정보통신공학과 졸업 논문

CNN 을 이용한 차종 판별 시스템 구현

Implementation of Car Classification System using Convolutional Neural Network

한국외국어대학교

정보통신공학과

201402750

임광효

지도 교수: 한 희 일

졸업논문제출 청구 및 승인서

제 목 : CNN을 이용한 차종 판별 시스템 구현

대 학 : 한국외국어대학교 학 과 : 정보통신공학과

학 번 : 201402750 성 명 : 임 광 효

이 논문을 제출하오니 승인하여 주십시오.

2019 년 11 월 28 일

성 명 : 임 광 효 (인)

.....

위 학생의 논문 제출을 승인함.

2019 년 11 월 28 일

지 도 교 수 : 한 희 일 (인)

목 차

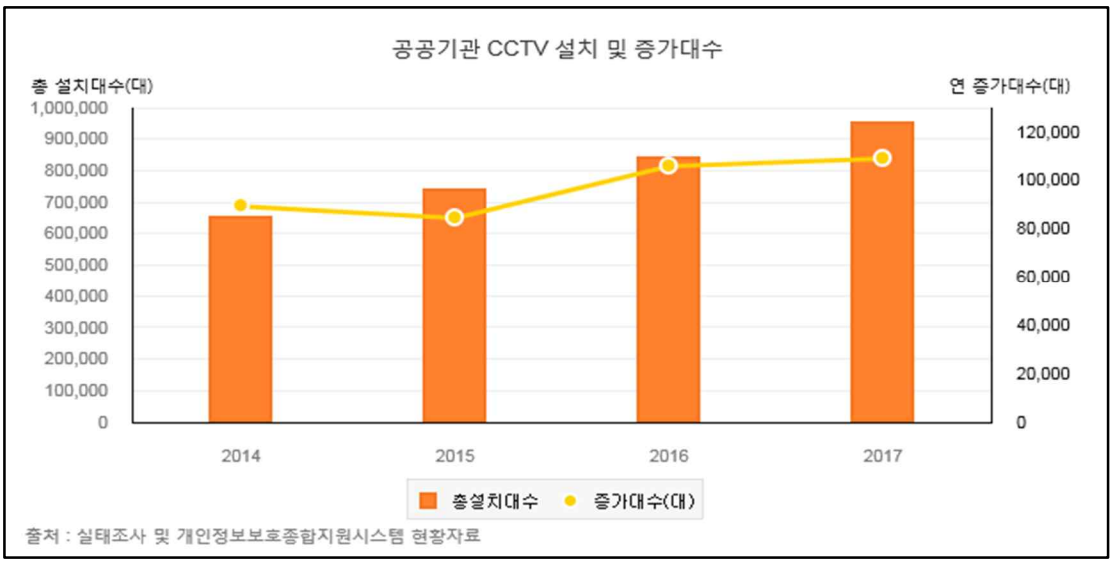
그 림 목 차

수 식 목 차

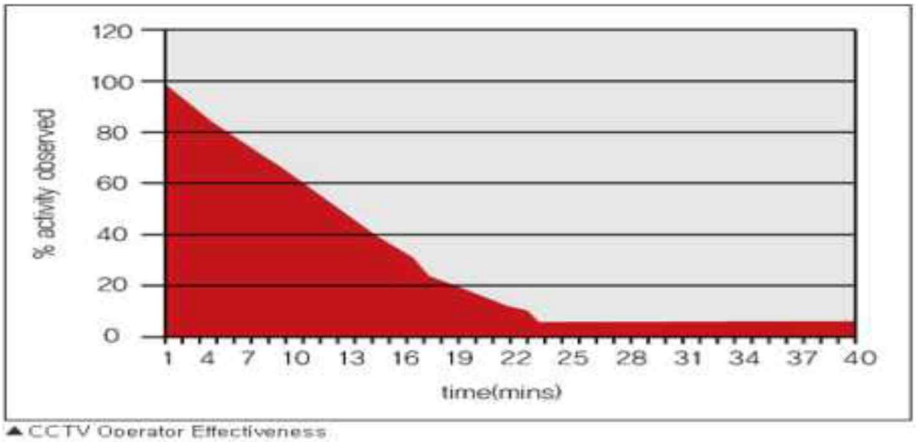
표 목 차

1. 서론

오늘날 각종 범죄 예방 및 교통 정보 수집 등의 이유로 CCTV 설치 대수는 매년 증가하는 추세이나, 한정된 인력이 매년 증가하는 CCTV 를 효율적으로 관리하고 모니터링하기에는 한계가 존재한다. 미 국립사법연구소(NIJ)의 보고서[1]에 의하면, 관제 요원이 직접 CCTV 를 통해 상황을 감시하는 기존 CCTV 시스템에서는 상황을 감시하기 시작한 후, 22 분이 지나면, 위험을 탐지하는 능력이 약 5%로 현저히 떨어진다는 실험결과가 존재한다. 따라서 이에 대한 해결 방안으로, 한정된 인력으로도 다수의 CCTV 를 관리하고 모니터링 할 수 있는 시스템의 필요성이 대두되고 있다.



[그림 1-1] 연도별 공공기관 CCTV 설치 현황



[그림 1-2] 수동 영상관제 시간에 따른 위험탐지율

본 논문에서는 이러한 한계를 극복하기 위해 빈번히 발생하는 뺑소니 차량이나 도난 차량 및 범죄의 연루된 차량에 대한 수배를 조기에 처리하는데 도움을 줄 수 있는 차종 판별 시스템을 구현하는 것에 목적이 있다. 현재 CNN 을 사용한 차종 판별은 시중에 나와있는 모든 차종을 판별하기에는 수가 방대하여 판별이 어렵고, 또한 모든 차종에 대한 데이터셋을 마련하기에는 무리가 있어 차종을 특정하여 차종을 판별하는 방향으로 연구가 진행되고 있다. Derrick Liu 와 Yushi Wang 은 196 종의 차종을 특정하고, 이를 판별하기 위해 SVM 과 1-layer CNN 를 기반으로 한 모델과 GoogLeNet, VGGNet, CaffeNet 을 각각 사전 학습, 부분 학습, 완전 학습으로 나누어 학습한 모델들을 제안하였으며[2], Burak Satar 와 Ahmet Emir Dirik 은 Detector 인 SSD[3]와 ResNet[4]를 사용한 모델을 제안하였다[5]. SVM 과

1-layer CNN 을 사용한 모델은 구조가 단순하고 간단하여 적은 시간으로도 학습이 가능하였으나, 100 여종이 넘는 차종을 판별하기에는 데이터셋의 크기가 작고 모델의 깊이가 충분히 깊지 못해 overfitting 문제가 발생하였다. SSD 와 ResNet 을 조합한 모델은 Detector 를 통해 먼저 차량을 인식한 후 학습한 모델로, Detector 를 사용하지 않은 모델보다 높은 성능을 보였다. 하지만 Detector 를 통해 먼저 차량을 인식하는 과정이 완벽하게 수행되지 않아 차종 판별에 있어 Detector 의 성능이 모델의 성능에 크게 영향을 끼치는 단점이 존재했으며, 판별하는 차종의 수가 7 종으로 매우 적었다. 또한 Lee 와 Chung 이 제안한 다른 모델[6]은 차량을 소그룹으로 묶은 데이터셋을 각각 학습한 local 모델과 차량 전체를 데이터셋으로 학습한 global 모델로 네트워크를 이루고 Softmax 로 결과를 출력하는 모델이다. 성능은 최저 68.72%에서 최고 99.66%으로 매우 정확하지만, 해당 모델을 위해서 12 개의 local 모델과 6 개의 global 모델을 사용해 엄청난 계산 능력이 요구되어 다수의 고 사양 GPU 를 필요로 하는 단점이 있다. 따라서 본 논문에서 구현하고자 하는 시스템에서는 차종을 한정한 후, 모델의 깊이를 조정하여 한정한 차종에 대해 적합한 모델의 깊이를 찾는 것을 목표로 하며, 실제 차량에 대한 차종 분류 성능을 시험해보고자 한다.

본 논문의 구현 환경은 Windows 와 Linux 환경에서 Python 기반 가상 개발 환경인 Anaconda 를 사용하여 진행하며, 해당 학습을 위해 오픈 소스 이미지 데이터 셋인 Stanford Cars Dataset 을 224*224 해상도로 일괄 조정한 이미지를

사용한다. 학습 방법은 ImageNet 데이터셋으로 사전 학습된 ResNet-152 모델을 통해 학습을 진행한다.

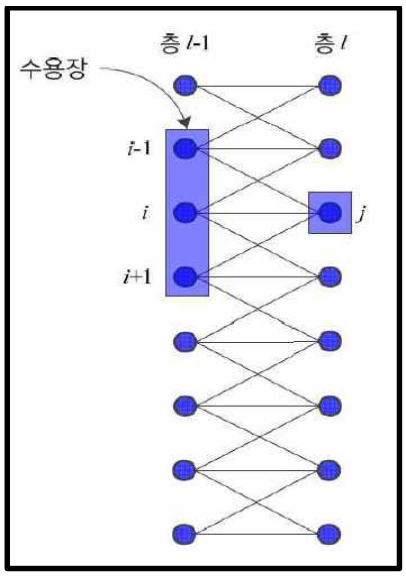
본 논문의 구성은 다음과 같다. 2 장에서는 학습에 사용되는 핵심 기술인 CNN 과 주요 CNN 모델에 대한 구조 및 특징을 기술하고, 각 성능을 비교하여 시스템에 적합한 CNN 모델을 선정한다. 3 장에서는 본 논문에서 제안한 차종 판별의 설계 및 흐름도와 학습 모델을 사용하여 테스트를 진행하는 과정을 기술하며, 해당 테스트의 결과를 설명하고 이를 분석하여 개선 방안을 제시한 뒤, 시스템의 개선을 진행한다. 이후 개선된 실험 결과와 이전 실험 결과를 비교 분석하여 최종 시스템을 결정짓고 실생활에 적용가능 한지를 실험한 후, 결과를 토대로 결론을 짓고 향후 연구 과제를 제시한다.

2. 기술조사

본장에서는 차종 판별 시스템 구현에 있어 핵심이 되는 기술인 CNN 에 대해 알아보고, CNN 을 활용한 각 학습 모델을 소개한 후 성능을 비교하여 본 논문에 사용할 학습 모델을 선정한다

2.1 CNN

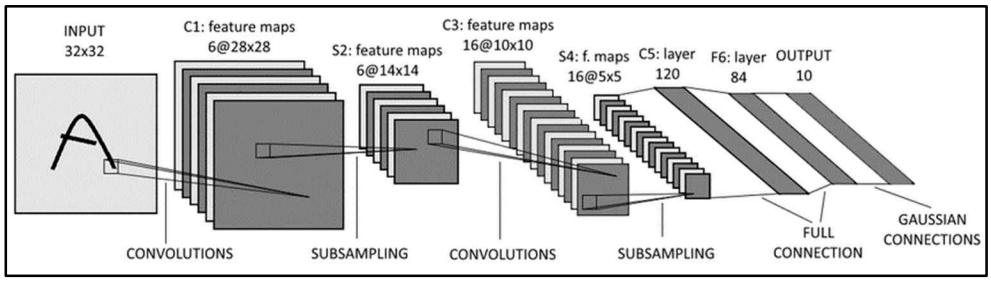
CNN(Convolution Neural Network)은 Convolution 연산을 이용한 부분연결 구조로 복잡도를 크게 낮췄으며, 연산을 위한 입력 데이터의 특징을 직접 추출하여 학습을 해야 하는 머신 러닝과는 달리, 입력 데이터의 특징을 해당 네트워크가 추출하여 학습에 사용하는 딥러닝 구조이다.



[그림 2-1] CNN 의 부분연결 구조

예로 [그림 2-1]과 같이, CNN 에서 j 노드는 이전 층의 i-1, i, i+1 노드와 연결되어 있는 부분 연결구조를 가진다.

[그림 2-2]는 LeCun 이 제안한 LeNet-5[7]로서, 오늘날 쓰이는 CNN 구조의 근간이 되는 네트워크이다. LeNet-5 의 구조를 사용하여 CNN 을 설명한다.



[그림 2-2] LeNet-5 구조

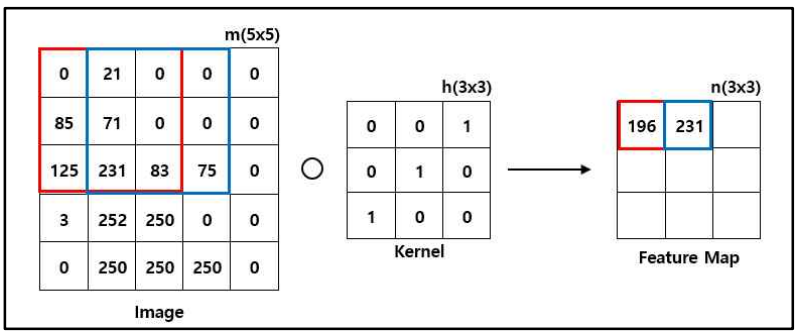
CNN 은 크게 특징을 추출하는 Convolution Layer 와 Topology 변화의 영향을 받지 않게 해주는 Pooling Layer 로 구성되어 있다. LeNet-5 에서는 3 번의 Convolution 과 2 번의 Pooling 을 거치며, Convolution 과 pooling 과정을 거쳐 남은 강인한 특징을 통해 결과를 분류하기 위해 Fully-Connected Layer 를 추가하여 이미지를 구별한다.

먼저 Convolution Layer 는, Convolution 연산을 통해 이미지의 특징을 추출한다. Convolution 연산의 공식은 식 (2-1)과 식 (2-2)와 같다.

$$s(i) = z \circledast u = \sum_{x=-(h-1)/2}^{(h-1)/2} z(i+x)u(x) \qquad \textbf{(2-1)}$$

$$s(j,i) = z \circledast u = \sum_{y=-(h-1)/2}^{(h-1)/2} \sum_{x=-(h-1)/2}^{(h-1)/2} z(j+y, i+x)u(y,x) \qquad \textbf{(2-2)}$$

식 (2-1)은 1 차원에서의 Convolution 연산이며, u 는 커널, h 는 커널의 크기, z 는 입력, s 는 출력이다. 이때의 출력 s 를 Feature Map 이라고 한다. 식 (2-2)는 영상 데이터와 같이 2 차원 구조를 가지는 데이터에서의 Convolution 연산의 공식이며, CNN 에서 커널의 높이와 너비는 동일하게 설정하므로 크기를 모두 h 라고 표기한다. [그림 2-3]은 2 차원 구조를 가지는 5x5 크기의 데이터와 h=3 인 커널과의 Convolution 연산 예시이다. 빨간 테두리 영역은 $\begin{pmatrix} 0 & 21 & 0 \\ 85 & 71 & 0 \\ 125 & 231 & 83 \end{pmatrix}$ 이고, 커널은 $\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$ 이므로 식은 0*0+21*0+0*1+85*0+71*1+0*0+125*1+231*0+83*0=196 이다. 같은 방식으로 파란색 테두리 영역에 같은 연산을 하면 231 이 나오며, 이후 모든 영역에 연산을 진행하여 Feature Map 을 출력한다. 이때의 Feature map 크기는 Convolution 연산하는 커널의 보폭을 1 로 했을 때 이미지의 크기 – 커널의 크기 + 1 이다.



[그림 2-3] Convolution 연산

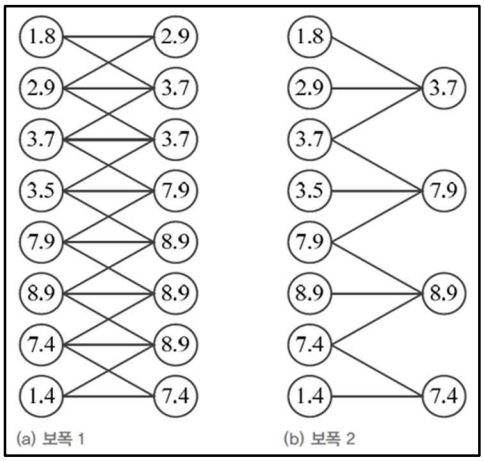
Feature Map 의 크기는 [그림 2-3]과 같이 커널의 크기가 클수록, 또는 Convolution 연산을 거칠수록 줄어드는데, 많은 Layer 를 가지는 깊은 신경망일수록 Convolution 연산이 여러 번 반복되므로 줄어드는 양이 증가하여 문제가 발생한다. 이러한 문제를 방지하기 위해 padding 을

수행하여 크기를 유지한다. [그림 2-4]는 [그림 2-3]의 Feature Map 을 zero-padding 한 결과이다.

0	0	0	0	0
0	196	231	83	0
0	234	335	325	0
0	335	575	250	0
0	0	0	0	0

[그림 2-4] Zero-padding

Convolution Layer 바로 다음에 Pooling Layer 가 따른다. Pooling Layer 는 Convolution Layer 출력된 결과를 Subsampling 하는데, CNN 에서는 추출된 특징 중 가장 강한 특징을 pooling 하는 방식인 max-pooling 방식을 취한다. 이때 보폭을 1 로 설정한다면, Feature Map 의 크기가 유지되지만, 보폭을 2 이상으로 하면 Feature Map 의 크기를 줄이는 효과가 있다. [그림 2-5]는 보폭을 1 과 2 로 했을 때의 경우를 나타낸다.



[그림 2-5] max-pooling

여러 번의 Convolution 연산과 Pooling 을 거치고 나면, Feature Map 의 크기가 작아지면서 동시에 이미지를 대표하는 강인한 특징만 남게 되는데, 이 Feature Map 을 입력하여 이미지를 분류하기 위해 Fully-Connected Layer 를 거쳐 결과를 추출한다.

[그림 2-2]의 LeNet-5 구조에서는 2 개의 Fully-Connected Layer 를 사용하며, 이후 softmax 활성함수를 사용하여 결과를 추출한다.

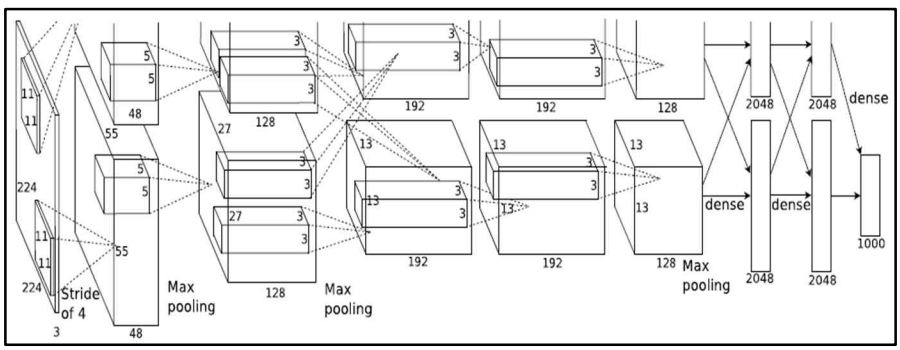
2.2 CNN Models

본 논문에서 사용할 CNN 모델을 선정하기 위해, 먼저 ImageNet 에서 매년 주최하는 이미지 분류 및 검출, 위치 지정 문제를 푸는 대회인 ILSVRC(ImageNet Large Scale Visual Recognition Challenge)에서 2012 년 우승한 AlexNet[7], 2014 년 우승한 GoogleNet[8], 그리고 2015 년에 우승한 ResNet-152[4] 이 3 가지를 소개한다.

2.2.1 AlexNet

AlexNet 은 깊은 구조를 가지는 CNN 모델이 가지는 한계인 학습 데이터의 부족을 ImageNet 데이터베이스로 해결함으로써 CNN 을 컴퓨터 비전 분야에 적용시킬 수 있다는 것을 널리 알린 모델이다.

그림[2-5]는 AlexNet 의 구조로, 5 개의 Convolution Layer 와 3 개의 Fully-Connected Layer 를 가지며, 224x224 크기의 RGB 영상을 입력으로 받는다. 총 8 개의 층으로 이루어져 있으며, 각 층에 253,440–186,624–64,896–64,896–43,264–4096–4096–1000 개의 뉴런이 배치되어 있는 복잡한 신경망이다.

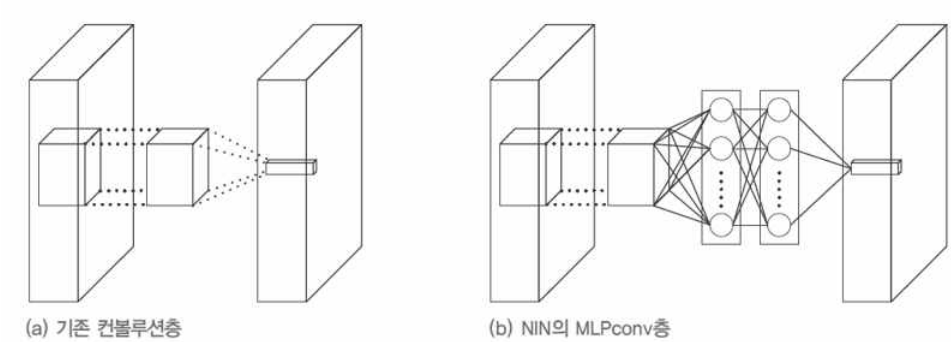


[그림 2-6] AlexNet 구조

복잡한 신경망을 가지는 만큼, AlexNet 은 원활한 학습을 위해 여러 가지 기법을 사용하였다. 먼저 학습을 위해 2 대의 GPU 를 사용하여 학습 시간을 단축하였고, 기존에 쓰이는 활성함수인 tanh 대신 ReLU 를 사용하였다. 또한 Convolution 연산의 결과를 이웃 커널의 값을 고려해 조정하는 Local Response Normalization 을 사용하였다. AlexNet 이 방대한 매개변수를 가지는 깊은 구조인 만큼, 과잉 적합 문제 발생을 방지하기 위해, 데이터를 선형 변환하거나 반전시키는 Data Augmentation 를 적용하였고, 기존 신경망에서 매번 학습을 수행할 때마다 임의로 일정 비율의 노드를 제거하여 학습을 수행하는 Dropout 기법을 실제로 적용하였다.

2.2.2 GoogleNet

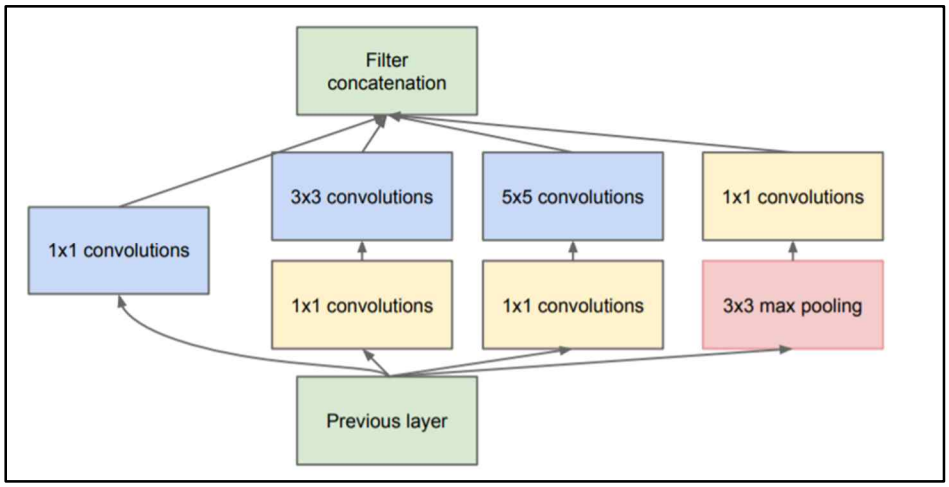
GoogleNet 은 Lin 의 NIN(Network in Network)[9] 아이디어를 차용하여 만든 Inception 모듈을 사용하는 모델이다. 먼저 [그림 2-7]을 통해 기존 Convolution Layer 와 NIN 을 비교한다. NIN 은 기존 Convolution Layer 에서 수행하는 Convolution 연산 후 Feature Map 을 출력하는 과정과 달리, Convolution 연산을 MLP(Multi-Layer Perceptron)로 대체하여 역 전파 과정을 거쳐 Feature Map 을 출력한다.



[그림 2-7] 기존 Convolution Layer 와 NIN 의 비교

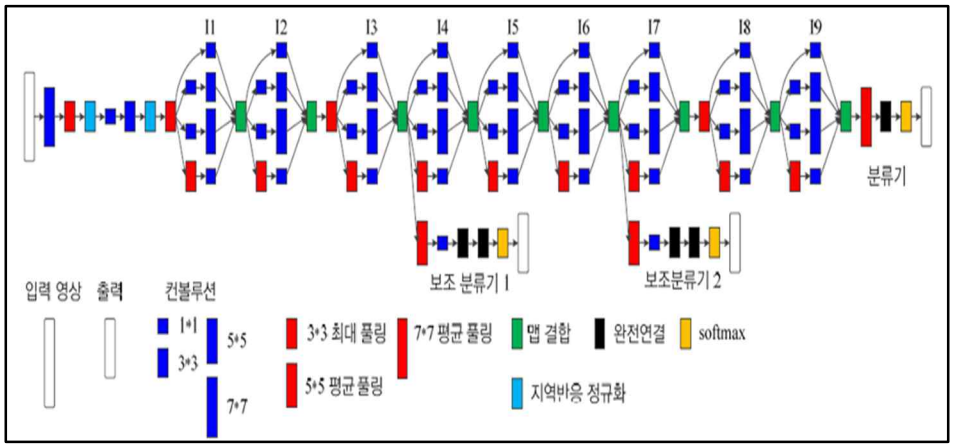
또한 기존 CNN 구조와 달리 Fully-Connected Layer 대신 전역 평균 풀링을 통해 매개변수를 줄임으로서 Fully-Connected Layer 가 가지는 단점인 과도한 매개변수 수로 인한 과잉적합을 방지하고자 하였다.

GoogleNet 은 이러한 NIN 의 아이디어를 확장하여 Inception 모듈을 제안하였으며, 9 개의 Inception 모듈을 결합하여 매개변수의 수를 줄이고자 하였다. [그림 2-8]은 Inception 모듈의 구조로서 NIN 과는 달리 Convolution 연산만으로 내부 네트워크가 이루어져 있으며, 4 종류의 Convolution 연산을 수행하고 결과를 결합한다.



[그림 2-8] Inception Module

4 종류의 연산 중, 3x3, 5x5 Convolution 과 3x3 max pooling 연산 전에 1x1 Convolution 연산을 수행하는데, 이는 Inception 모듈의 핵심 기법으로, 1x1 Convolution 연산을 먼저 수행함으로써 Feature Map 의 수를 줄이게 되어 이후 연산에서 필요한 연산 량을 줄였다.



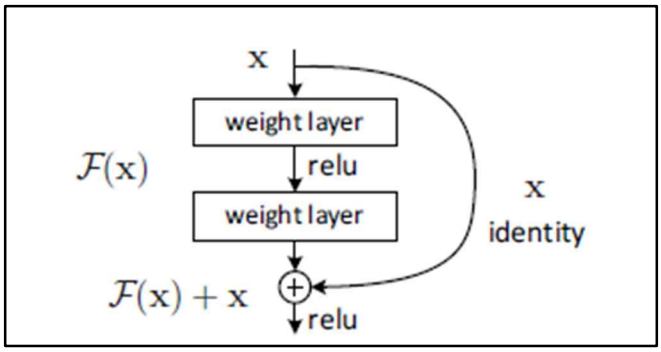
[그림 2-9] GoogleNet 구조

[그림 2-9]은 GoogleNet 의 구조로, 9 개의 Inception 모듈과, 1 개의 Fully-Connected Layer 를 둔 22 개의 Layer 를 가지는 네트워크이다. 깊은 망을 가짐에도 불구하고, Inception 모듈의 사용으로 인해 망이 깊어질수록 생기는 gradient 소멸 문제나 과잉 적합 문제를 해결하였다.

2.2.3 ResNet

ResNet 은 잔류 학습 기법을 사용하여 깊은 망을 구현한 네트워크이다.

망이 깊어질수록 생기는 문제인 Gradient 소멸 문제와 증가한 매개변수 수로 인한 과잉 적합 문제를 Inception 모듈을 사용하여 해결한 GoogleNet 과는 달리, ResNet 은 잔류 학습(Residual Learning)이라는 기법을 사용하였는데, 이를 통해 성능 저하를 피하면서 Layer 를 최대 1,202 까지 늘렸다. [그림 2-10]는 잔류 학습의 구조를 나타내며, 식 (2-3)과 잔류 학습을 나타내는 식 (2-4)을 통해 설명한다. 식 (2-3)과 식(2-4)에서 $\circledast$는 Convolution 연산, $\mathbf{x}$ 는 입력, τ는 활성함수 ReLU 라고 가정한다.



[그림 2-10] 잔류 학습의 구조

$$\mathbf{F}(\mathbf{x}) = \boldsymbol{\tau}(\mathbf{x} \circledast \mathbf{w}_1) \circledast \mathbf{w}_2 \qquad \textbf{(2-3)}$$

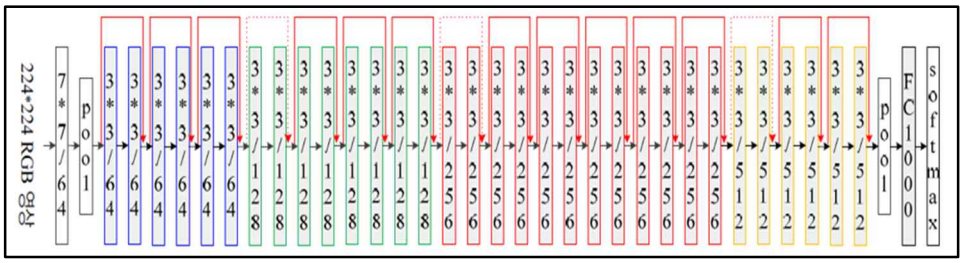
$$\mathbf{y} = \boldsymbol{\tau}(\mathbf{F}(\mathbf{x}) + \mathbf{x}) \qquad \textbf{(2-4)}$$

[그림 2-10]에서 **x** 는 2 개의 Layer 를 거쳐 **F(x)**가 된다. 이를 식 (2-3)으로 나타냈으며, 이후 기존 Network 에서는 활성 함수 ReLU 를 거쳐 다음 Layer 로 전달한다. 하지만 잔류 학습은 식 (2-4)와 같이, 기존 **F(x)**에 지름길 연결을 통해 **x** 를 더한 **F(x)+x** 에 ReLU 를 적용하여 다음 Layer 에 전달한다. 이 과정을 잔류 학습이라고 하며, 이때 **F(x)**를 잔류라 한다.

ResNet 이 잔류 학습을 사용한 이유가 Gradient 소멸 문제를 해결하기 위해서라고 하였는데 이를 해당 Layer 의 Gradient 를 계산하는 식 (2-5)를 통해 설명한다. 이때 ε은 목적함수, l과 L 은 Layer 를 나타내며 L 이 이후 Layer 이다.

$$\frac{\partial \mathcal{E}}{\partial \mathbf{x}_l} = \frac{\partial \mathcal{E}}{\partial \mathbf{x}_L}\frac{\partial \mathbf{x}_L}{\partial \mathbf{x}_l} = \frac{\partial \mathcal{E}}{\partial \mathbf{x}_L}\left(1 + \frac{\partial}{\partial \mathbf{x}_l}\sum_{i=l}^{L-1}\mathbf{F}(\mathbf{x}_i)\right) \quad \textbf{(2-5)}$$

식 (2-5)에서, 역 전파되는 Gradient 가 $\frac{\partial \boldsymbol{\varepsilon}}{\partial \boldsymbol{X_L}}$과 $\frac{\partial \boldsymbol{\varepsilon}}{\partial \boldsymbol{X_L}}\frac{\partial}{\partial \boldsymbol{X_l}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{L-1}}\boldsymbol{F}(\boldsymbol{X_i})$ 로 이루어져 있다. 이때 Gradient 가 0 이 되기 위해서는 $\frac{\partial}{\partial \boldsymbol{X_l}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{L-1}}\boldsymbol{F}(\boldsymbol{X_i})$가 -1 이 되어야 하지만, 이전의 결과에 x 가 더해지는 잔류 학습의 특성상 가능성이 거의 존재하지 않는다. 즉, ResNet 에서 Gradient 소멸 문제는 발생하지 않는다.

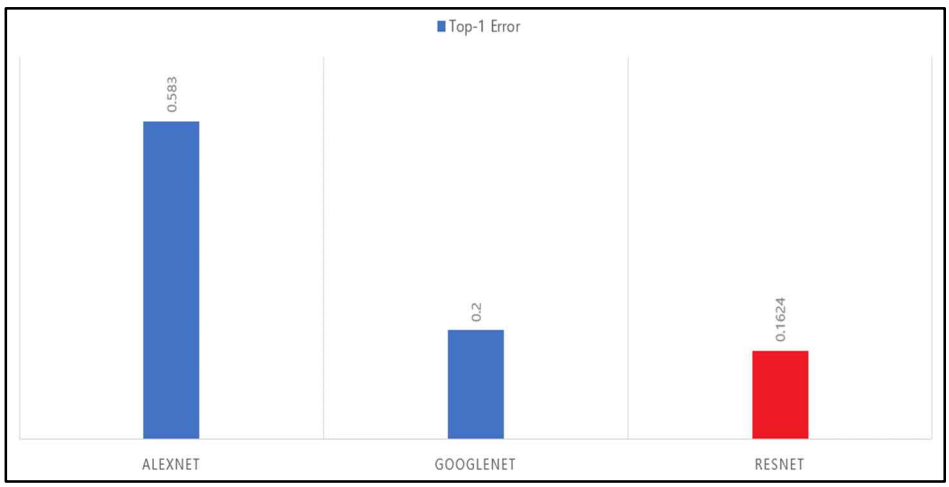


[그림 2-11] ResNet-34 Layer 구조

Gradient 소멸 문제를 잔류 학습을 통해 해결한 Resnet 은 이전 CNN 모델보다 훨씬 깊은 망 구조를 가지게 되었다. ResNet 은 34, 50, 101, 152, 1,202 개의 Layer 로 성능을 시험하였는데, 152 개의 Layer 로 구성된 ResNet 이 2015 년 ILSVRC 에서 Top-5 error 4.49%를 달성하며 우승하였다. [그림 2-10]은 34 개의 Layer 로 이루어진 ResNet 예시로서, 2 개의 Convolution Layer 마다 지름길 연결을 두었고 ,224x224 의 3 채널 RGB 영상을 입력하여 softmax 로 결과를 출력하는 구조이다. Fully-Connected Layer 를 1 개 사용하는 대신 전역 평균 풀링 Layer 를 추가하였고, 모든 Layer 에서 연산 결과에 배치 정규화를 진행한 후 ReLU 를 적용하였다.

2.3 CNN 모델 선정

본 차종 판별 시스템 구현에 사용할 CNN 학습 모델을 선정하기 위해, 다양한 네트워크 모델을 지정하여 성능 비교를 진행하였다. 본 시스템에서 사용할 오픈 소스 데이터셋의 차종이 196 종으로 적지 않은 수로 이루어져 있고, 획일적으로 구성되어 작은 차이가 차종을 구별하는 차의 특성상 독자적으로 구성한 CNN 모델로 학습하기에는 무리가 있다고 판단하여 2.2 절에서 설명한 AlexNet, GoogleNet, ResNet-152 를 후보로 선정하였다. 오픈 소스 데이터 셋인 Stanford Cars Dataset 에서 제공하는 Train Dataset 과 Test Dataset 을 변형없이 학습에 사용하였고, 모델이 답으로 판별한 차종을 정답으로 판별하지 못한 Error 를 나타내는 Top-1 Error 를 기준으로 성능을 비교하여 [그림 2-12]에 명시하였다.



[그림 2-12] 모델 별 Top-1 Error

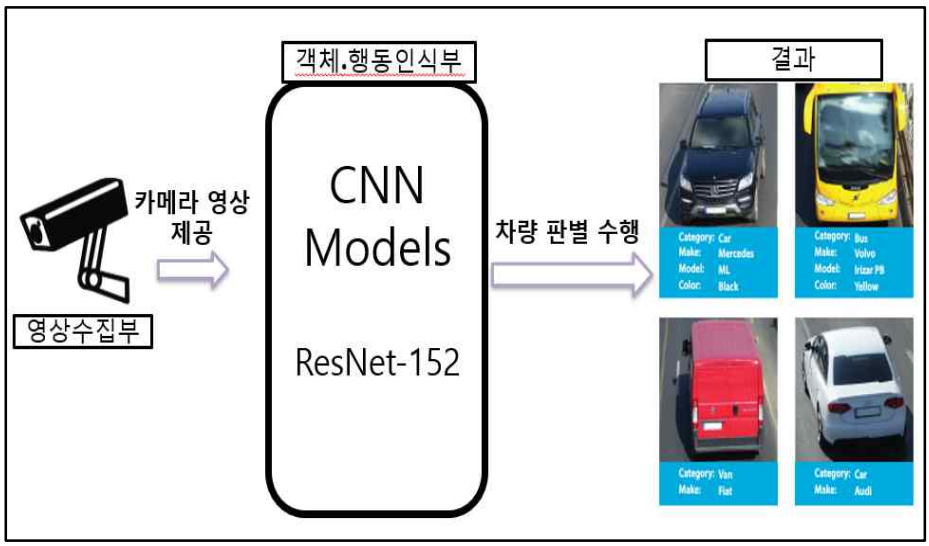
성능 비교 결과 ResNet-152 가 Top-1 Error 에서 0.1624 로 가장 낮은 오류율을 보였으며, GoogleNet, AlexNet 순으로 오류율이 낮았다. 따라서 본 논문에서는 CNN 학습 모델로 ResNet-152 를 선정하여 사용할 것이다.

3. 차종 판별 시스템의 구현

본장에서는 차종 판별 시스템의 구성도와 작동 방식을 소개하고, 데이터셋의 구성 비율을 조정한 후 학습한 결과를 토대로 문제점을 개선한다. 이후 완성된 시스템으로 실제 차량에 적용하여 결과를 확인한다.

3.1 시스템 구성도

차종 판별 시스템의 구성도는 [그림 3-1]과 같다.



[그림 3-1] 시스템 구성도

해당 시스템은 영상을 수집하는 영상수집부와 수집한 영상을 입력받아 차량을 판별하는 객체•행동인식부로 나뉜다. 영상수집부에서 수집한 영상 프레임은 차량 판별을 위해 224X224 해상도로 변환하며, 변환된 프레임은 객체•행동인식부에서 CNN 사전 학습 모델인 ResNet-152 모델로 차종 판별을 진행하고 결과를 도출한다. 차종 판별을 위한 CNN 모델 학습을 위해 서론에서 명시한 오픈 소스 이미지 데이터

셋인 Stanford Cars Dataset 을 사용하여 진행하며, 정해진 차종에 대한 새로운 이미지를 입력하여 학습이 정상적으로 진행되었는지 확인하고 문제점을 개선함으로써, 시스템을 구현하고자 한다

3.2 데이터 셋 구성 및 모델 학습

Stanford Cars Dataset
✓ 196 classes, 6515 train images, 1629 valid images, 8041 test images, resolution 224X224

[그림 3-2] Stanford Cars Dataset

서론에서 명시한 Dataset 을 사용하여 데이터셋을 구성하였다. 196 종의 차종으로 이루어져 있으며, 224x224 해상도를 가지는 6515 장의 트레인 이미지와 1629 장의 validation 이미지, 8041 장의 test 이미지로 약 50:50 비율로 트레인과 테스트 셋이 구성 되어있다. ImageNet Dataset(약 1400 만 장)이나 COCO Dataset(약 33 만 장) 과 같은 대형 Dataset 으로 학습을 진행한다면 트레인과 테스트셋을 나누는 비율이 크게 학습에 영향을 끼치진 않지만, 위 학습에 사용하는 데이터셋은 2 만장이 채 되지 않아 학습에 있어 트레인 테스트 데이터셋의 비율은 훨씬 overfitting 에 민감하다. 따라서 주어진 Dataset 을 기존 50:50 비율과 다른 비율로 나누어 학습을 진행하였고 [표 3-1]로 나타냈으며, 가장 성능이 좋은 비율을 강조하여 표시하였다.

Ratio	50:50	60:40	70:30	80:20	90:10
Train/ Test	6515/ 8041	9711/ 6474	11330/ 4855	12948/ 3237	14567/ 1618
acc	83.76%	86.31%	88.24%	**91.69%**	90.89%

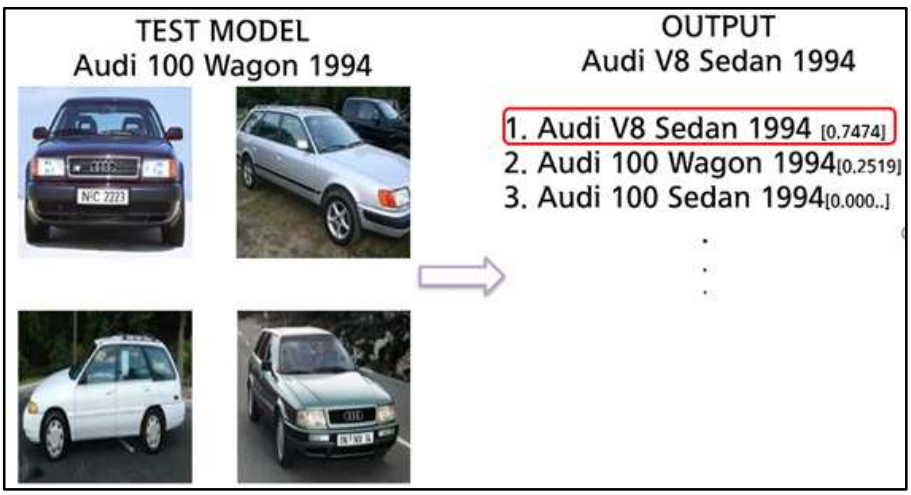
[표 3-1] 비율 성능표

학습을 진행한 후 실시한 테스트는 학습에 사용되지 않은 데이터셋으로 테스트를 진행하였으며, 트레인 셋과 테스트 셋을 80:20 비율로 나눈 데이터셋의 성능이 가장 좋아 데이터셋의 비율을 조정해 재 학습하였다.

3.3 학습 결과 분석 및 개선

앞서 3.2 절에서 진행한 테스트에서, 가장 성능이 좋았던 80:20 비율로 나눈 데이터셋의 테스트에 사용한 1608 장의 이미지 중 오답으로 출력된 130 장의 이미지를 토대로 결과를 분석하였고, 이후 분석된 결과를 바탕으로 데이터셋을 수정해 학습하였다.

3.3.1 결과 분석



[그림 3-3] 오답 예시 1

[그림 3-4] 정답 차종 예시

[그림 3-3]은 130 장의 차량 중 가장 많이 오답을 출력한 차종이다. 해당 차종인 'Audi 100 Wagon 1994'에 대한 정답 차종은 2 위 후보에 위치하였으며, 가장 유력한 차종으로 판단하여 출력한 차종은 같은 브랜드의 같은 차량 군이지만 다른 차종으로 오인하여 출력한 경우로, 오답으로 출력한 차종의 학습에 사용한 이미지 예시인 [그림 3-4]를 보아도 판별이 모호한 것을 확인할 수 있었다. 해당 차종 이외에도 오답 이미지의 약 90%(117 장)는 [그림 3-5]과 같이 같은 브랜드, 같은 차량 군이지만 다른 차종으로 오인하여 출력한 경우이며, 정답 차종의 순위 또한 Top-5 안에 위치하여 판별에 어려움이 있는 것을 확인할 수 있었다.

[그림 3-5] 오답 예시 2

오답 이미지의 90%를 차지하는 오답 유형이외에 나머지 10%를 차지하는 오답 유형에는 테스트 차종과 정답으로 출력된 차종의 구분이 육안으로 보기에 명확함에도 구분이 되지 않는 유형이었으며, 정답 차종의 순위가 Top-5 에 위치하지 못했다. 예시인 [그림 3-5]또한 해당 차종인 'Volkswagen Golf Hatchback 2012' 과 오답 차종인 'Audi S6 Sedan 2011' 과 확연히 구분됨에도 정답 차종의 순위가 9 위에 위치하는 결과를 보였다.

3.3.2 결과 개선

3.3.1 절에서 제기한 문제를 개선하기 위해 먼저 정답 차종의 순위가 5 순위를 초과한 차종에 대해 Train Dataset 을 확인하였다. [그림 3-6]은 Train Dataset 에 포함된 학습에 부적절하다고 판단되는 이미지이며, 정상적인 Image 로 대체하였다.

[그림 3-6] 불량 데이터 교체

또한 다른 문제로 대두된 같은 브랜드 차량간 차종 판별 문제에서, 같은 차종 (예: SUV, 세단, Coupe)간의 판별 문제의 개선 방안으로 같은 연식의 차량 또는 같은 차체를 가지는 차량을 병합하여 차종을 196 종에서 120 종으로 압축함으로써, 학습을 진행하였다.

	Dataset ratio	Car Classes	Top-5 acc
AlexNet	50:50	196	0.417
GoogleNet	50:50	196	0.8
ResNet-152	50:50	196	0.8376
	60:40	196	0.8631
	70:30	196	0.8824
	90:10	196	0.9089
	80:20	196	0.9169
	80:20	120	**0.9730**

[표 3-2] 모델 성능표

본 논문에서 학습한 모델들과 개선한 모델과의 성능을 비교하여 [표 3-2]로 표시하였다. 불량한 이미지를 대체하고, 차종을 196 종에서 120 종으로 압축하여 학습한 결과, 정답을 5 순위 안에 맞추는 모델의 성능이 약 91.7%에서 97.3%로 5.7%가량의 성능 개선이 이루어졌으며, 초기의 ResNet-152 학습 모델과는 약 13%의 개선이 이루어진 것을 확인할 수 있었다.

3.3.3 실생활 적용

3.3.2 절에서 개선을 진행하여 얻은 학습 모델을 사용하여 차종 판별 시스템이 실생활에 적용 가능한지를 확인하기 위해 국내에서 운행중인 차량 중, 시스템에서 분류가능한 차종을 대상으로 실험을 진행하였다. 실생활에서 접하기 쉬운 차종 5 가지를 특정하여 사진을 촬영하였고, 각 차종 마다 50 장 정도의 데이터를 수집할 수 있었다. 이후 수집한 데이터의 해상도를 시스템의 입력하기 위해 224x224 로 일괄 조정하였고, 데이터에 따른 차종을 분류 후 판별을 진행하였다. 시스템의 차종 판별을 진행한 결과를 차종 별로 [표 3-3]에 정리하였다.

	Top-1 err	Top-5 err	acc avg.
Hyundai Veloster Hatchback 2012	0.12	0.06	0.8413
Hyundai Santa Fe SUV 2012	0.0652	0.0217	0.9231
Hyundai Elantra Sedan 2007	0.4258	0.1296	0.5429
Hyundai Accent Sedan 2012	0.2353	0.1372	0.7682
Hyundai Azera Sedan 2012	0.2037	0.0741	0.7773
TOTAL	**0.261**	**0.1314**	**0.7107**

[표 3-3] 차종별 성능표

[표 3-3]을 보면, 시스템이 답으로 출력한 차종들 중에 정답 차종이 Top-5 안에 위치하지 못한 Error 를 나타내는 Top-5 Error 가 평균 0.1314 로, 우수한 성능을 보였다. 또한, 시스템이 답으로 출력한 차종의 확률을 나타내는 acc avg 도 평균 0.7107 로 준수한 성능을

보였다. ‘Hyundai Veloster Hatchback 2012’와, ‘Hyundai Azera Sedan 2012’, ‘Hyundai Santa Fe SUV 2012’은 Top-5 Error 가 시스템에 쓰인 학습 모델의 성능과 비슷하거나 더 우수한 결과를 보였다. 하지만, ‘Hyundai Elantra Sedan 2007’은 Top-5 Error 에서는 준수한 성능을 보였으나, Top-1 Error 에서 상대적으로 낮은 결과를 보였다.

4. 결론 및 향후 연구 과제

차종을 판별하는 데는 많은 기술적인 방법들이 존재하지만, 시간에 관계없이 지속적으로 정확하게 차종을 판별할 수 있어야 더 활용성이 높은 방법이라고 할 수 있다.

본 논문에서 차종을 판별하기 위해 사용한 CNN 은 Layer 의 구성 방법이나, Hyper-Parameter 의 조정 방법, 모델의 학습 방법, 학습에 사용하는 데이터셋의 구성 방법 등, 기술을 사용하기 위해 제어해야 하는 변수가 많고, 피드백을 위한 학습 시간이 상당하다는 단점이 존재하지만, 알맞은 변수의 제어와 충분한 시간이 존재한다면, 우수한 성능을 가지는 모델을 추출할 수 있음을 증명하였다. 최종 학습된 모델의 성능은 해당하는 차종 내에서 Top-5 Error 0.027 로, 테스트 데이터 97.3%의 정답 차종을 5 순위 안에 출력하였다. 또한, 국내에서 자주 접할 수 있는 차종 5 종을 특정하여 판별을 진행한 결과에서도 준수한 성능을 보여 실질적인 차종 판별 방법으로는 신뢰도가 높은 방법이라고 여겨진다. 이는 각종 이유로 설치된 CCTV 의 관리를 더욱 용이하게 만들 수 있으며, 한정된 인력으로도 다수의 CCTV 를 관리하고 모니터링 할 수 있을 것이다.

그러나 본 논문에서 구현한 시스템에서는 아직 결점이 존재한다. 16,185 장으로 구성된 작은 데이터셋으로 차종을 한정하여 시스템을 구현하였기 때문에 한정 차종 이외에서의 성능을 알지 못하며, 데이터 부족으로 충분한 차종 별 특징을 학습하지 못하여 같은 브랜드 내의 같은 차량 종류 간의 판별이나, 유사한 형태의 차종 간의 판별에서 많은 어려움을 보였다.

데이터셋의 한계를 극복하기 위해선, 직접 이미지를 검색하여 차종 별 데이터를 수집하는 방법이 있겠지만, 이를 혼자 진행하기에는 엄청난 시간과 노력을 필요로 하는 소모적인 해결책이다. 따라서 다른 방법으로 더 방대한

데이터 셋을 마련하는 방법이 존재하는데, Linjie Yang, Ping Luo, Chen Change Loy, Xiaoou Tang 이 제안한 차량 미세 분류[10] 방법에서 사용한 방대한 크기의 데이터 셋인 'ComCars' Dataset 이 있다. 이 데이터 셋에는 1,716 종의 차종이 존재하며, 총 136,726 장의 데이터로 구성되어 있다. 하지만 이 데이터 셋은 오픈 소스가 아니며, 특히 학생에게는 공유가 제한되어 있어 현재 구성하기에는 어려움이 있기에 차후 데이터 셋이 마련된다면, 본 시스템의 결점을 개선하기에 적당하다고 생각하며, 더 진보된 시스템이 구현될 것이라 생각한다.

참고문헌

[1] NIJ. <Ministry of Justice, USA>. Buyer Beware Vol.10 (11, 2002)

[2] Derrick Liu, Yushi Wang. Image Classification of Vehicle Make and Model Using Convolutional Neural Networks and Transfer Learning. Stanford University. (2015)

[3] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg. SSD: Single Shot MultiBox Detector. UNC Chapel Hill, Zoox Inc. Google Inc. University of Michigan, Ann-Arbor. (29 Dec 2016)

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Deep Residual Learning for Image Recognition. (10 Dec 2015)

[5] Burak Satar, Ahmet Emir Dirik. 'Deep Learning Based Vehicle Make-Model Classification'. Uludag University, Bursa, Turkey (9 Feb 2019)

[6] Jong Taek Lee, Yunsu Chung. Deep Learning-based Vehicle Classification using an Ensemble of Local Expert and Global Networks. Electronics and Telecommunications Research Institute (ETRI), Daegu, South Korea (2017)

[7] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. University of Toronto. 2012

[8] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, Zbigniew Wojna. Rethinking the Inception Architecture for Computer Vision. Google Inc. 2015

[9] Yann LeCun, Patrick Haffner, Léon Bottou, and Yoshua Bengio. Object Recognition with Gradient-Based Learning. AT&T Shannon Lab, 100 Schulz Drive, Red Bank NJ 07701, USA. 1999

[10] Linjie Yang, Ping Luo, Chen Change Loy, Xiaoou Tang. A Large-Scale Car Dataset for Fine-Grained Categorization and Verification Department of Information Engineering. The Chinese University of Hong Kong Shenzhen Key Lab of CVPR, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China (arXiv:1506.08959v2 [cs.CV] 24 Sep 2015).

[11] 신동, 김병만 (2016). Convolutional Neural Network 와 Tensorflow 를 활용한 차량 모델 판별. 한국정보과학회 학술발표논문집, 2074-2076.

[12] Adami Fatima Zohra, Salmi Kamilia, Abbas Fayçal, Saadi Souad. Detection And Classification Of Vehicles Using Deep Learning. Department Of Computer Science University of Abbes Laghrour khenchela, Algeria. International Journal of Computer Science Trends and Technology (IJCST) – Volume 6 Issue 3, May - June 2018.

[13] Ms. Vijayasanthi D,Mrs. Geetha S. DEEP LEARNING APPROACH MODEL FOR VEHICLE CLASSIFICATION USING ARTIFICIAL NEURAL NETWORK, International Research Journal of Engineering and Technology (IRJET), Volume: 04 Issue: 06, June -2017

[14] Jong Taek Lee and Yunsu Chung. Deep Learning-based Vehicle Classification using an Ensemble of Local Expert and Global Networks. Electronics and Telecommunications Research Institute (ETRI), Daegu, South Korea. 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)

[15] Jonathan Krause, Jia Deng, Michael Stark, and Li Fei-Fei. Collecting a Large-Scale Dataset of Fine-Grained Cars. Computer Science Department, Stanford University. Second Workshop on Fine-Grained Visual Categorization (FGVC2). Portland, OR. June 28, 2013.

[16] Timnit Gebru, Jonathan Krause, Yilun Wang, Duyun Chen, Jia Deng, Li Fei-Fei. Fine-Grained Car Detection for Visual Census Estimation. Department of Computer Science, Stanford University. AAAI Conference on Artificial Intelligence AAAI. 2017.