정보통신공학과 졸업 논문

CNN을 이용한 차종 판별 시스템 구현

Implementation of Car Classification System

using Convolutional Neural Network

한국외국어대학교

정보통신공학과

201402750

임광효

지도 교수 : 한 희 일

목 차

[1. 서론 4](#_Toc12357733)

[2. 관련 기술 조사 6](#_Toc12357734)

[2.1 CNN 6](#_Toc12357735)

[3. 본론 7](#_Toc12357736)

[3.1 차종 판별 시스템 7](#_Toc12357737)

[3.1.1 시스템 구성도 7](#_Toc12357738)

[3.1.2 데이터 구성에 따른 학습 결과 8](#_Toc12357739)

[3.1.2.1 1차 성능 개선 10](#_Toc12357740)

[3.1.2.2 중간 결과 13](#_Toc12357741)

[3.1.2.3 2차 성능 개선 14](#_Toc12357742)

[3.1.2.4 2차 결과 16](#_Toc12357743)

[3.1.3 모델에 따른 학습 결과 18](#_Toc12357744)

[4.결론 및 향후 연구 과제 19](#_Toc12357745)

[참고문헌 20](#_Toc12357746)

그 림 목 차

[그림 1-1] 연도별 공공기관 CCTV 설치 현황………………………………………………………...4

[그림 1-2] 수동 영상관제 시간에 따른 위험 탐지율………………………………………………4

[그림 2-1] CNN의 부분연결 구조………………………………………………………………………….6

[그림 2-2] CNN 구조……………………………………………………………………………………………..6

[그림3-1] 시스템 구성도……………………………………………………………….……………………..7

[그림3-2] Stanford Cars Dataset…………………………………………………………………..……….8

[그림 3-3] Loss rate…………………………………………………………………………………….………..8

[그림 3-4] 학습 결과……………………………………………………………………………….……………8

[그림 3-5] 변형된 데이터 셋…………………………………………………………………….…………...9

[그림 3-6] 변형된 데이터셋 학습 결과………………………………………………………….……….9

[그림 3-7] 차종 판별 예시…………………………………………………………………….……………..10

[그림 3-8] 차종 판별 오답…………………………………………………………………………………..10

[그림 3-9] 오답 차종 이미지………………………………………………………………….……………11

[그림3-10] 차종 판별 중 특이한 오답………………………………………………….…………...…12

[그림 3-11] 불량 데이터 교체…………………………………………………………….……………….14

[그림 3-12] 기존 모델과 개선된 모델과의 성능 비교…………………………………………..15

[그림 3-13] 가장 낮은 출력을 보인 차종의 비교…………………………….…………………….16

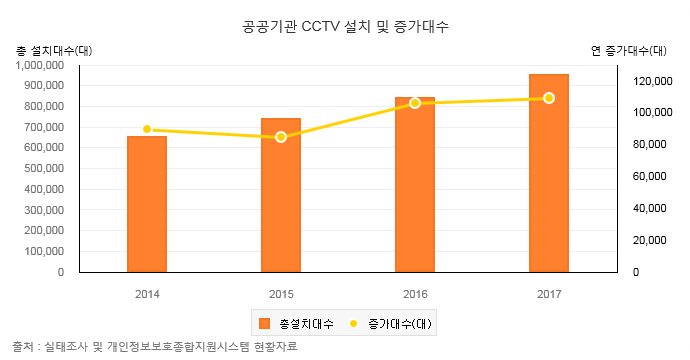
[그림 3-14] 정답이Top-5 이상의 순위로 출력된 오답 이미지……………………………..17

[그림 3-15] 모델 별 성능 비교………………………………………………………………….…………18

요 약

# 서론

오늘날 각종 범죄 예방 및 교통 정보 수집 등의 이유로 CCTV 설치 대수는 매년 증가하는 추세이나, 한정된 인력이 매년 증가하는 CCTV를 효율적으로 관리하고 모니터링하기에는 한계가 존재한다. 미 국립사법연구소(NIJ)의 보고서[1]에 의하면, 관제 요원이 직접 CCTV를 통해 상황을 감시하는 기존 CCTV 시스템에서는 상황을 감시하기 시작한 후, 22분이 지나면, 위험을 탐지하는 능력이 약 5%로 현저히 떨어진다는 실험결과가 존재한다. 따라서 이에 대한 해결 방안으로, 한정된 인력으로도 다수의 CCTV를 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 CNN을 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 Liu와 Yushi Wang은 196종의 차종을 특정하고, 이를 판별하기 위해 SVM과 1-layer CNN를 기반으로 한 모델과 GoogLeNet, VGGNet, CaffeNet을 각각 사전 학습, 부분 학습, 완전 학습으로 나누어 학습한 모델들을 제안하였으며[2], Burak Satar와 Ahmet Emir Dirik은 Detector인 SSD[3]와 ResNet[4]를 사용한 모델을 제안하였다[5]. SVM과 1-layer CNN을 사용한 모델은 구조가 단순하고 간단하여 적은 시간으로도 학습이 가능하였으나, 100여종이 넘는 차종을 판별하기에는 데이터셋의 크기가 작고 모델의 깊이가 충분히 깊지 못해 overfitting 문제가 발생하였다. SSD와 ResNet을 조합한 모델은 Detector를 통해 먼저 차량을 인식한 후 학습한 모델로, Detector를 사용하지 않은 모델보다 높은 성능을 보였다. 하지만 Detector를 통해 먼저 차량을 인식하는 과정이 완벽하게 수행되지 않아 차종 판별에 있어 Detector의 성능이 모델의 성능에 크게 영향을 끼치는 단점이 존재했으며, 판별하는 차종의 수가 7종으로 매우 적었다. 또한 Lee 와 Chung이 제안한 또 다른 모델[6]은 차량을 소그룹으로 묶은 데이터셋을 각각 학습한 local 모델과 차량 전체를 데이터셋으로 학습한 global 모델로 네트워크를 이루고 Softmax로 결과를 출력하는 모델이다. 성능은 최저 68.72%에서 최고 99.66%으로 매우 정확하지만, 해당 모델을 위해서 12개의 local 모델과 6개의 global 모델을 사용해 엄청난 계산 능력이 요구되어 다수의 고 사양 GPU를 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.

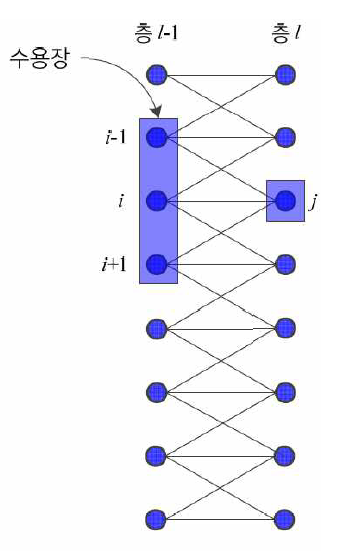
“””학습을 진행하였을 때 발생하기 쉬운 overfitting 문제에 있어 dropout을 사용해 문제를 해결하고자 하였다.”””

본 논문의 구현 환경은 Windows와 Linux환경에서Python기반 가상 개발 환경인 Anaconda를 사용하여 진행하며, 해당 학습을 위해 오픈 소스 이미지 데이터 셋인 Stanford Cars Dataset을 224\*224 해상도로 일괄 조정한 이미지를 사용한다. 학습 방법은 ImageNet 데이터셋으로 사전 학습된 ResNet 모델을 사용해 “””Layer를 조정해가며””” 학습한다.

본 논문의 구성은 다음과 같다. 2장에서는 학습에 사용되는 핵심 기술인 CNN과 주요 CNN모델에 대한 특징을 기술하고, 성능을 비교하여 시스템에 적합한 CNN 모델을 선정한다. 3장에서는 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, 개선된 실험 결과와 이전 실험 결과를 비교 분석한다. 마지막으로 4장에서는 개선된 시스템으로 실제 차량에 대한 차종 분류를 진행하여 분석한 결과를 토대로 결론을 짓고 향후 연구 과제를 제시한다.

1. 기술조사
   1. CNN

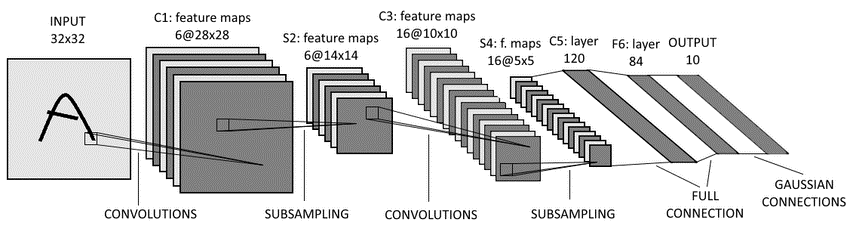
CNN(Convolution Neural Network)은 Convolution 연산을 이용한 부분연결 구조로 복잡도를 크게 낮췄으며, 연산을 위한 입력 데이터의 특징을 직접 추출하여 학습을 해야 하는 머신 러닝과는 달리, 입력 데이터의 특징을 해당 네트워크가 추출하여 학습에 사용하는 딥러닝 구조이다.



[그림 2-1] CNN의 부분연결 구조

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

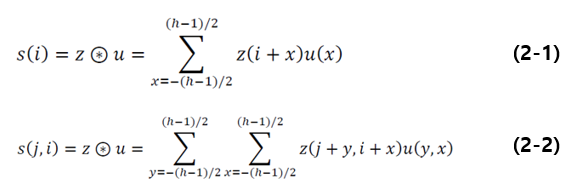
[그림 2-2]는 LeCun이 제안한 LeNet-5[7]로서, 오늘날 쓰이는 CNN 구조의 근간이 되는 네트워크이다. LeNet-5의 구조를 사용하여 CNN을 설명한다.



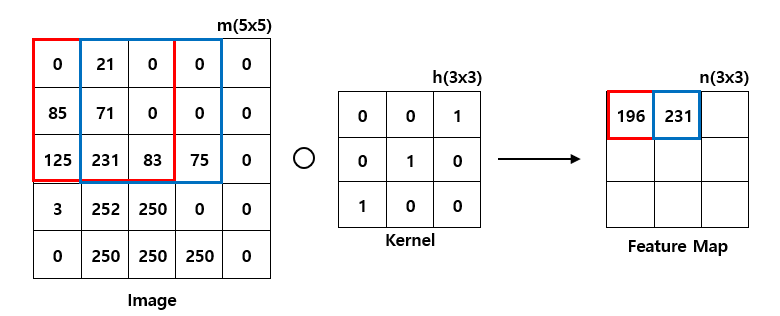
[그림 2-2] LeNet-5 구조

CNN은 크게 특징을 추출하는 Convolution Layer와 Topology 변화의 영향을 받지 않게 해주는 Pooling Layer로 구성되어 있다. LeNet-5 에서는 3번의 Convolution과 2번의 Pooling을 거치며, Convolution과 pooling 과정을 거쳐 남은 강인한 특징을 통해 결과를 분류하기 위해 Fully-Connected Layer를 추가하여 이미지를 구별한다.

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

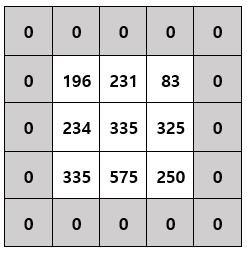


식 (2-1)은 1차원에서의 Convolution 연산이며, u는 커널, h는 커널의 크기, z는 입력, s는 출력이다. 이때의 출력 s를 Feature Map이라고 한다. 식 (2-2)는 영상 데이터와 같이 2차원 구조를 가지는 데이터에서의 Convolution 연산의 공식이며, CNN에서 커널의 높이와 너비는 동일하게 설정하므로 크기를 모두 h라고 표기한다. [그림 2-3]은 2차원 구조를 가지는 5x5크기의 데이터와 h=3인 커널과의 Convolution 연산 예시이다. 연산할 빨간 테두리 영역은 이고 커널은 이므로 연산하면 0\*0+21\*0+0\*1+85\*0+71\*1+0\*0+125\*1+231\*0+83\*0=196이다. 같은 방식으로 파란색 테두리 영역에 같은 연산을 하면 231이 나오며, 이후 모든 영역에 연산을 진행하여 Feature Map을 출력한다. 이때의 Feature map 크기는 Convolution 연산하는 커널의 보폭을 1로 했을 때 이미지의 크기 – 커널의 크기 + 1 이다.



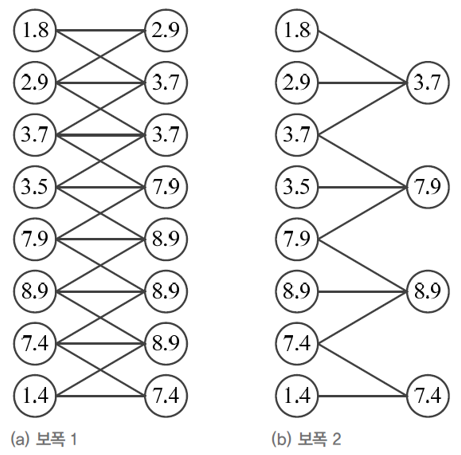
[그림 2-3] Convolution 연산

[그림 2-3]과 같이 Feature Map의 크기는 커널의 크기가 클수록, Convolution 연산을 거칠수록 줄어드는데, 깊은 Layer를 가지는 신경망일수록 Convolution 연산이 여러 번 반복되므로 줄어드는 양이 증가하는 문제가 발생한다. 이러한 문제를 방지하기 위해 padding을 수행하여 크기를 유지한다. [그림 2-4]는 [그림2-3]의 Feature Map을 zero-padding한 결과이다.



[그림 2-4] Zero-padding

Convolution Layer 바로 다음에 Pooling Layer가 따른다. Pooling Layer는 Convolution Layer 출력된 결과를 Subsampling 하는데, CNN에서는 추출된 특징 중 가장 강한 특징을 pooling하는 방식인 max-pooling 방식을 취한다. 이때 보폭을 1로 설정한다면, Feature Map의 크기가 유지되지만, 보폭을 2이상으로 하면 Feature Map의 크기를 줄이는 효과가 있다. [그림2-5]는 보폭을 1과 2로 했을 때의 경우를 나타낸다.



[그림 2-4] max-pooling

여러 번의 Convolution 연산과 Pooling을 거치고 나면, Feature Map의 크기가 작아지면서 동시에 이미지를 대표하는 강인한 특징만 남게 되는데, 이 Feature Map을 입력하여 이미지를 분류하기 위해 Fully-Connected Layer를 거쳐 결과를 추출한다.

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

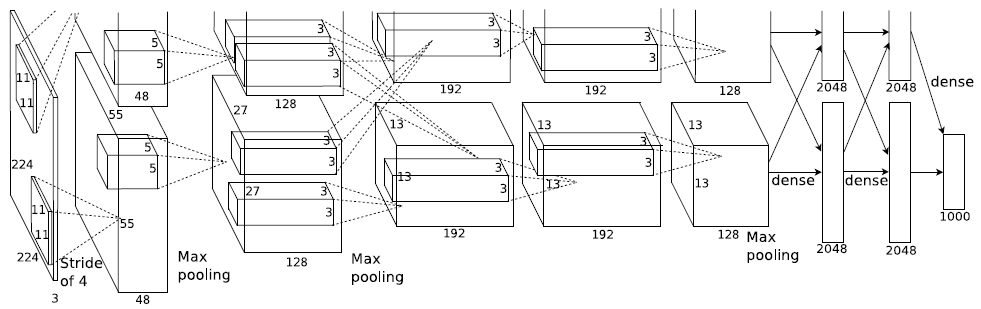
* 1. CNN Models

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

* + 1. AlexNet

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

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



[그림 2-5] AlexNet 구조

복잡한 신경망을 가지는 만큼, AlexNet은 원활한 학습을 위해 여러 가지 기법을 사용하였다. 먼저 학습을 위해 2대의 GPU를 사용하여 학습 시간을 단축하였고, 기존에 쓰이는 활성함수인 tanh 대신 ReLU를 사용하였다. 또한 Convolution 연산의 결과를 이웃 커널의 값을 고려해 조정하는Local Response Normalization을 사용하였다. AlexNet이 방대한 매개변수를 가지는 깊은 구조인 만큼, 과잉 적합 문제 발생을 방지하기 위해, 데이터를 선형 변환하거나 반전시키는 Data Augmentation를 적용하였고, 기존 신경망에서 매번 학습을 수행할 때 마다 임의로 일정 비율의 노드를 제거하여 학습을 수행하는 Dropout 기법을 실제로 적용하였다.

* + 1. GoogleNet

GoogleNet은 Lin의 NIN[9] 을 차용하여 만든 Inception 모듈을 사용하는 모델이다.

* + 1. ResNet

ResNet은 망이 깊어질수록 생기는 문제인 Gradient 소멸 문제를

Alexnet, inceptionv2(기존사용),resnet 3개

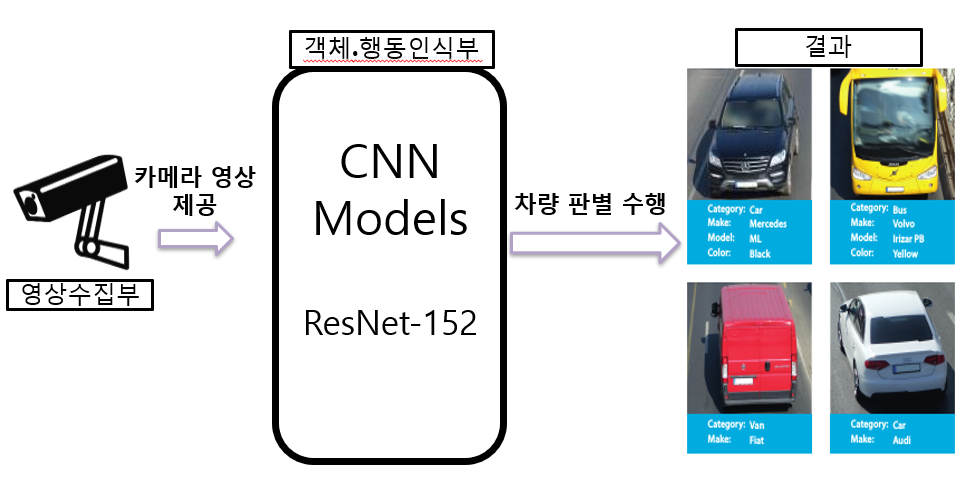
다른 네트워크 모델을 사용했을 때 무엇이 문제인지 -> resnet residual learning의 장점과 기존 모델의 gradient 소멸문제

* 1. 성능비교, 책정 기준

다른 네트워크 모델과 resnet 성능 비교하여 최종적으로 renset사용하는 정당성 부여

1. 차종 판별 시스템의 구현
   1. 시스템 구성도

차종 판별 시스템의 구성도는 다음과 같다.



[그림3-1] 시스템 구성도

해당 시스템은 영상을 수집하는 영상수집부와 수집한 영상을 입력 받아 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 오픈 소스 이미지 데이터 셋인Stanford Cars Dataset을 사용하여 진행하며, 정해진 차종에 대한 새로운 이미지를 입력하여 학습이 정상적으로 진행되었는지 확인하고 문제점을 개선함으로써, 시스템을 구현하고자 한다.

* 1. 데이터 셋 구성 및 모델 학습



[그림3-2] Stanford Cars Dataset

3.1에서 명시한 dataset을 사용하여 데이터 셋을 구성하였다. 196종의 차종으로 이루어져 있으며, 224x224 해상도를 가지는6515장의 트레인 이미지와 1629장의 validation이미지, 8041장의 test 이미지로 약 50:50 비율로 트레인과 테스트 셋이 구성 되어있다. ImageNet dataset(약 1400만 장)이나 COCO dataset(약 33만 장) 과 같은 대형 Dataset으로 학습을 진행한다면 트레인과 테스트셋을 나누는 비율이 크게 학습에 영향을 끼치진 않지만, 위 학습에 사용하는 데이터셋은 2만장이 채 되지 않아 학습에 있어 트레인 테스트 데이터 셋의 비율은 훨씬 overfitting에 민감하다. 따라서 주어진 dataset을 기존 50:50 비율과 다른 비율로 나누어 학습을 진행하였다.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 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 좋았다.

* 1. 학습 결과 분석 및 개선

앞서 3.2에서 진행한 테스트에서, 테스트에 사용한 1608장의 이미지 중 130장이 오답을 출력하였으며, 130장을 토대로 결과를 분석하였다.

* + 1. 결과 분석

같은 브랜드간 혼동문제, 같은 차종간 혼동 문제 -> 차종 줄여서 해결로 개선

정답 차종의 순위가 top-5 이상인것 -> data 확인후 대체

기존 resnet에서 제외된 dropout 추가

이외- 이미지가 찌그러진 것 교체

* + 1. 개선된 결과

3.3.1에서 제기된 문제를 개선한 모델의 학습 결과

이후 layer에 따른 성능 비교

* + 1. 최종 시스템

최종 시스템의 데이터셋과 모델 설명

1. 결론

# 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. 신동, 김병만 (2016). Convolutional Neural Network와 Tensorflow를 활용한 차량 모델 판별. 한국정보과학회 학술발표논문집, 2074-2076.
11. 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).
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.