을 말한다.

이러한 현상을 완화하고자 처음 사용된 방법이 whitening이다. 이는 입력값의 분포를 평균은 0으로, 분산은 1로 정규화(normalization)하는 방법이다. whitening은 기본적으로 입력값의 feature들을 비상관화(decorrelation)하며, 각각의 분산을 1로 만든다. 하지만 whitening은 공분산 행렬(covariance matrix)의 계산과 역행렬의 계산이 필요하며 일부 파라미터의 영향을 무시할 가능성 또한 존재한다. 이 때문에 제안된 방식이 batch normalization이다. batch normalization은 각 feature는 이미 비상관화되어 있다고 가정하고 각각의 feature들에 대해서 스칼라(scalar) 형태로 평균을 구한 뒤 분산을 계산하여 정규화를 거친다. 단순히 평균(식 3)과 분산(식 4)을 0과 1로 바꾸는 것은 오히려 활성화 함수의 비선형성(nonlinearity)을 없앨 수 있다.

$$\mu_B = \frac{1}{m} \sum_{i=1}^m x_i \quad (3)$$

여기서, x 는 mini-batch에 대한 값, $B = \{x_{1...m}\}$ 을 나타낸다.

$$\sigma_B^2 = \frac{1}{m} \sum_{i=1}^m (x_i - \mu_B)^2 \quad (4)$$

예를 들어 시그모이드 함수에서 입력값의 평균이 0이고 분산이 1이라면 결괏값은 직선에 가까울 것이다. 이를 보완하기 위해 식 5와 같이 정규화된 값에 식 6과 같이 scale factor(γ)와 shift factor(β)를 더해주고 이 변수들을 오차 역전파 과정에서 같이 학습시켜준다(loffe and Szegedy, 2015). 또한, batch normalization은 평균과 분산 등의 통계를 학습 데이터 전체에서 구하는 것이 아닌 mini-batch 단위로 접근하여 계산한다.

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \quad (5)$$

여기서, ϵ 은 수치 안정성(numerical stability)을 위해 mini-batch 분산에 추가된 상수를 나타낸다.

$$y_i = \gamma \hat{x}_i + \beta \equiv BN_{\gamma, \beta}(x_i) \quad (6)$$

여기서, γ 는 scale factor, β 는 shift factor를 나타낸다.

실제 batch normalization을 네트워크에 적용할 때는 특정 은닉층에 들어가기 전에 batch normalization 레이어를 더하여 입력값을 수정한 뒤 활성화 함수에 넣어주는 방식으로 사용된다. 합성곱 신경망에서 적용될 때는 편향을 값이 대체한다. 또한, 채널 별 batch normalization 변수를 생성한다. 예를 들어 개의 mini-batch와 개의 채널이 있을 때 합성곱(convolution) 이후 $p \times q$ 의 feature map이 생성되면 $m \times p \times q$ 개의 평균과 분산을 구한 후 채널당 1개씩의 γ 와 β 를 총 n 개 생성할 수 있다. 또한, batch normalization은 순전파(propagation) 과정에서 파라미터 scale에 영향을 받지 않기 때문에 학습률(learning rate)을 크게 잡을 때 파라미터의 scale 때문에 기울기가 소실 혹은 증폭되는 문제를 완화하고 이러한 특성은 학습 속도의 가속으로 이어진다. batch normalization의 경우 자체적 regularization 효과가 있으므로 기존의 weight regularization을 제외하는 것이 가능하고 드롭아웃(drop out) 또한 제외할 수 있게 한다.

4.1.3 모델 아키텍처

본 연구는 잔차 학습으로 구성된 ResNet중 ResNet-50과 ResNet-152를 베이스 모델로 하여 전이 학습한 후 예측을 수행하였다. 각각의 입력 레이어와 출력 레이어는 예측하고자 하는 도로 네트워크의 개수에 따라 수정하였다. 본 연구에서 제안한 모델과 비교 분석을 위한 Ma et al.'s 모델의 구조는 표 4.1과 같다. ResNet-50의 경우 기본적인 구조는 표 4.2와 같으나, 가중치를 고정(freeze) 레이어의 개수에 따라 나뉜다(모델 1). 모델 2는 ResNet-152를 전이 학습하여 구축하였으며, 모델 1과 마찬가지로 가중치를 고정된 구간에 따라 나뉜다.

표 4.1 비교 분석을 위한 Ma et al. 's 모델 구조

layer name	layer	layer
input	-	-
layer 1	convolution	filter(256, 3, 3)
	pooling	pooling(2, 2)
layer 2	convolution	filter(128, 3, 3)
	pooling	pooling(2, 2)
layer 3	convolution	filter(64, 3, 3)
	pooling	pooling(2, 2)
Flatten		
Fully-connected		

표 4.2 Model 1: ResNet-50

layer name	output size	layer
conv1	112×112	$7 \times 7, 64, stride\ 2$
conv2	56×56	$3 \times 3\ max\ pooling, stride\ 2$
		$\begin{pmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{pmatrix} \times 3$
conv3	28×28	$\begin{pmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{pmatrix} \times 3$
conv4	14×14	$\begin{pmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{pmatrix} \times 3$
conv5	7×7	$\begin{pmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{pmatrix} \times 3$
1×1		global average pooling 2D
Flatten		
Fully-connected		

ResNet-50은 표 3에서 확인할 수 있는 것과 같이 5단계로 나뉘어 있다. ResNet-50과 ResNet-152의 차이는 conv3의 block을 4개에서 8개로, conv4의 block은 6개에서 36개로 늘렸다는 점이다. ResNet은 처음에 7x7, 3x3 사이즈의 필터로 합성곱과 max pooling을 수행한다. 이후 입력의 크기를 기준으로 단계를 구분한다. 연산이 stride 2로 수행됨으로 이미지의 높이와 너비는 절반으로 줄어들며, 단계마다 필터의 개수가 늘어 채널의 개수는 두 배가된다. 즉, 한 단계에서 다른 단계로 진행함에 따라 채널의 개수는 두 배가되고 입력 크기는 절반으로 줄어드는 것이다. 또한, ResNet은 깊이에 따라 다른 block을 사용한다. 이것은 ResNet-34와 ResNet-50/101/152의 차이이며, 그림 4.3은 서로 다른 두 block을 나타낸다.

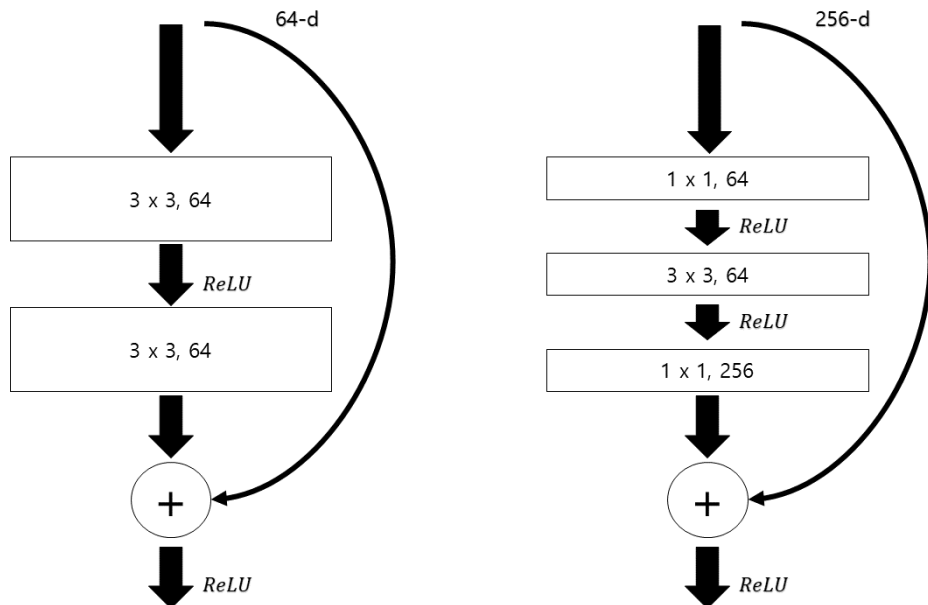


그림 4.3 ResNet-34를 위한 building block(왼쪽 이미지)과 ResNet-50/101/152를 위한 “bottleneck” building block(오른쪽 이미지)(He et al., 2016)

ResNet-50/101/152 모델은 bottleneck이라고 하는 block을 사용한다. 이러한 개념은 깊이가 더 깊어짐에 따라, 학습에 걸리는 시간을 고려하여 도입되었다. 그림 26에서의 bottleneck block은 256차원의 입력값을 1x1 합성곱 레이어를 이용하여 64차원으로 감소시킨 뒤 필터의 개수를 뒤에서 늘림에 따라, 다시 차원을 증가시킨다. 이는 3 x 3 합성곱 레이어를 곧바로 연결한 구조에 비해 연산량을 감소시킬 수 있다.

4.2 모델 하이퍼 파라미터 및 학습 과정

4.2.1 모델 하이퍼 파라미터

딥 러닝은 식 7(Cauchy, 1847)과 같은 원리로 가중치를 업데이트하며, 이를 mini-batch 단위로 적용한 것이 확률적 경사 하강법(stochastic gradient descent, SGD)이다.

$$w^+ = w - \eta \frac{\partial E}{\partial w} \quad (7)$$

여기서, w^+ 는 업데이트된 가중치, w 는 현재의 가중치, η 는 학습률, E 는 손실 함수의 결과를 나타낸다.

이때 사용되는 손실 함수와 최적화 함수에 따라 학습 성능이 달라진다. 본 연구에서 사용된 손실 함수는 mean absolute error(MAE)이며 식 8과 같이 정의된다(Willmott and Matsuura, 2005).

$$MAE = \frac{\sum_{i=1}^n |y_i - x_i|}{n} = \frac{\sum_{i=1}^n |e_i|}{n} \quad (8)$$

여기서, y 는 참값, x 는 예측된 값을 나타낸다.

최적화 함수는 adaptive momentum estimation(Adam)이며 momentum과 root mean square propagation(RMSprop)을 합친 기법이다. momentum은 확률적 경사 하강법을 통해 이동하는 과정 중 관성을 주는 것, 즉 이전의 학습 결과를 반영하는 것이다. 식 9는 momentum을 나타낸다.

$$\begin{aligned} \Delta w &= \alpha w - \eta \frac{\partial E}{\partial w} \\ w^+ &= w + \Delta w \end{aligned} \quad (9)$$

여기서, α 는 momentum term을 나타낸다(default: 0.9).

momentum의 특징은 batch마다 학습으로 인해 가중치의 값이 크게 변동하는 것을 방지하기 위해 momentum term을 이용하여 이전까지의 학습 결과를 더 크게 반영하는 것이다. RMSprop는 AdaGrad를 개선한 방식이다. AdaGrad는 학습률이 작으면 학습 시간이 길어지고, 너무 크면 발산해서 학습이 제대로 이루어지지 않는 문제를 학습을 통해 크게 변동이 있었던 가중치에 대해서는 학습률을 감소시키고 변동이 별로 없

있던 가중치는 학습률을 증가시킴으로써 해결한다. 식 10은 AdaGrad를 나타낸다 (Duchi et al., 2011).

$$\begin{aligned} h &:= h + \frac{\partial E}{\partial w} \circ \frac{\partial E}{\partial w} \\ w^+ &= w - \eta \frac{1}{\sqrt{h}} \frac{\partial E}{\partial w} \end{aligned} \tag{10}$$

하지만 AdaGrad는 학습 도중 어느 순간 h 가 너무 커져서 학습이 이루어지지 않을 수 있다. 이를 보완한 방식이 RMSprop이다. AdaGrad는 단순한 볼록 함수(convex function)에서 정상적으로 작동하지만, 복잡한 다차원 함수에서는 극점에 도달하기 전에 학습률이 0에 수렴할 수 있다. RMSprop는 이를 보완하기 위해 가중치의 기울기를 단순 누적시키는 것이 아니라 최신의 기울기 값을 더 반영하도록 한다. 식 11은 RMSprop의 h 를 나타낸다(Tieleman and Hinton, 2012).

$$h^+ = \rho h + (1 - \rho) \frac{\partial E}{\partial w} \circ \frac{\partial E}{\partial w} \tag{11}$$

momentum에서와 반대로 ρ 가 작을수록 최근의 기울기를 더 크게 반영한다. 이전에 큰 변동이 존재했다라도 현재의 batch에 대한 기울기가 완만하다면 학습률을 높인다. 이는 극솟값에 근접할수록 학습 속도가 느려지는 문제와 local minimum에 빠지는 문제를 완화하며 h 의 값이 무한히 커지지 않게 한다. 하지만 momentum과 RMSprop는 각각의 v, h 가 0으로 초기화되면 학습 초반에 가중치가 0으로 편향되는 문제가 있다. 이러한 문제를 해결하기 위해 제시된 것이 Adam이며, 그림 4.4는 Adam의 의사코드(pseudocode)를 나타낸다(Kingma and Ba, 2014).

```

gt2 indicates the elementwise square gt ⊙ gt.
Good default settings for the tested machine learning problems are α = 0.001, β1 = 0.9, β2 = 0.999 and ε = 10-8


---


Require: α : Stepsize
Require: β1, β2 ∈ [0,1) : Exponential decay rates for the moment estimates
Require: f(θ) : Stochastic objective function with parameters θ
Require: θ0 : Initial parameter vector
m0 ← 0 (Initialize 1st moment vector)
v0 ← 0 (Initialize 2nd moment vector)
t ← 0 (Initialize timestep)
while θt not converged do
  t ← t + 1
  gt ← ∇θ ft(θt-1) (Get gradients w.r.t. stochastic objective at timestep t)
  mt ← β1 · mt-1 + (1 - β1) · gt (Update biased first moment estimate)
  vt ← β2 · vt-1 + (1 - β2) · gt2 (Update biased second raw moment estimate)
  m̂t ←  $\frac{m_t}{(1-\beta_1^t)}$  (Compute bias - corrected first moment estimate)
  v̂t ←  $\frac{v_t}{(1-\beta_2^t)}$  (Compute bias - corrected first moment estimate)
  θt ← θt-1 - α ·  $\frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$  (Update parameters)
end while
return θt (Resulting parameters)

```

그림 4.4 Adam 최적화 함수의 pseudocode(Kingma and Ba, 2014)

기존 momentum에서는 m_0 가 0으로 초기화되면 $m_1 = 0.1g_1$ 이므로 0으로 편향된다. 그러나 Adam에서는 $\frac{m_1}{(1-\beta_1)}$ 로 m 을 업데이트하기 때문에 $m_1 = g_1$ 이 되어 0으로 편향을 해결할 수 있다. v 역시 마찬가지이며 최종적으로 RMSprop 방식에 momentum을 곱하여 파라미터를 업데이트한다. 실제 실험에서 사용된 Adam의 학습률은 추천되는 값인 0.001로 설정하였으며, batch size는 60으로 고정하여 실험을 진행하였다.

4.2.2 학습 데이터 및 과정

실험에 사용된 학습 데이터는 이미지로 변환된 도로 네트워크의 교통 데이터이며, 이는 고려된 네트워크의 규모, 시차에 따라 식 12와 같이 정의된다.

$$X_S = \begin{pmatrix} x_{1,1} & \cdots & x_{1,N} \\ \vdots & \ddots & \vdots \\ x_{M,1} & \cdots & x_{M,N} \end{pmatrix} \quad (12)$$

여기서, x_{mn} ($m = 1, \dots, M$, $n = 1, \dots, N$, $s = 1, \dots, S$)는 n^{th} 의 time step에서 도로 구간의 평균 속도, S 는 샘플의 개수를 나타낸다.

본 연구에서 사용된 M 의 값은 897이며 각각 도로 네트워크의 규모를 나타낸다. N 의 값은 24이며, 이는 한 시간 예측을 위해 고려한 시간이 하루임을 뜻한다. S 는 학습 데이터의 샘플 개수를 나타내며 전체 학습 데이터의 길이에서 $N \times M$ 을 뺀 후 M 으로 나눈 값과 같다. 또한, 연구에 사용된 방법은 지도 학습(supervised learning)이므로 학습 데이터에 대한 레이블을 식 13과 같이 정의한다.

$$Y_S = \begin{pmatrix} x_{1,N+1} \\ \vdots \\ x_{M,N+1} \end{pmatrix} \quad (13)$$

여기서, Y_s ($s = 1, \dots, S$)는 학습 데이터 중 s^{th} 의 레이블을 나타낸다.

연구에 사용된 데이터는 2020년 10월 2일 00시 00분부터 2021년 1월 2일 23시 59분까지 총 92일이며, 80%(74일)는 학습 데이터로 사용하였다. 학습 데이터 중 20%(약 15일)는 validation data로 사용하였으며, hold-out 방식을 이용하였다. 그림 4.5는 학습 데이터, validation dataset, test dataset의 구분과 샘플, time steps에 따른 데이터 정의 등 전반적인 데이터의 형태를 나타낸다.

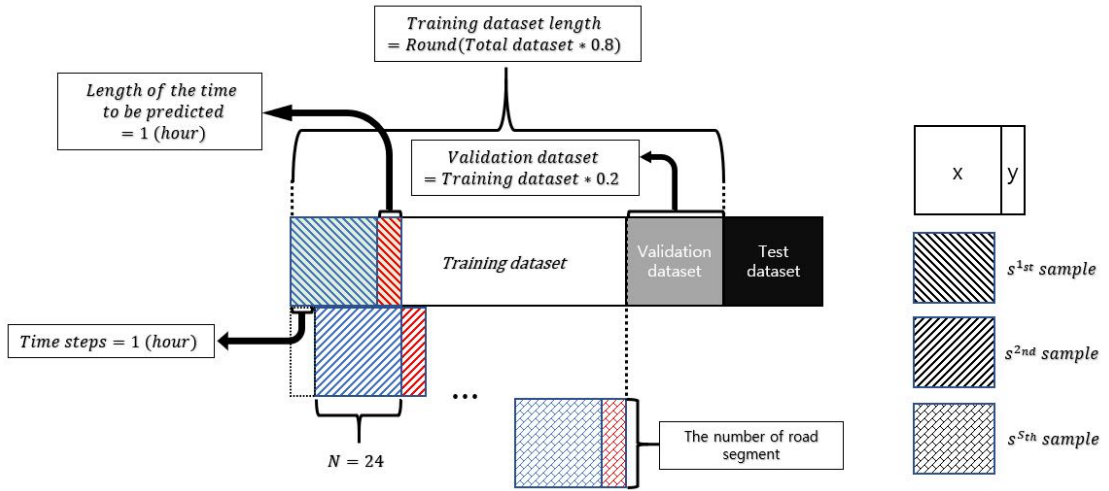


그림 4.5 모델 학습을 위한 데이터 개요

그림 4.5에서 x 가 나타내는 범위는 신경망에 입력되는 이미지의 부분으로 입력값을 의미하며, 동시에 도로 네트워크에 대한 속도를 예측할 때 고려하고자 하는 시간의 범위이다. y 는 매핑하고자 하는 타겟값이며, 예측을 수행할 때 얼마만큼의 시간을 예측할 것인지를 나타낸다. 본 연구는 이처럼 생성된 데이터를 이용해 ResNet을 전이 학습하여 도로 네트워크의 속도를 예측하였으며, 그림 4.6은 예측 과정을 나타낸다.

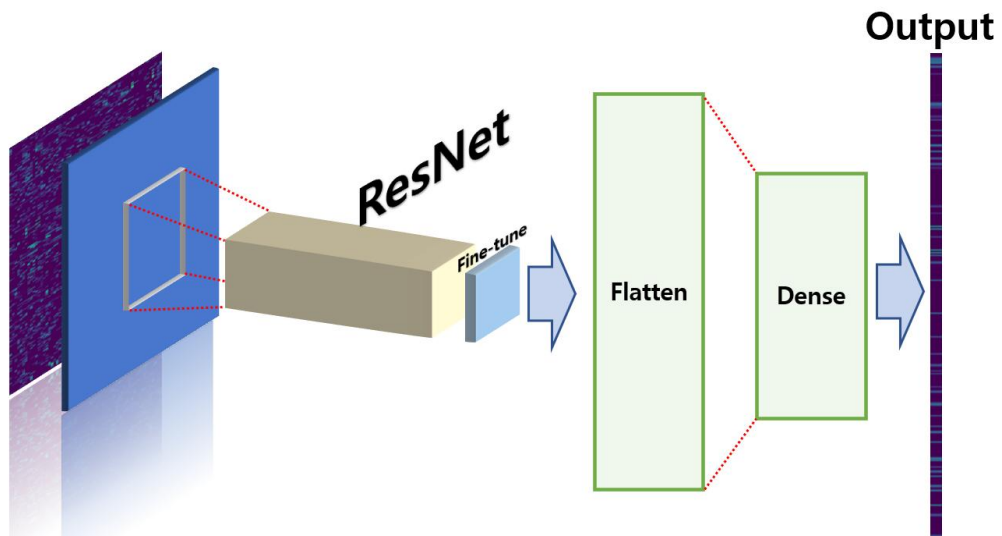


그림 4.6 ResNet을 이용한 통행 속도 예측 아키텍처

본 연구에서 생성된 이미지 데이터는 기존 ResNet의 가중치를 학습시킨 ImageNet의 데이터와는 형태가 다르다. ImageNet의 데이터는 약 1,400만 개의 이미지로 구성되어 있으며, 이들은 동물부터 가전기기까지 다양한 종류의 이미지를 포함한다. ImageNet의 데이터는 실세계에서 찾아볼 수 있는 종류지만 연구에서 생성한 이미지는 도로 네트워크 환경에서의 통행 속도를 이미지로 만든 것이기 때문에 본 연구에서 사용된 데이터의 특성을 반영하기 위한 과정이 필요하다.

ResNet의 입력값의 형태는 본래 ImageNet의 데이터에 맞춰져 있다. 때문에 본 연구에서 생성한 이미지를 적용하기 위해선 이미지의 크기를 ImageNet의 이미지의 크기와 같게 만들어야 한다. 이 과정에서 이미지의 모서리 부분을 0으로 채우는 zero padding이 적용되었다. 이후 ResNet을 통해 교통 이미지 데이터의 특징을 추출하고, 추출된 이미지의 특징을 예측할 시퀀스로 매핑하는 과정을 거쳐 학습한다. 본 연구에서는 ResNet을 전이 학습하였으며, 그 과정에서 미세조정(fine-tuning)하였다. 이는 본래 ResNet의 학습을 위해 사용된 ImageNet과 다른 특성을 가진 이미지 데이터의 특성을 반영하기 위함이다.

제 5장 분석 결과

5.1 도로 네트워크 사용 빈도 상위 5% 예측 결과

전체 도로 네트워크 중 빈도수 상위 5%에 대한 예측은 본 연구에서 제시한 모델 1, 2와 Ma et al.'s 모델에 의해 수행되었다. 그림 5.1은 Ma et al.'s 모델의 epoch과 loss의 관계를 나타내며, 표 5.1과 그림 5.2는 Ma et al.'s 모델의 예측 수행결과를 나타낸다.

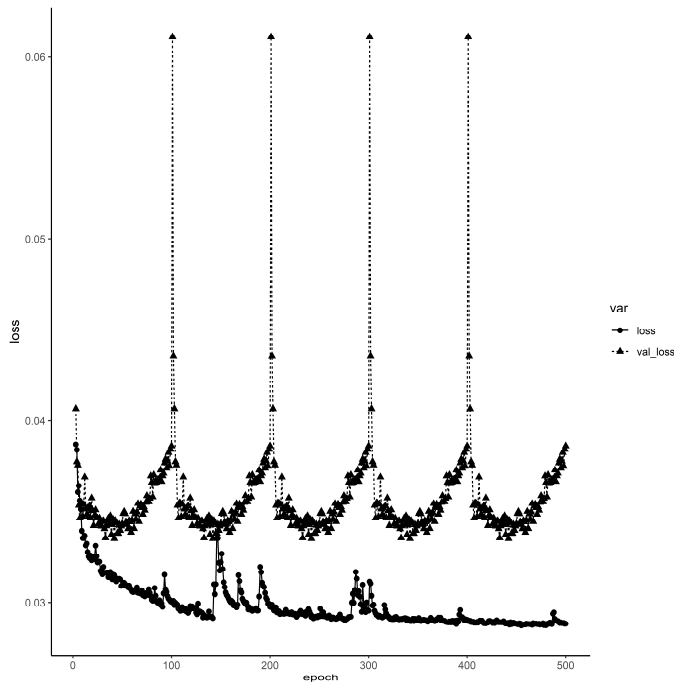


그림 5.1 Ma et al. 's 모델의 epoch과 loss의 관계

표 5.1 Ma et al. 's 모델의 학습 결과

epoch	MAE
10	9.479
50	9.405
100	10.751

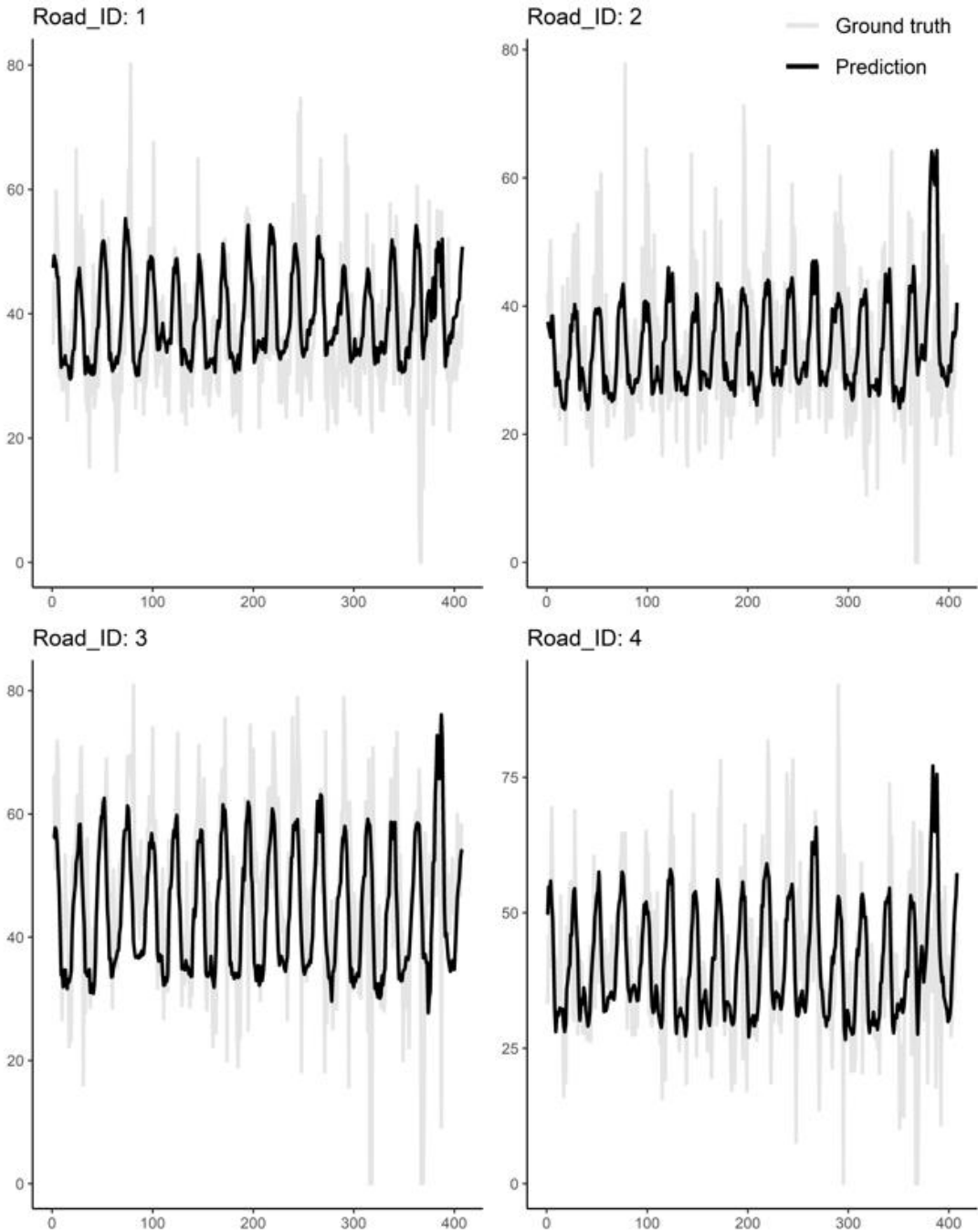


그림 5.2 Ma et al.'s 모델의 속도 예측 결과 중 일부

ResNet-50을 이용한 모델 1의 경우, stage 3까지 고정한 후 학습하였을 때 가장 낮은 MAE 값(9.363)을 보였다. 그림 5.3은 모델 1의 epoch과 loss의 관계를 나타내며, 표 5.2와 그림 5.4는 모델 1의 예측 수행결과를 나타낸다.

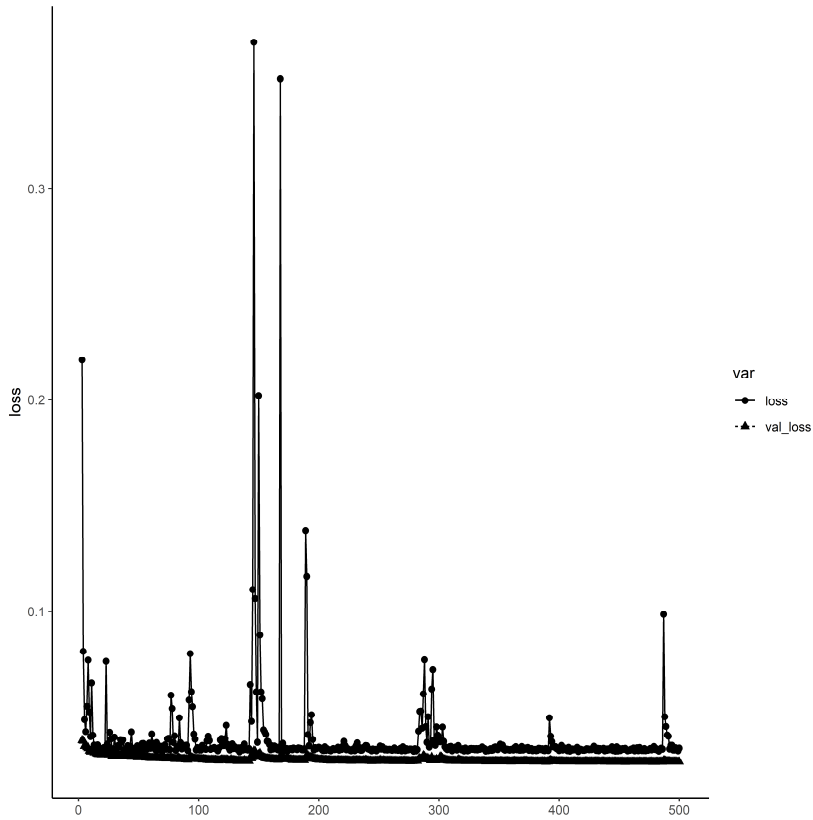


그림 5.3 모델 1의 epoch과 loss의 관계

표 5.2 모델 1의 학습 결과

Fine-tuning(freeze)	MAE
Stage 1-1	10.415
Stage 1-2	9.428
Stage 1-3	9.363
Stage 1-4	9.802
Stage 1-5	31.806

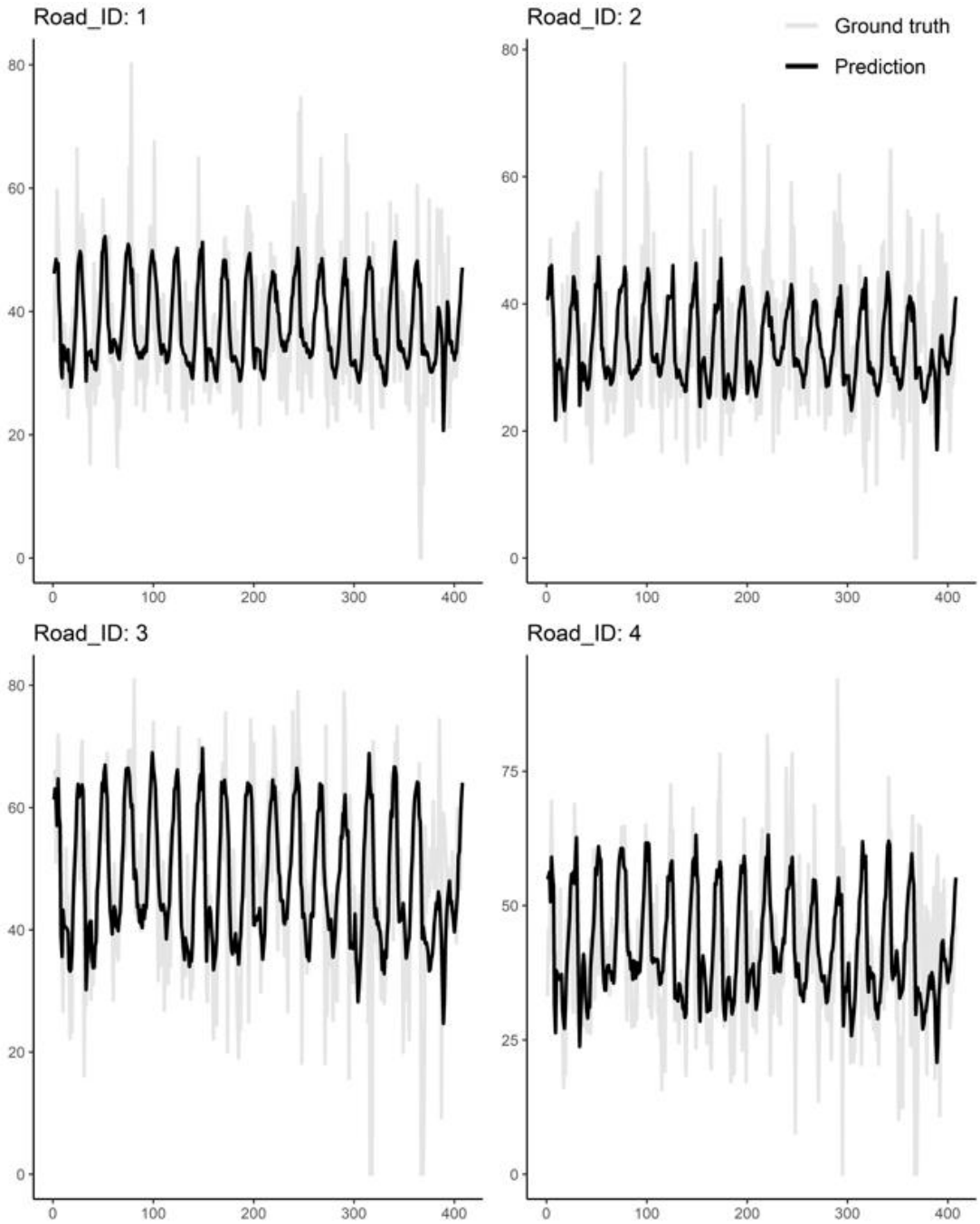


그림 5.4 모델 1의 속도 예측 결과 중 일부

ResNet-152을 이용한 모델 2의 경우, stage 3까지 고정된 후 학습하였을 때 가장 낮은 MAE 값(9.056)을 보였다. 그림 5.5는 모델 1의 epoch과 loss의 관계를 나타내며, 표 5.3과 그림 5.6은 모델 2의 예측 수행결과를 나타낸다.

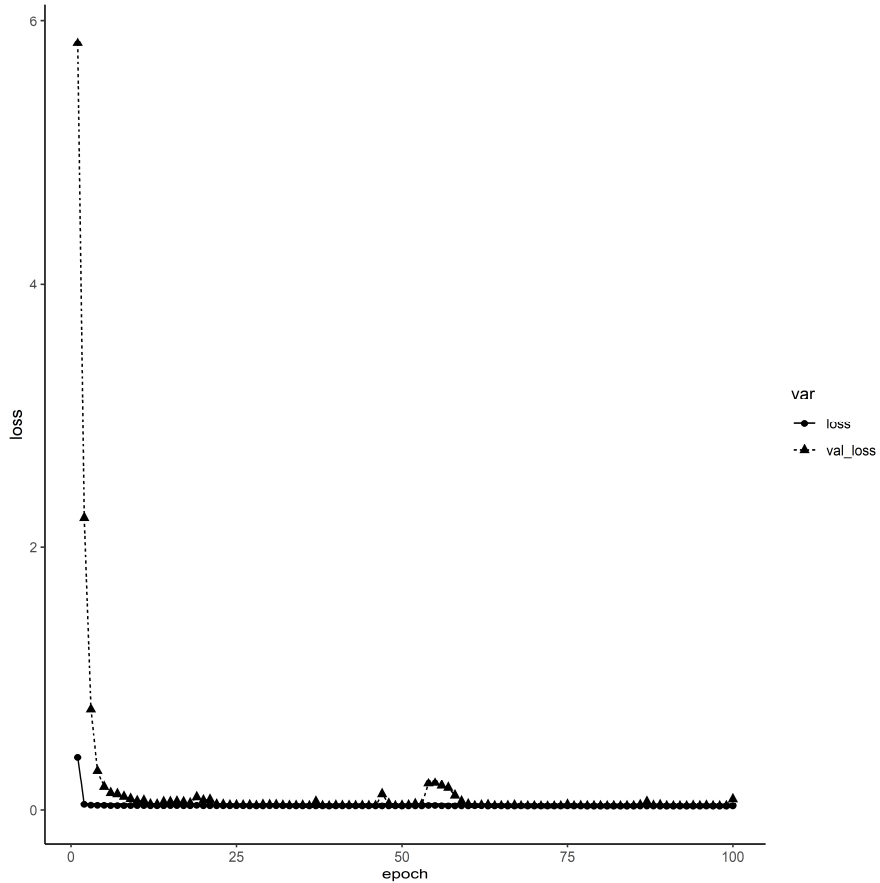


그림 5.5 모델 2의 epoch과 loss의 관계

표 5.3 모델 2의 학습 결과

Fine-tuning(freeze)	MAE
Stage 1-1	9.359
Stage 1-2	12.097
Stage 1-3	9.056
Stage 1-4	9.451
Stage 1-5	10.065

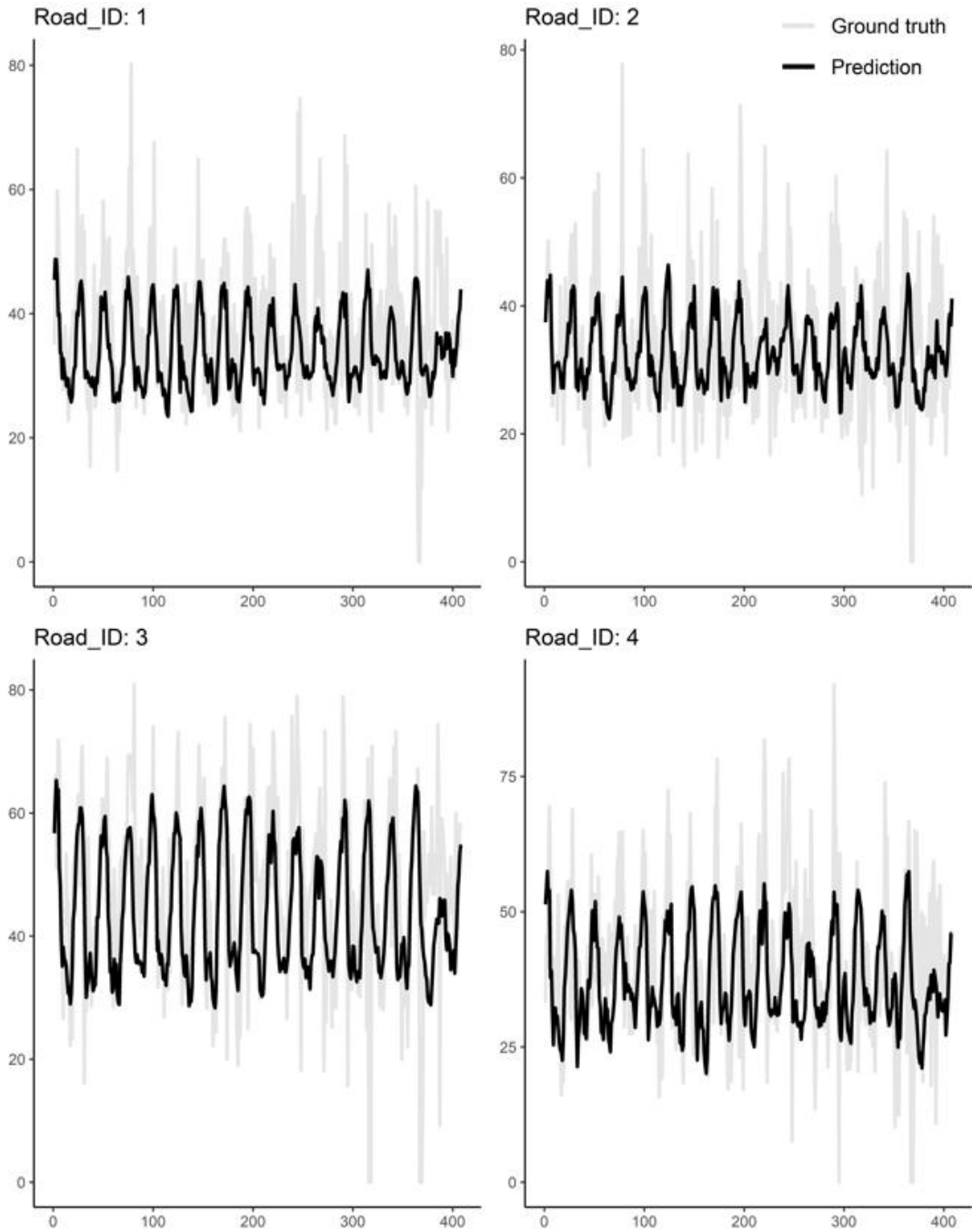


그림 5.6 모델 2의 속도 예측 결과 중 일부

5.2 결과 비교

본 연구는 대구광역시의 도로 중 사용 빈도수 상위 5%의 네트워크에 대해 속도 예측을 수행하였다. 본 연구에서 제시한 모델은 ResNet을 전이 학습한 모델이며 Ma et al.'s 모델과 비교하여 분석하였다. 결과적으로 본 연구에서 제시한 모델은 MAE를 기준으로 약 3.8% 더 높은 성능을 보였다. 정확도 측면에서 향상된 정도가 통계적으로 유의미한 성능의 향상인지 확인하기 위해 각 모델을 통해 생성된 예측 결과값들 사이에 차이가 존재하는지 알아보기 위한 통계적 검증 방법을 적용하였다. 검증하기 이전에 각각의 예측 결과가 정규성을 따르는지 검정하였으며, 표본 개수에 따라 kolmogorov-smirnov test(Smirnov, 1939)를 수행하였다. 식 14는 두 분포의 차이를 나타낸다.

$$D_{n,m} = \sup |F_{1,n}(x) - F_{2,m}(x)| \quad (14)$$

여기서 \sup 는 상한(supremum), F 는 distribution function을 나타낸다.

정규성 검정 수행결과 Ma et al.'s 모델과 본 연구에서 제시한 모델을 통해 예측한 결과값의 집합은 모두 정규분포를 따르지 않았다. 그러므로, 비 모수적 검증 방식인 wilcoxon rank sum test를 시행하였다. wilcoxon rank sum test는 서로 다른 두 집단이 중앙값이 같은 분포에서 추출된 표본이라는 귀무가설을 검정한다.

수행결과 결과값(p-value < 0.05)이 몹시 작으므로 귀무가설을 기각하고 대립가설을 채택하였다. 두 집단은 같은 중앙값의 분포에서 추출된 표본이 아니며, 이것은 통계적으로 유의미한 결과값 생성의 차이라고 있다고 할 수 있다. 그림 5.7은 각 모델의 예측 수행결과와 분포를 나타낸다.

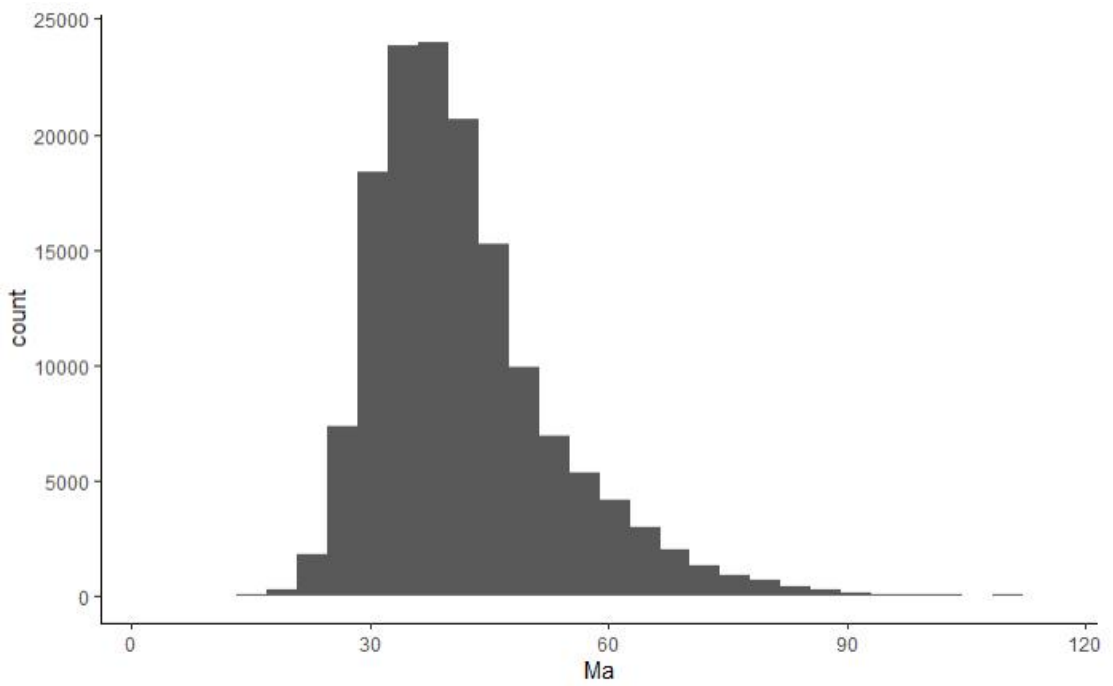
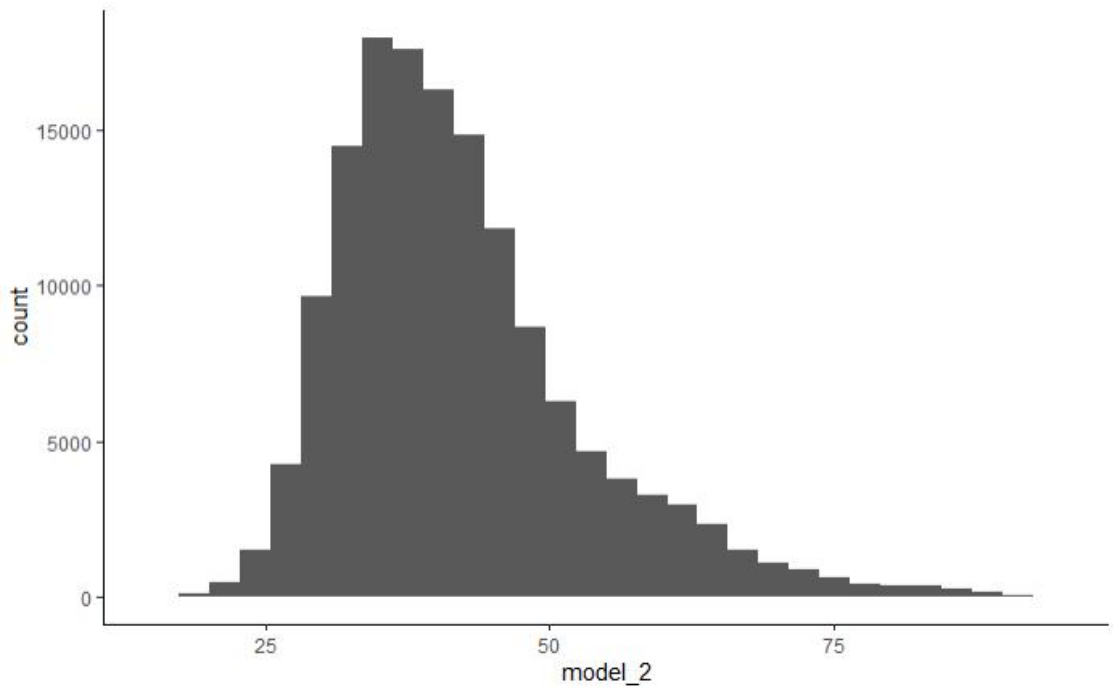


그림 5.7 각 모델의 예측 결괏값 분포

6. 결 론

시간의 영향을 받는 교통 데이터의 특성을 반영하기 위해 사용되었던 기존의 시계열 분석 방식은 네트워크의 규모가 증가함에 따라 정의되어야 하는 모델의 수가 같이 증가하는 문제가 있었다. 또한, 시간뿐만 아니라 공간에도 종속되어 서로의 상태에 영향을 미치는 네트워크의 환경에서 기존의 시계열 분석 방식은 각각의 도로 구간별로 모델을 정의하기 때문에 네트워크의 공간적 상관관계를 고려하지 못한다는 한계가 존재한다. 이를 보완하고자 합성곱 신경망을 적용하는 방식이 제안되었고 본 연구는 기존에 적용되었던 합성곱 신경망보다 더 깊은 구조를 가진 ResNet을 전이 학습하여 도로 네트워크의 환경에서 통행 속도를 예측하였다. 예측을 수행하기 위해 모델 학습에 필요한 이미지 데이터를 이모션에서 제공된 택시 이동 포인트 데이터와 국토교통부에서 제공된 도로 링크 데이터를 공간 조인의 방식으로 결합하여 이미지 데이터를 생성하였다. 모델을 통해 예측을 수행하여 비교 분석한 결과 기존의 Ma et al.'s 모델에 비해 ResNet-152를 전이 학습한 모델은 정확도 측면에서 최대 3.8%, 향상되었다. 이는 MAE 값을 척도로 하여 계산된 수치이며, 통계적으로 유의미한 성능의 향상이다.

본 연구의 실험에서 ImageNet의 데이터로 학습이 이루어진 ResNet을 교통 데이터로 전이 학습을 수행할 경우, satage 3까지 가중치를 고정했을 때 높은 성과를 보인다는 것을 확인하였다. 이후 본 연구의 데이터에 한정하는 것인지 추가적인 실험이 더 필요할 것으로 보인다. 또한, 본 연구의 모델은 시간과 공간의 영향만을 고려하였기 때문에 다양한 외적 요소에 영향을 받는 도로 네트워크의 환경을 모두 고려하지 못했다는 한계가 존재한다. 뿐만 아니라 모델을 학습시킬 때 사용한 이미지 데이터는 택시의 데이터만을 사용하여 생성한 이미지이다. 이러한 데이터로 모델을 구축하였기 때문에 연구에서 제시된 모델을 통해 수행한 예측이 일반적인 도로 네트워크에서의 이동 객체에 대한 속도를 대표한다고 보기 어렵다. 향후 택시를 포함한 이동 객체의 데이터와 도로 네트워크를 구성하는 다양한 요소들을 고려한 모델을 구축할 필요가 있을 것으로 보인다.

참 고 문 헌

1. 전승배, 오행열, & 정명훈. (2020). 머신러닝을 이용한 해양수질 등급 추정. *대한공간정보학회지*, 28(4), 145-152.
2. 조한선, 이호, & 김영춘. (2014). 2011, 2012 년 전국 교통혼잡비용 추정과 추이 분석. *한국교통연구원 수시연구보고서*, 1-121.
3. Ahmed, M. S., & Cook, A. R. (1979). Analysis of freeway traffic time-series data by using Box-Jenkins techniques (No. 722).
4. Box, G. E., Jenkins, G. M., & Reinsel, G. (1970). Time series analysis: forecasting and control Holden-day San Francisco. *BoxTime Series Analysis: Forecasting and Control Holden Day1970*.
5. Cauchy, A. (1847). Méthode générale pour la résolution des systemes d'équations simultanées. *Comp. Rend. Sci. Paris*, 25(1847), 536-538.
6. Chen, M., Liu, X., Xia, J., & Chien, S.I. (2004). A dynamic bus arrival time prediction model based on APC data. *Computer Aided Civil and Infrastructure Engineering*, 19(5), 364-376.
7. Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.
8. Dai, G., Ma, C., & Xu, X. (2019). Short-term traffic flow prediction method for urban road sections based on space - time analysis and GRU. *IEEE Access*, 7, 143025-143035.
9. Dougherty, M. (1995). A review of neural networks applied to transport. *Transportation Research Part C: Emerging Technologies*, 3(4), 247-260.
10. Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7).
11. Fukushima, K. (1979). Neural network model for a mechanism of pattern recognition unaffected by shift in position-Neocognitron. *IEICE Technical*

- Report, A*, 62(10), 658-665.
12. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *In Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).
 13. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735-1780.
 14. Hochreiter, S., & Schmidhuber, J. (1997). LSTM can solve hard long time lag problems. *Advances in neural information processing systems*, 473-479.
 15. Ioffe, S., & Szegedy, C. (2015, June). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *In International conference on machine learning* (pp. 448-456). PMLR.
 16. Jeon, S.B., Jeong, M.H., Lee, T.Y., Lee, J.H., & Cho, J.M. (2020). Bus Travel Speed Prediction Using Long Short-term Memory Neural Network. *Sensors and Materials*, 32(12), 4441-4447.
 17. Jeong, M.H., Lee, T.Y., Jeon, S.B., & Youm, M. (2021). Highway Speed Prediction Using Gated Recurrent Unit Neural Networks. *Applied Sciences*, 11(7), 3059.
 18. Jia, Y., Wu, J., & Du, Y. (2016, November). Traffic speed prediction using deep learning method. *In 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)* (pp. 1217-1222). IEEE.
 19. Khan, A., Sohail, A., Zahoor, U., & Qureshi, A. S. (2020). A survey of the recent architectures of deep convolutional neural networks. *Artificial Intelligence Review*, 53(8), 5455-5516.
 20. Kiefer, J., & Wolfowitz, J. (1952). Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics*, 23(3), 462-466.
 21. Kim, Y., Wang, P., Zhu, Y., & Mihaylova, L. (2018, October). A capsule network for traffic speed prediction in complex road networks. *In 2018 Sensor Data Fusion: Trends, Solutions, Applications (SDF)* (pp. 1-6). IEEE.
 22. Kingma, D.P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

23. Krizhevsky, A., Sutskever, I., & Hinton, G.E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 1097-1105.
24. LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4), 541-551.
25. Ma, X., Dai, Z., He, Z., Ma, J., Wang, Y., & Wang, Y. (2017). Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction. *Sensors*, 17(4), 818.
26. Ma, X., Tao, Z., Wang, Y., Yu, H., & Wang, Y. (2015). Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transportation Research Part C: Emerging Technologies*, 54, 187-197.
27. Mou, L., Zhao, P., Xie, H., & Chen, Y. (2019). T-LSTM: A long short-term memory neural network enhanced by temporal information for traffic flow prediction. *Ieee Access*, 7, 98053-98060.
28. Okutani, I., & Stephanedes, Y. J. (1984). Dynamic prediction of traffic volume through Kalman filtering theory. *Transportation Research Part B: Methodological*, 18(1), 1-11.
29. Ren, S., Yang, B., Zhang, L., & Li, Z. (2018, November). Traffic speed prediction with convolutional neural network adapted for non-linear spatio-temporal dynamics. *In Proceedings of the 7th ACM SIGSPATIAL International Workshop on Analytics for Big Geospatial Data* (pp. 32-41).
30. Rosenblatt, F. (1957). The perceptron, a perceiving and recognizing automaton Project Para. *Cornell Aeronautical Laboratory*.
31. Rubin, D. B. (1987). The calculation of posterior distributions by data augmentation: Comment: A noniterative sampling/importance resampling alternative to the data augmentation algorithm for creating a few imputations when fractions of missing information are modest: The SIR algorithm. *Journal of the American Statistical Association*, 82(398), 543-546.

32. Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1985). Learning internal representations by error propagation. *California Univ San Diego La Jolla Inst for Cognitive Science*.
33. Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, 323(6088), 533-536.
34. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... & Fei-Fei, L. (2015). Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3), 211-252.
35. Schafer, J. L. (1999). Multiple imputation: a primer. *Statistical methods in medical research*, 8(1), 3-15.
36. Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
37. Smirnov, N. V. (1939). On the estimation of the discrepancy between empirical curves of distribution for two independent samples. *Bull. Math. Univ. Moscou*, 2(2), 3-14.
38. Spear, M. E. (1952). Charting statistics.
39. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. *In Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1-9).
40. Tieleman, T., & Hinton, G. (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2), 26-31.
41. TN, H. D. W. (1959). Receptive fields of single neurones in the cat's striate cortex. *Journal of Physiology*, 148, 574-591.
42. Tran, D. Q., Park, M., Jung, D., & Park, S. (2020). Damage-Map Estimation Using UAV Images and Deep Learning Algorithms for Disaster Management System. *Remote Sensing*, 12(24), 4169.
43. Van Lint, J. W. C., Hoogendoorn, S. P., & van Zuylen, H. J. (2002). Freeway travel time prediction with state-space neural networks: modeling state-space dynamics with recurrent neural networks. *Transportation Research Record*,

- 1811(1), 30-39.
44. Vlahogianni, E. I., Karlaftis, M. G., & Golias, J. C. (2014). Short-term traffic forecasting: Where we are and where we're going. *Transportation Research Part C: Emerging Technologies*, 43, 3-19.
 45. Weisbrod, G., Vary, D., & Treyz, G. (2003). Measuring economic costs of urban traffic congestion to business. *Transportation research record*, 1839(1), 98-106.
 46. Williams, B. M., Durvasula, P. K., & Brown, D. E. (1998). Urban freeway traffic flow prediction: application of seasonal autoregressive integrated moving average and exponential smoothing models. *Transportation Research Record*, 1644(1), 132-141.
 47. Willmott, C. J., & Matsuura, K. (2005). Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate research*, 30(1), 79-82.
 48. Zhao, L., Song, Y., Zhang, C., Liu, Y., Wang, P., Lin, T., ... & Li, H. (2019). T-gcn: A temporal graph convolutional network for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems*, 21(9), 3848-3858.
 49. Zhao, Z., Chen, W., Wu, X., Chen, P. C., & Liu, J. (2017). LSTM network: a deep learning approach for short-term traffic forecast. *IET Intelligent Transport Systems*, 11(2), 68-75.

부 록

도로 네트워크 통행 속도의 이미지화 pseudocode

Time steps and prediction period = 1 (hour)

Require: i : The number of road

Require: j : Lookback period

Require: t : Time index

Require: v : Average speed value for training data set

$x_{it} \leftarrow [v_{11}, v_{21}, \dots, v_{i1}, v_{12}, \dots, v_{it}]$

$n_{sample} \leftarrow \frac{\text{length}(x_{it})}{i} - j$

for k in 1: n_{sample}

$X_{train}[k, ,] \leftarrow x_{it}[(i * (k - 1) + 1): (i * (j + k - 1))]$

$Y_{train}[k,] \leftarrow x_{it}[(i * (j + k - 1) + 1): (i * (j + k))]$

end for

return X_{train}, Y_{train}

감사의 글

2017년을 시작으로 연구실에서 보낸 시간을 한 권의 논문으로 표현하기엔 부족했던 것 같습니다. 본 논문은 많은 분들의 격려와 도움이 밑거름되어 맺어진 소중한 결실임을 알기에, 이 글을 통해 고마운 분들께 진심으로 감사의 마음을 전하고자 합니다.

세상 물정 모르던 저를 진심 어린 배려와 관심으로 학문적 지도뿐만 아니라 인생을 대하는 태도까지 가르침을 주신 정명훈 교수님께 진심으로 감사드립니다. 학부 시절부터 학위 논문까지 관심을 가지고 지켜봐 주신 박정웅, 김성홍, 김대현, 박상준, 장우식, 이희영 교수님, 부족한 논문임에도 불구하고 시간을 내어 읽어 주시고 세세한 부분까지 지도해주셔서 감사드립니다. 조선대학교 선배로서 후배들을 많이 아끼고 챙겨주시는 이정환 박사님께 진심으로 감사드립니다. 함께 연구실에서 많은 시간을 보낸 이태영, 김건, 오행열 연구원과 비록 짧은 시간 함께 하였지만, 논문을 완성하는 데 도움을 준 Muhammad Sarfraz Khan, Almirah Monireth Peter에게 감사드립니다.

한때 학업을 포기할 생각을 하던 저에게 묵묵히 도움을 주시고 가족들을 챙겨주신 송강석 삼촌께 진심으로 감사드립니다. 그리고 무엇보다 힘든 일이 있을 때 버팀목이 되어주고 항상 저를 즐겁게 해준 강진모, 우세림, 김진석, 김예지, 조동현과 김령현에게 감사드립니다. 앞으로도 인생을 살아가면서 많은 어려움과 직면하겠지만 저에게 주신 은혜를 평생 가슴에 새기고 기억하면서 부끄럽지 않게 열심히 살아가겠습니다.

마지막으로, 힘든 세월 서로 눈물을 닦아주던 동생 예은이, 암 투병 중에 더는 걸을 수 없게 되어도 아들 걱정에 얼른 학교 가라고 하시던 아버지, 아버지께서 세상을 떠나시고 성치 않은 몸으로 홀로 자식들 뒷바라지하면서도 항상 웃으며 씩씩하게 저희를 돌봐주신 어머니께 못난 아들이 죄송한 마음으로 감사와 사랑을 담아 이 논문을 바칩니다.

2021년 6월
전 승 배 올림