

딥러닝을 활용한 은하 분류

차 정 훈

경희대학교

jhcha08@khu.ac.kr

Galaxy Classification using Deep Learning

Cha Jung Hun
KyungHee University

요약

최근까지 수집된 은하 데이터는 모두 수작업을 통해 형태학적 분류가 이루어졌다. 그러나 시간이 지날수록 축적되는 은하 데이터의 양은 점점 많아질 것이며, 그에 따라 은하의 형태 분류에 소요되는 시간이 증가할 것이다. 이는 곧 은하 연구에 문제점으로 작용할 수 있다. 따라서 본 논문은 딥러닝 모델을 활용하여 은하의 형태를 자동으로 분류하는 방법을 제시한다. 이번 연구에선 ImageNet 데이터셋으로 이미 훈련된 딥러닝 모델인 VGG16, Inception-v3, ResNet-50을 활용하여 세 종류(edge, smooth, spiral)의 은하를 분류하였다. 그 결과 배치 사이즈 64, 에포크 40, 학습률 0.001, 최적화 알고리즘은 Adam을 사용하였을 때 ResNet-50에서 분류 정확도가 88.11%로 가장 높게 기록되었다.

1. 서론



[그림1. 안드로메다은하]

은하는 항성, 밀집성, 성간 물질, 암흑 물질 등이 중력에 의해 묶인 거대한 천체이다. 은하를 이해하는데 있어서 첫 번째 단계는 형태에 따라 은하를 분류하는 것이라고 할 수 있다. 이러한 형태학적 분류는 항상 어느 정도 주관적이지만, 그래도 이것이 은하의 정량적인 특성을 체계적인 방법으로 연구할 수 있는 골격을 마련해 준다.^[1] [그림1]은 안드로메다은하다.

은하는 크게 타원 은하, 나선 은하, 불규칙 은하의 세 가지의 형태로 분류된다. 하지만 이 분류는 밤하늘에서 크고 밝게 빛나는 은하들에 국한하여 사용된다. 굉장히 멀리 있고 작은 은하들은 다른 형태로도 분류가 된다. 대표적으로 edge 형태와 smooth 형태가 있다. edge 형태는 나선 은하이지만 시선 방향에 의해서 뾰족하게 보이는 은하를 말한다. smooth 형태는 은하 디스크의 두께가 얇지 않고 두꺼울 때, 디스크를 두껍게 만드는 요인인 은하 팽대부가 classical 형태가 아닌 pseudo 형태라서 약간의 찌그러진 모습이 있는 은하를 말한다.

2000년부터 현재까지 미국 뉴멕시코주의 아파치 포인트 천문대에서 'SDSS(Sloan Digital Sky Survey)'라는 계획 아래에 수많은 은하 사진이

촬영되었다. 위에서도 언급했듯이, 은하 연구의 첫 번째 단계는 은하를 형태에 따라 분류하는 것이다. 연구자들은 은하 분류에 소요하는 시간을 단축하기 위해 '갤럭시 주(Galaxy Zoo)'라는 웹사이트를 제작하여 시민들이 직접 은하의 모양을 보고 분류할 수 있도록 하였다. 현재는 은하 사진을 보면서 질문에 대답하는 방식으로 은하 분류가 이루어지고 있다. 이처럼 일반적인 은하 분류는 사람이 직접 은하 사진을 하나하나 보고 분류하는 방식으로 이루어진다.

딥러닝은 머신러닝의 특정한 한 분야로서, 2개 이상의 은닉층으로 구성된 심층 신경망이라는 모델을 사용하는 기법이다. 심층 신경망은 데이터로부터 의미 있는 표현을 피드백을 통해 점진적으로 학습하는 알고리즘이며, 심층 신경망에 입력된 데이터 정보는 여러 층을 통과하면서 순도 높게 정제된다. 정제된 정보를 바탕으로 심층 신경망 모델은 예측값을 출력한다. 그다음 손실 함수를 이용해 예측값과 실제값의 차이를 계산한 후, 역전파를 통해 뒤쪽 층에서 앞쪽 층으로 피드백 신호, 즉 손실 함수의 gradient를 돌려준다. 모델은 이 gradient를 참고하여 실제값에 더욱 가깝게 예측값을 출력할 수 있도록 각 층에 존재하는 가중치들을 업데이트하는데, 이를 학습이라고 한다. 딥러닝의 목표는 이전에 쉽게 풀 수 없었던 일들을 잘 수행할 수 있는 좋은 모델을 여러 번의 학습을 통해 얻어내는 것이다.^[2]

사람에게는 자연스럽고 직관적으로 보이지만 기계로는 오랫동안 해결하기 어려웠던 시각과 청각 같은 지각의 문제들을 딥러닝을 통해 해결할 수 있었다. 특히 2015년에 등장한 딥러닝 모델인 ResNet은 사람보다 더 뛰어난 수준인 인식 오류율 3.5%로 1,000개의 클래스를 가진 이미지들의 분류를 할 수 있었다.^[3]

본 논문에서는 이미지 인식 기술 대회인 ILSVRC (ImageNet Large Scale Visual Recognition Challenge)에서 주목받은 분류 모델 VGGNet, Inception(GoogleNet), ResNet을 탐구하고, 이를 이용하여 은하를 자동으로 분류하는 방법을 제안한다. 딥러닝을 활용해 더욱 은하를 효율적으로 분류함으로써 앞으로의 은하 연구에 보탬이 될 것이라 기대한다.

II. 선행 연구

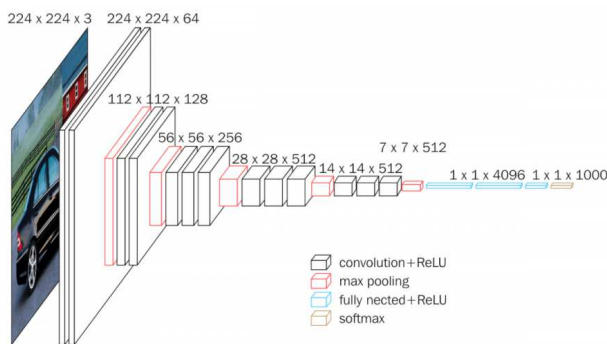
2004년, Calleja 등은 feed-forward 방식의 단순 3층 신경망 모델과 Locally weighted regression 알고리즘을 사용해 세 종류의 은하(타원, 나선, 불규칙) 이미지 310장을 약 91.8%의 정확도로 분류하였다.^[4] 2013년, Marin 등은 나이브-베이지 모델과 랜덤 포레스트 모델에 네 종류의 은하(타원, 나선, 렌즈, 불규칙) 이미지 800장을 훈련해 테스트 데이터들을 약 64.2%의 정확도로 분류하였다.^[5] 2017년, Khalifa 등은 Convolution layer 1개, Pooling layer 1개, Fully-connected layer 1개로 구성된 모델에 세 종류의 은하(타원, 나선, 불규칙) 이미지 808장을 훈련해 테스트 데이터 283장을 97.3%의 정확도로 분류하였다.^[6]

III. 본론

1. 사용 모델

이 장에서는 본 연구에서 사용한 세 가지 딥러닝 모델 VGGNet, Inception(GoogLeNet), ResNet에 대해 간략한 설명을 한다.

1) VGGNet^[7]



[그림2. VGGNet]

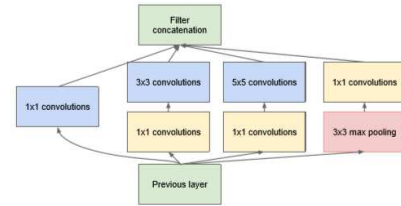
VGGNet은 옥스포드 대학의 연구팀 VGG에 의해 개발된 모델로서, 2014년 ILSVRC에서 준우승했다. 19층 모델의 경우 구조는 [그림2]와 같다. VGGNet의 의의는 모델을 깊게 만들수록, 즉 층을 많이 쌓을수록 모델의 성능이 향상된다는 것을 밝힌 데에 있다. 2012년과 2013년에 우승한 모델은 모델의 층수가 모두 8개였는데, VGGNet은 층수가 11개부터 시작된다.

VGGNet의 6가지 구조 중 대표적으로 VGG16과 VGG19라는 모델이 있는데, 각각 16개, 19개의 층으로 구성된 모델이다. VGGNet은 단순하고 쉬운 모델 구조와 밀리지 않는 성능 덕분에 같은 해에 우승한 모델인 GoogLeNet에 비해 더 널리 쓰이게 되었다. 본 연구에선 VGGNet의 대표적 모델 중 하나인 VGG16을 사용하며, ImageNet 데이터셋에 대한 이미지 인식 오류율은 8.1%다.

2) Inception^[8]



[그림3. GoogLeNet]

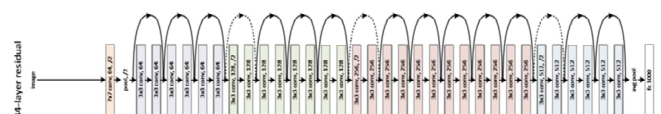


[그림4. Inception 모듈]

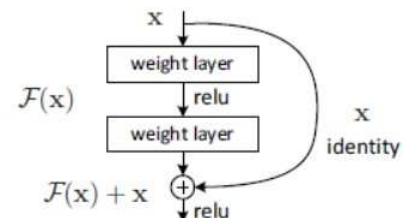
Inception은 구글 연구팀에 의해 개발된 모델로서, 2014년 ILSVRC에서 우승했다. 첫 번째 버전의 경우 구조는 [그림3]과 같다. Inception의 의의는 딥러닝 모델이 깊어질수록 연산량이 증가하는 문제를 해결하기 위해 [그림4]와 같은 'Inception 모듈'이라는 방법을 이용하는 데에 있다.

Inception 모듈은 본래 의도했던 연산을 함과 동시에 parameter 수까지 줄여 연산량도 줄이는 것이 핵심이다. 따라서 Inception은 [그림3]과 같이 깊은 구조를 가짐에도 불구하고 parameter 수가 그리 많지 않다. Inception 모듈을 처음 사용한 Inception은 Inception-v1, 혹은 GoogLeNet이라고도 불리며, 후에 이를 더 발전시킨 모델인 Inception-v2와 Inception-v3가 발표되었다. 본 연구에서 사용할 모델은 Inception-v3이며, ImageNet 데이터셋에 대한 이미지 인식 오류율이 3.58%다.

3) ResNet^[9]



[그림5. ResNet]



[그림6. Residual Block]

ResNet은 마이크로소프트 연구팀에 의해 개발된 모델로서, 2015년 ILSVRC에서 우승했다. 34층 모델의 경우 구조는 [그림5]와 같다. VGGNet, Inception이 발표된 이후에 딥러닝 모델의 깊이가 단순히 깊어질수록 좋은 성능을 낼 것으로 예상했으나 그렇지 않았다. 모델이 깊을수록 손실 함수의 gradient 값이 역전파를 거치며 크게 변형되는 문제 때문이었다. ResNet의 의의는 'Residual Block'이라는 방법으로 이 문제를 어느 정도 해결하여 모델의 성능을 높인 데에 있다. [그림6]의 Residual Block은 'shortcut connection'이라는 기법을 활용한다.

후속 연구에서 ResNet의 더 나은 최적화를 위해 배치 정규화(Batch Normalization)층과 ReLU 활성화 함수 층의 배치에 관한 연구가 진행되었다. 그 결과 배치 정규화 층 - ReLU 활성화 함수 층 - 일반층 순으로 쌓는, 'full pre-activation'이라는 기법을 썼을 때 모델 최적화가 가장 잘 이루어진다는 결론이 나왔다.

ResNet에는 5가지 구조가 있으며, 각각 ResNet-18, ResNet-34, ResNet-50, ResNet-101, ResNet-152이다. 뒤에 붙은 숫자는 각 모델의

층수를 의미한다. 본 연구에서 사용할 모델은 ResNet 후속 연구에서 제안된 기법을 적용한 ResNet-50이며, ImageNet 데이터셋에 대한 이미지 인식 오류율이 5.25%다.

2. 모델 학습



[그림7. 왼쪽부터 순서대로 edge, smooth, spiral 형태의 은하]

활용한 데이터는 세 가지 클래스(edge, smooth, spiral)를 가진 은하 사진이며, 클래스별로 훈련용 데이터 2,000장, 검증용 데이터 300장, 테스트용 데이터 300장씩을 사용하였다. 사진들은 가로와 세로 길이가 각각 300픽셀인 RGB형 이미지 데이터이다. [그림7]은 왼쪽부터 순서대로 edge, smooth, spiral 형태의 예시 은하를 보여준다.

사용한 모델들은 사전에 ImageNet 데이터셋으로 이미 훈련된 VGG16, Inception-v3, ResNet-50이며, 각 모델의 끝에 fully-connected layer를 두 개 붙여서 사용하였다. 본래 세 모델을 직접 구현하고 처음부터 훈련하려고 했으나 시간과 자원이 많이 소모되고, 이미 잘 훈련된 모델을 사용하는 것이 더욱 효율적이라고 판단하여 세 모델 모두 케라스 라이브러리에서 불러와 사용하였다. 기본 훈련 방법에서 사용한 하이퍼 파라미터(hyper parameter) 설정은 다음 [표1]과 같다.

배치 사이즈(batch size)	64
에포크(epochs)	40
학습률(learning rate)	0.001
최적화 알고리즘(optimizer)	Adam
손실 함수(loss function)	categorical cross entropy

[표1. 기본 하이퍼 파라미터]

3. 실험 결과

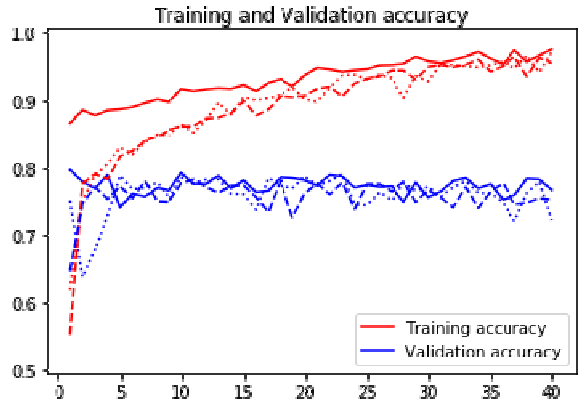
	VGG16	Inception-v3	ResNet-50
1번	79.44%	86.89%	88.11%
2번	76.11%	83.00%	87.33%
3번	76.33%	86.56%	87.67%

[표2. 하이퍼 파라미터 설정과 모델에 따른 실험 결과]

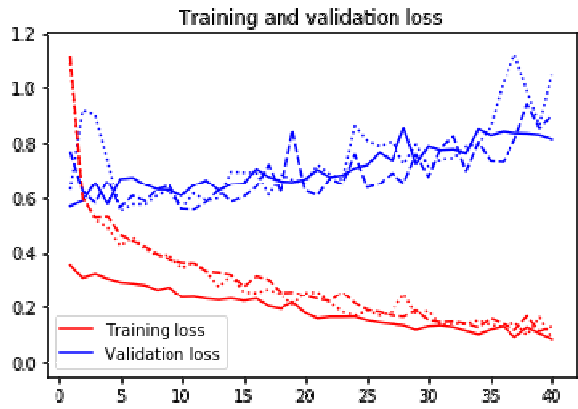
이번 실험은 [표1]의 하이퍼 파라미터를 그대로 사용한 기본 훈련 방법(1번), 기본 방법에서 배치 사이즈만 32로 변경한 훈련 방법(2번), 기본 방법에서 학습률만 0.0005로 변경한 훈련 방법(3번) 총 3가지로 나누어 진행했다. 모델 평가는 테스트 데이터에 대한 정확도(accuracy) 기준으로 이루어졌다. 평가 결과는 [표2]와 같다.

모델 평가 결과, 기본 훈련 방법을 사용한 실험인 1번이 모든 모델에 대해 가장 좋았으며, ResNet-50이 모든 실험 방법에 대해 가장 좋은 성능을 나타냈다. 또한 케라스 공식 홈페이지에 따르면, Top-5 정확도 기준으로 Inception-v3, ResNet-50, VGG16 순으로 성능이 좋다. 이때 Top-5 정

확도는 ImageNet의 검증용 데이터에 대한 성능을 나타낸다. 그러나 이번 실험에서 사용된 세 가지 종류의 은하를 분류하는 문제에서는 ResNet-50, Inception-v3, VGG16 순으로 성능이 좋다는 것을 확인하였다. ImageNet에 공식적으로 포함된 데이터가 아닌, 외부의 데이터이기 때문에 성능 차이가 있는 것으로 보인다.



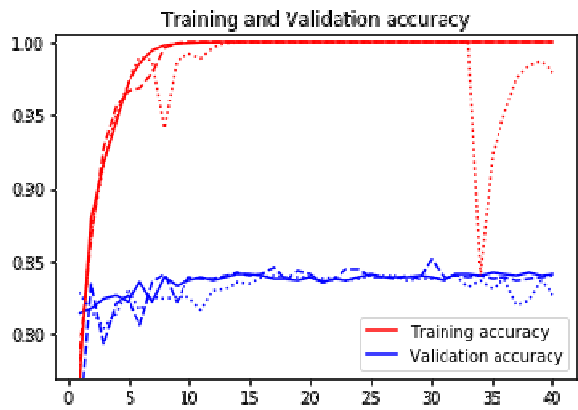
[그림8]



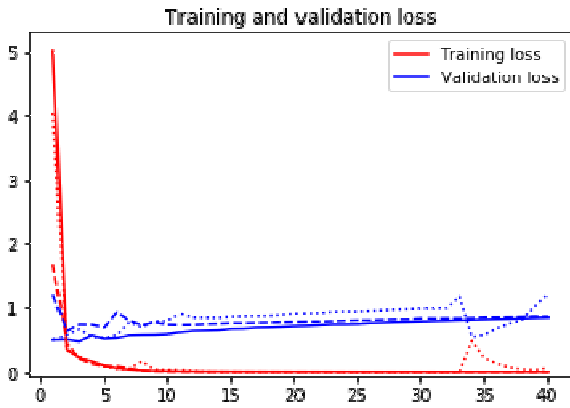
[그림9]

[그림8]부터 [그림13]의 그래프의 붉은 선은 훈련용 데이터, 푸른 선은 검증용 데이터를 뜻하며, 실선은 1번 모델, 얇은 점선은 2번 모델, 굵은 점선은 3번 모델에 해당하는 값들의 추이를 의미한다.

[그림8]은 VGG16의 1, 2, 3번 모델에서 훈련용 데이터에 대한 정확도와 검증용 데이터에 대한 정확도 추이, [그림9]는 VGG16의 1, 2, 3번 모델에서 훈련용 데이터에 대한 손실(loss) 값과 검증용 데이터에 대한 손실 값을 나타낸다.

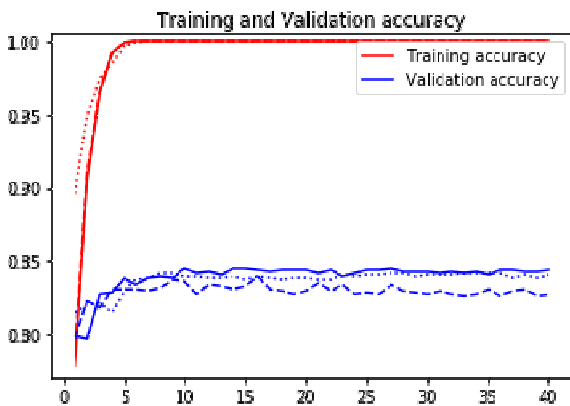


[그림10]

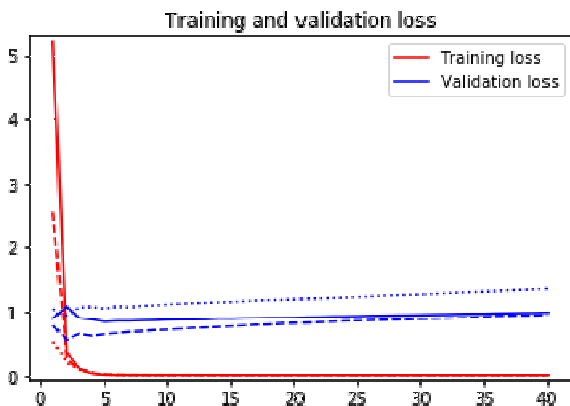


[그림11]

[그림10]은 Inception-v3의 1, 2, 3번 모델에서 훈련용 데이터에 대한 정확도와 검증용 데이터에 대한 정확도 추이, [그림11]은 Inception-v3의 1, 2, 3번 모델에서 훈련용 데이터에 대한 손실 값과 검증용 데이터에 대한 손실 값 추이를 나타낸다. 2번 모델의 정확도가 35 에폭 근처에서 급격히 떨어졌다가 다시 증가한 경우를 제외하고는 정확도 추이와 손실 값 추이가 세 모델 모두 비슷한 것을 볼 수 있다.



[그림12]



[그림13]

[그림12]는 ResNet-50의 1, 2, 3번 모델에서 훈련용 데이터에 대한 정확도와 검증용 데이터에 대한 정확도 추이, [그림13]은 ResNet-50의 1, 2, 3번 모델에서 훈련용 데이터에 대한 손실 값과 검증용 데이터에 대한 손실 값 추이를 나타낸다. Inception-v3과 비슷하게, 정확도 추이와 손실 값 추이가 세 모델 모두 비슷한 것을 볼 수 있다.

IV. 결론

이번 실험에서 훈련 정확도는 세 모델 모두 훈련을 거듭할수록 증가하고, 훈련 손실 값은 감소하는 모습을 보였다. 특히 파라미터가 많은 모델인 Inception-v3, ResNet-50은 일정 훈련 횟수가 지나자 훈련 정확도가 100%에 도달하는 모습을 볼 수 있다. 그러나 검증 정확도는 훈련 횟수가 일정 값 이상이 되면 더 증가하지 않고 진동하는 모습을 보였으며, 검증 손실 값의 경우 VGG16은 훈련을 거듭할수록 많이 증가하고, 나머지 두 모델은 적게 증가하는 모습을 보였다.

이처럼 훈련용 데이터에 대해서만 높은 정확도와 낮은 손실 값을 보이고, 검증용 데이터에 대해서 그렇지 않은 현상을 과적합(overfitting) 문제라고 한다. 다시 말해, 과적합 문제는 모델이 훈련용 데이터에 치중해 학습했기 때문에 새로운 데이터가 들어왔을 때 성능을 발휘하지 못하는 현상이다. 과적합 문제는 딥러닝 연구의 대표적인 난점으로 손꼽힌다. 따라서 이를 해결하기 위한 다양한 방법이 제시되어 있고, 관련 연구 또한 현재 활발히 진행 중이다. 이번 연구의 한계점과 해결 방안 또한 과적합 문제를 해결하는 것과 연관되어 있다.

이번 연구의 가장 큰 한계점은 훈련용 데이터 수가 부족했다는 것이다. ‘연구 결과’에서 언급하였듯이, 이번 연구에서 사용한 훈련용 데이터 수는 은하의 클래스 하나당 2,000장의 사진이었다. 그러나 더 많은 양의 데이터를 모델에 입력했다면 모델은 각 은하의 더 다양한 특징들을 추출해 학습했을 것이며, 결국 과적합이 방지되어 검증용 데이터에 대한 모델의 성능은 이번 연구 결과보다 더 좋아질 것으로 전망된다. 데이터 수의 부족은 당연히 검증 데이터를 수집하는 과정에서 발생하였는데, 약 10,000장에 가까운 데이터를 가진 smooth, spiral 두 클래스의 은하와 달리, edge 클래스를 가진 은하의 수는 약 2,000장으로 그 데이터 수가 현저히 적었다.

이러한 클래스별 데이터 수의 불균형을 무시하고 모든 데이터를 한 모델에 학습시키면 각 클래스를 예측하는 정확도 사이에 큰 차이가 발생할 수 있다고 결론지었다. 따라서 클래스마다 2,000장의 데이터를 맞춰서 사용할 수밖에 없었고, 그 결과 훈련용 데이터 수가 부족해져 과적합 문제가 발생한 것으로 보인다.

이를 해결하기 위해 향후 연구에선 edge 형태의 은하 사진 데이터를 더 가져오거나, 원본 데이터를 조금씩 변형시켜 비슷한 데이터를 많이 만들어내는 데이터 증강(Data Augmentation) 기법을 사용할 필요가 있다. 이 외에도, 좀 더 다양하게 하이퍼 파라미터를 바꿔가며 실험하거나 과적합을 막기 위한 다양한 기법 등을 사용한다면 더 좋은 결과가 있으리라 기대된다.

참 고 문 헌

- [1] Hannu Karttunen 외 4인, 강혜성 외 6인 옮김, 『기본천문학』, Springer, 2006, 444-445
- [2] 프랑소와 솔레, 박해선 옮김, 『케라스 창시자에게 배우는 딥러닝』, 길벗, 2018, 33-37
- [3] 사이토 고기, 개얏맵시 옮김, 『밑바닥부터 시작하는 딥러닝』, 한빛미디어, 2017, 268-269
- [4] Jorge de la Calleja and Olac Fuentes, “Machine learning and image analysis for morphological galaxy classification”, *Monthly Notices of the Royal Astronomical Society*, 2004
- [5] Maribel Marin, L. Enrique Sucar, Jesus A. Gonzalez and Raquel Diaz, “A Hierarchical Model for Morphological Galaxy Classification”, *Florida Artificial Intelligence Research Society Conference*, 2013

- [6] Nour Eldeen M. Khalifa, Mohamed Hamed N. Taha, Aboul Ella Hassanien and I. M. Selim, "Deep Galaxy: Classification of Galaxies based on Deep Convolutional Neural Networks", *arXiv:1709.02245*, 2017
- [7] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition", *ICLR*, 2015
- [8] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. "Going deeper with convolutions", *CVPR*, 2015.
- [9] K. He, X. Zhang, S. Ren, J. Sun, J.: "Deep residual learning for image recognition", *CVPR*, 2016