

전이학습에 방법에 따른 컨벌루션 신경망의 영상 분류 성능 비교

박성욱[†], 김도연^{**}

Comparison of Image Classification Performance in Convolutional Neural Network according to Transfer Learning

Sung-Wook Park[†], Do-Yeon Kim^{**}

ABSTRACT

Core algorithm of deep learning Convolutional Neural Network(CNN) shows better performance than other machine learning algorithms. However, if there is not sufficient data, CNN can not achieve satisfactory performance even if the classifier is excellent. In this situation, it has been proven that the use of transfer learning can have a great effect. In this paper, we apply two transition learning methods(freezing, retraining) to three CNN models(ResNet-50, Inception-V3, DenseNet-121) and compare and analyze how the classification performance of CNN changes according to the methods. As a result of statistical significance test using various evaluation indicators, ResNet-50, Inception-V3, and DenseNet-121 differed by 1.18 times, 1.09 times, and 1.17 times, respectively. Based on this, we concluded that the retraining method may be more effective than the freezing method in case of transition learning in image classification problem.

Key words: Deep Learning, Computer Vision, Convolutional Neural Network, Transfer Learning

1. 서 론

기계 학습(Machine Learning)의 한 분야인 딥러닝(Deep Learning)[1]은 그 계산이 방대하고 목표 값으로 수렴하는 속도가 너무 느려서 실질적으로 사용하기 힘들다는 평가를 받았다. 그러나 2010년 이후 컴퓨터 성능이 급속도로 발달하고, 여러 이론과 방법론적인 문제가 해결되면서 그 응용 분야를 넓혔다.

최근 위 분야에서 컨벌루션 신경망(Convolutional Neural Network, CNN)[2]은 다른 기계 학습 알고리즘보다 우수한 성능을 보여준다[3]. 그러나 데이터가

충분히 확보되지 못한 상황이라면, CNN 역시 분류기(Classifier)가 우수하더라도 만족스러운 성능을 얻기 힘들다. 이와 같은 상황에서, 전이학습(Transfer Learning)[4]을 사용하면 큰 효과를 거둘 수 있다는 것이 입증되었다[5].

영상 분류(Image Classification)에서 전이학습은 사전 학습된 모델(Pre-trained Model)의 가중치(Weight)를 이용한다. 사전 학습된 모델은 입력 영상을 1,000개 클래스로 분류하는 대회인 ILSVRC를 대상으로 만들기 때문에 출력층의 유닛(Unit)이 1,000개이다. 사전 학습된 모델의 가중치는 ILSVRC에서

* Corresponding Author: Do-Yeon Kim, Address: (57922) Jungang-ro 255, Suncheon, Jeonnam, Korea, TEL: +82-61-750-3628, FAX: +82-61-750-3620, E-mail: dykim@sunchon.ac.kr

Receipt date: Aug. 28, 2018, Revision date: Oct. 23, 2018
Approval date: Nov. 7, 2018

[†] Dept. of Computer Eng., Suncheon National University (E-mail: park7231654@naver.com)

^{**} Dept. of Computer Eng., Suncheon National University
* This paper was supported by Suncheon National University Research Fund in 2018(Grant number: 2018-0179).

사용한 이미지넷(ImageNet)의 데이터베이스로 학습되어 만들어졌지만, 이미지넷 이외의 데이터 세트에도 적용이 가능하다.

가장 널리 알려진 방법은 사전 학습된 모델의 가중치를 신경망 가중치의 초깃값으로 동결(Freezing)하고 새로운 훈련 데이터 세트를 사용하여 학습하는 동결 방법이다. 사전 학습된 모델의 출력층은 유닛이 1,000개이므로 새로운 훈련 데이터 세트에 맞는 출력층으로 대체한다. 또 입력 데이터의 크기도 입력층의 유닛 수에 맞게 영상 크기를 조정해야 된다. 이와 같은 방법으로 신경망을 수정하면 작은 에폭(Epoch) 수로도 높은 성능을 얻을 수 있다. 하지만 동결 방법을 사용해 출력되는 분류 정확도가 정말 최적인지, 아직 이렇다 할 연구사례가 부족한 실정이다.

본 연구에서는 사전 학습된 모델의 가중치를 동결하지 않고 재훈련(Retraining)시키는 방법을 추가하여 방법에 따라 CNN의 분류 성능이 어떻게 변화하는지 비교 및 분석하였다. 학습에 사용 될 CNN 모델은 He et al.[6]의 ResNet-50, Szegedy et al.[7]의 Inception-V3, Huang et al.[8]의 DenseNet-121이고 각 네트워크 출력층의 클래스 수를 변형시켜 사용했다. 구동을 위한 딥러닝 라이브러리로 텐서플로(Tensorflow)[9], 상위 틀로 케라스(Keras)[10]를 사용했다. 케라스는 텐서플로 외에도 다양한 딥러닝 라이브러리를 호출하여 그에 맞는 알고리즘을 수행한다. 텐서플로는 기계 학습과 딥러닝을 위한 오픈소스 라이브러리다. 실험 결과, 학습 시간을 제외한 논문의 모든 평가항목에서 재훈련 방법이 동결 방법보다 높은 성능을 보였다.

논문의 2장에서는 연구 된 전이학습 방법들에 대한 설명을 기술하였고 3장에서는 학습 데이터 세트 및 실험 환경 구성에 대한 설명을 기술하였다. 4장에서는 실험 결과에 대한 설명을 기술하였고 마지막 5장에서는 결론 및 추후 연구과제에 대하여 기술하였다.

2. 전이학습

큰 규모의 깊은 컨볼루션 신경망(Deep Convolutional Neural Network, DCNN)[11]을 처음부터 훈련시키는 데엔 많은 비용과 시간이 소요된다. 따라서 해결하려는 것과 비슷한 유형의 문제를 처리한 신경망이 있는지 사전에 탐색해보고 그 신경망의 일부를

재사용하는 것이 좋다. 이 방법을 전이학습이라고 한다. 전이학습은 훈련 속도를 크게 높여줄 뿐만 아니라 필요한 훈련 데이터도 훨씬 적게 들며 입력이 비슷한 저수준 특징을 가질 때 잘 작동한다.

2.1 가중치 동결

동결 방식[4]은 가장 널리 쓰이는 방법으로 출력층 이전까지의 가중치를 고정시킨 채 재사용하는 경우다.

기존 사전 학습된 모델의 특징 추출(Feature Extraction)을 담당하는 컨볼루션 층은 영상 내 저수준 특징을 감지하도록 학습 되어서 다른 영상 분류 작업에 유용하다. 그래서 해당 층의 가중치들을 초깃값으로 고정시킨 후 재사용 하곤 하는데 이 방법이 동결 방식이다.

동결 방식을 사용하면 학습 하려는 대상이 바뀌지 않기 때문에 출력층의 가중치를 훈련시키기 쉽다. 이때 원본 모델의 출력층은 새로운 작업과 다를 수 있고 출력 유닛 수도 같지 않아 보통 삭제 후 추가된다. 텐서플로에서 동결 방식은 해당 층의 변수를 제외하고 훈련시킬 출력층의 변수 목록을 옵티마이저(Optimizer)에 전달하여 구현할 수 있다.

2.2 가중치 재훈련

재훈련 방식[12]은 신경망 모든 층의 가중치를 다시 훈련시키는 경우다.

재훈련 방식은 사전 학습된 모델의 가중치를 초깃값으로 설정하고 처음부터 다시 훈련시키는 방법이다. 재훈련 방식은 학습률을 낮게 설정하는 것이 중

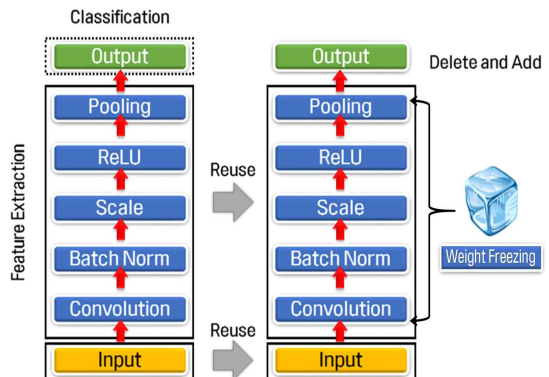


Fig. 1. Weight freezing method of transfer learning.

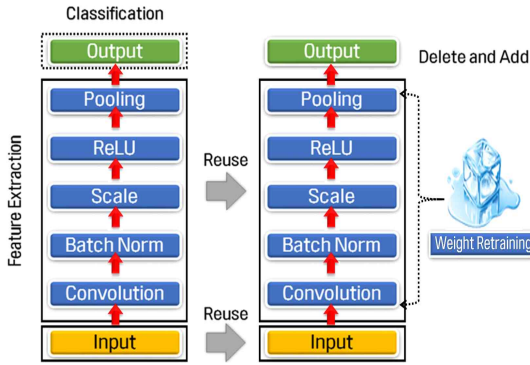


Fig. 2. Weight retraining method of transfer learning.

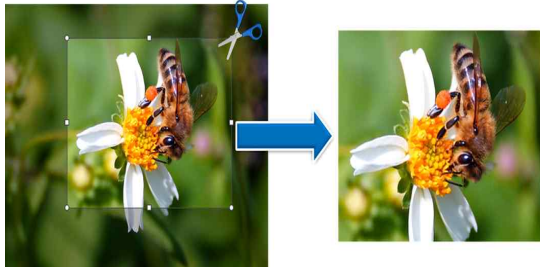


Fig. 3. Example of a cropped insect image.

요하다. 학습률이 높으면 모델이 가지고 있던 원래 가중치가 많이 훼손되어 좋은 성능을 기대하기 어렵기 때문이다. 낮은 학습률과 학습 하려는 대상이 바뀌었기 때문에 재훈련 방식은 동결 방식보다 출력층의 파라미터를 훈련시키는데 시간이 많이 소요된다는 단점이 있다. 텐서플로에서 재훈련 방식은 모든 층의 변수 목록을 옵티마이저에 전달하여 구현할 수 있다.

3. 학습 데이터 세트 및 실험 환경 구성

3.1 학습 데이터 세트 구성

실험 대상은 주위에서 쉽게 볼 수 있는 30종의 산림곤충으로 선정했다. 이미지넷에서 5종, 스크린 스크래핑(Screen Scraping)으로 25종 클래스에 해당하

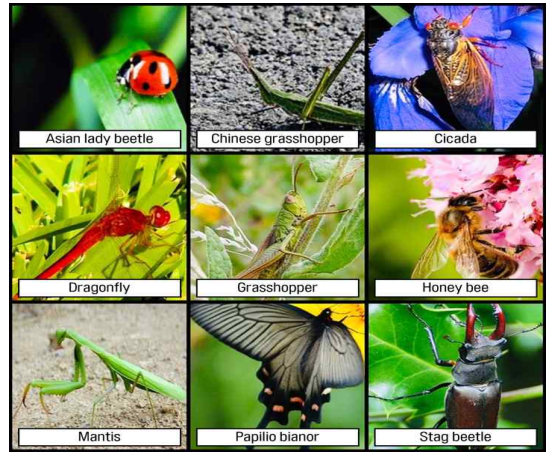


Fig. 4. Representative insect images used in the experiment.

는 영상들을 수집했고 별도의 선별작업을 거쳐 데이터세트로 구성했다. 스크린 스크래핑은 인터넷 스크린에 보이는 데이터 중에서 필요한 데이터만을 추출하도록 만들어진 프로그램이다. 곤충 데이터 세트의 영상은 Fig. 3과 같이 영상 내 곤충을 중심으로 크롭(Crop)했다.

훈련 시 영상 크기는 Inception-V3의 경우 299 X 299 화소(Pixel), ResNet-50 및 DenseNet-121의 경우 224 X 224 화소로 조절했다. 데이터가 불균형하면 모델 성능 우수하더라도 실용성과 객관성을 확보하는데 어려움이 있다. 따라서 본 연구에서는 각 곤충 클래스의 영상 개수를 훈련 세트(Training Set) 1,458장, 검증 세트(Validation Set) 162장, 테스트 세트(Test Set) 180장으로 동일하게 조정했다. 그러므로 훈련 세트는 43,740장, 검증 세트는 4,860장, 테스트 세트는 5,400장이 된다. 실험은 총합 54,000장의 영상을 가지고 진행했다. 데이터 세트 세부사항은 Table 1, 대표 곤충 영상은 Fig. 4와 같다.

3.2 실험 환경 구성

3가지 CNN 모델(ResNet-50, Inception-V3, Dense

Table 1. Data set information

Input Size		Training Set	Validation Set	Test Set
Inception-V3	299 × 299	43,740 (1,458 × 30)	4,860 (162 × 30)	5,400 (180 × 30)
ResNet-50	224 × 224			
DenseNet-121	224 × 224			

Net-121)과 두 가지 전이학습 방법(동결, 재훈련)을 조합하여 6가지의 실험 환경을 구성했다. 3가지 CNN 모델 모두 손실 함수(Loss Function)로 교차 엔트로피(Cross Entropy)[13]를 사용하였고 활성화 함수(Activation Function)는 정규화 선형 유닛(Rectified Linear Unit, ReLU)[14]을 사용했다. 사전 학습된 가중치는 각 네트워크의 마지막 컨벌루션 층까지만 사용하였고 FC 대신 전역 평균 풀링(Global Average Pooling, GAP)[15]층을 추가시켰다. GAP를 사용하면 오버피팅 및 많은 계산량 문제를 해결할 수 있다. 최종 출력 노드는 소프트맥스(Softmax) 활성화 함수를 사용했다. 소프트맥스 활성화 함수는 각 클래스에 대한 출력이 가장 높은 것을 정답 클래스로 분류한다. 매개변수 갱신 방법의 경우 Inception-V3는 실효값 전파(Root Mean Square Propagation, RMSProp)[16]를 사용하였고 초기 학습률(Base Learning Rate)=‘0.001’, 가중치 감쇠(Weight Decay)=‘1e-6’으로 설정했다. ResNet-50과 DenseNet-121은 네스테로프 모멘텀(Nesterov Momentum, NM)[17]을 사용하였고 초기 학습률=‘0.001’, 가중치 감쇠=‘1e-6’, 모멘텀=‘0.9’로 설정했다. 가중치 초기화의 경우 Inception-V3는 ‘Xavier 초기화[18]’, ResNet-50과 DenseNet-121은 ‘He 초기화[19]’ 방법을 사용했다. 미니 배치(Mini Batch) 크기는 ‘32’, 에폭은 ‘200’으로 설정했다. 그리고 검증 집합의 오류가 최저인 지점에서 멈추기 위해 조기 멈춤(Early Stopping)[20] 알고리즘을 사용했다. 사용한 알고리즘은 검증 집합의 오류가 최저인 지점을 기준으로 n 번 동안 성능 향상이 없으면 멈춘다. 본 연구에서는 n 값을 ‘20’으로 설정했다.

CPU는 인텔 코어i7-7세대를 사용했고 그래픽 카드는 NVIDIA TITAN Xp를 사용하였다. 램은 DDR4 32GB를 사용하였으며 SSD는 512GB를 사용하였다. 실험에 사용된 운영체제 및 소프트웨어 버전은 Table 2와 같다.

Table 2. Operating system and middleware information used in the experiment

	Version
Operating System	Ubuntu 16.04.4 LTS
CUDA	9.0.176
cuDNN	7.1
Tensorflow	1.8.0
Keras	2.2.2

4. 실험 결과

본 연구에서는 검증 및 테스트 데이터 세트를 사용하여 출력된 정확도(Accuracy), 손실률(Loss Rate), 에폭, 소요 시간(Lead Time)과 컴퓨터 비전 분야에서 주로 분류율에 대한 지표로 사용하는 정밀도(Precision)와 재현율(Recall) 그리고 f 점수가 학습 방법의 평가 지표로 사용되었다.

정밀도는 양성으로 예측된 것 중 얼마나 많은 샘플이 진짜 양성인지 측정하고 재현율은 전체 양성 샘플 중에서 얼마나 많은 샘플이 양성 클래스로 분류되는지를 측정한다. 정밀도와 재현율은 우수한 측정 방법이지만 전체 그림을 보기에는 부족하다. 따라서 정밀도와 재현율의 조화 평균, 즉 이 두 가지 방법을 하나로 요약해주는 f 점수를 평가 지표에 추가했다. 본 연구에서는 진짜 양성(True Positive, TP), 진짜 음성(True Negative, TN), 거짓 양성(False Positive, FP), 거짓 음성(False Negative, FN)을 산정하고 식(1), 식(2), 식(3)에 대입하여 정밀도, 재현율, f 점수를 산출했다.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (1)$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (2)$$

$$F - Score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (3)$$

4.1 ResNet-50

Table 3의 재훈련 방법은 52 에폭에서 99.05%로 가장 높은 정확도를 보여주었다. 동결 방법은 94 에폭에서 83.96%로 재훈련 방법과 15.09%의 성능 차이를 보였다.

동결 방법은 사전 학습된 모델 컨벌루션 층의 가중치를 모두 고정시키기 때문에 새로 추가 된 층에서만 학습이 진행된다. 따라서 훈련 파라미터 수가 61,470개이다. 반면 재훈련 방법은 모델의 컨벌루션 층까지 학습하기 때문에 훈련 파라미터 수가 23,596,062개로 384배가량 많다.

Table 3에서 동결 방법은 학습 후 테스트까지 1시간 34분 28초가 소요됐으며 재훈련 방법은 2시간 19분 6초가 소요됐다. 동결 방법은 94 에폭에서 훈련이 종료됐음에도 불구하고 재훈련 방법보다 학습 후 테

Table 3. Comparison of the two transfer learning results in ResNet-50

Method	Test Accuracy[%]	Epoch	Trainable parameter	Lead Time[s]
Freezing	83.96	94	61,470	5668
Retraining	99.05	52	23,596,062	8340

Table 4. The precision, recall, and f-score of ResNet-50

Method	Precision[%]		Recall[%]		F-Score[%]	
	Min	Avg	Min	Avg	Min	Avg
Freezing	46	88	60	84	62	84
Retraining	93	99	92	99	95	99

스트까지 44분 38초 빨랐다.

Table 4는 테스트 결과로 출력된 각 클래스 수치의 가중치 평균이다. Table 4에서 정밀도 가중치 평균은 동결 방법이 88%, 재훈련 방법이 99%로 11%의 차이를 보였다. 재현율과 f점수는 동결 방법이 각각 84%, 재훈련 방법이 99%로 15%의 차이를 보였다.

동결 방법에서 가장 낮은 정밀도를 기록한 클래스는 46%의 귀뚜라미 클래스였다. 귀뚜라미 클래스의 재현율은 93%로 180장 중 167장을 올바르게 분류했다. 가장 낮은 재현율을 기록한 클래스는 60%의 방아깨비였다. 방아깨비 클래스는 180장 중 108장을 올바르게 분류했다. 방아깨비 클래스는 전체 클래스에서 98%의 세 번째로 높은 정밀도를 기록했다.

재훈련 방법에서도 가장 낮은 정밀도를 기록한 클래스는 귀뚜라미 클래스였지만 93%로 동결 방법보다 47% 높았다. 귀뚜라미 클래스의 재현율은 96%로 180장 중 173장을 올바르게 분류했다. 가장 낮은 재현율을 기록한 클래스는 92%의 메뚜기였다. 메뚜기 클래스는 180장 중 166장을 올바르게 분류했다. 메뚜기 클래스는 전체 클래스에서 97%의 네 번째로 높은 정밀도를 기록했다. 이와 같이 정밀도와 재현율이 높게 유지되면 우수한 모델이라고 할 수 있다.

Fig. 5는 ResNet-50 모델의 훈련 결과를 보여주는 그래프이다. 그래프의 세로축은 손실률과 정확도를 나타내고 가로축은 에폭을 나타낸다. Fig. 5에서는 100 에폭까지의 검증 정확도와 손실률이 어떻게 변화하는지 관찰하고자 조기 멈춤 알고리즘을 사용하지 않았다.

1 에폭때 동결 방법의 정확도는 80.23%, 재훈련 방법의 정확도는 95.75%로 15.52%의 성능 차이를 보였다. 100 에폭에서 동결 방법의 정확도는 82.88%로

2.65% 상승세를 보였고 재훈련 방법의 정확도는 99.39%로 3.64%의 상승세를 보였다. 재훈련 방법은 동결 방법보다 정확도 상승폭이 0.99%더 컸고 최종 정확도도 더 높았다.

손실률의 경우 1 에폭때 동결 방법은 0.6329%를 기록했고 재훈련 방법은 0.1548%를 기록했다. 100 에폭때 동결 방법은 0.6033%를 기록했고 재훈련 방법은 0.0182%를 기록했다. 동결 방법은 0.0296%, 재훈련 방법은 0.1366%의 성능이 개선됐다. Fig. 5에서 재훈련 방법은 동결 방법보다 정확도, 손실률 모두 더 높은 성능을 보여주었다.

4.2 Inception-V3

Table 5의 재훈련 방법은 69 에폭에서 96.27%로 가장 높은 정확도를 보여주었다. 동결 방법은 68 에폭에서 88.16%로 재훈련 방법과 8.11%의 성능 차이를 보였다.

Inception-V3 모델에서 동결 방법의 훈련 파라미

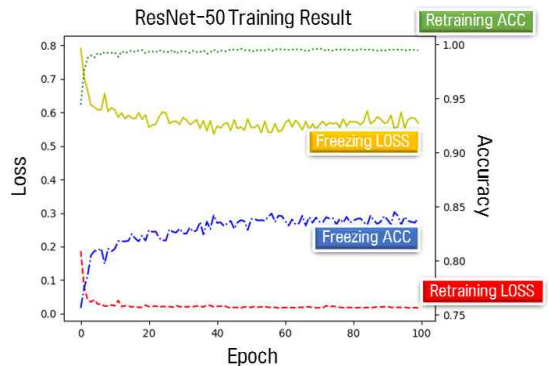


Fig. 5. Comparison of accuracy and loss rates up to 100 epoch without early stopping in ResNet-50

Table 5. Comparison of the two transfer learning results in Inception-V3

Method	Test Accuracy[%]	Epoch	Trainable parameter	Lead Time[s]
Freezing	88.16	68	61,470	5085
Retraining	96.27	69	21,829,822	14201

Table 6. The precision, recall, and f-score of Inception-V3

Method	Precision[%]		Recall[%]		F-Score[%]	
	Min	Avg	Min	Avg	Min	Avg
Freezing	62	90	56	88	71	88
Retraining	82	96	81	96	84	96

터 수는 61,470개, 재훈련 방법의 훈련 파라미터 수는 21,829,822개로 355배가량 차이가 있다. Table 5에서 동결 방법은 학습 후 테스트까지 1시간 24분 45초가 소요됐으며 재훈련 방법은 3시간 56분 41초가 소요됐다. 1 에폭 수행 시간을 환산해보면 동결 방법이 1분 15초, 재훈련 방법이 3분 26초로 훈련 속도 면에서 동결 방법이 2분 11초 빨랐다.

Table 6에서 정밀도 가중치 평균은 동결 방법이 90%, 재훈련 방법이 96%로 6%의 차이를 보였다. 재현율과 f 점수는 동결 방법이 각각 88%, 재훈련 방법이 96%로 8%의 차이를 보였다.

동결 방법에서 가장 낮은 정밀도를 기록한 클래스는 62%의 하늘소 클래스였다. 하늘소 클래스의 재현율은 98%로 180장 중 176장을 올바르게 분류했다. 가장 낮은 재현율을 기록한 클래스는 56%의 산누에나방 이었다. 산누에나방 클래스는 180장 중 100장을 올바르게 분류했다. 산누에나방 클래스는 전체 클래스에서 98%의 세 번째로 높은 정밀도를 기록했다.

재훈련 방법에서 가장 낮은 정밀도를 기록한 클래스는 82%의 귀뚜라미와 메뚜기 클래스였다. 재현율은 각각 87, 88%로 동결 방법보다 9~10% 낮았다. 그러나 정밀도와 재현율의 차이는 5~6%로 36%의 차이를 기록한 동결 방법보다 30~31% 낮다. 이상적인 그림은 정밀도가 가장 조금 감소하면서 재현율이 가장 많이 증가하는 방향이므로 동결 방법의 결과가 더 낫다고 할 수 없다. 귀뚜라미와 메뚜기 클래스는 180장 중 각각 157, 159장을 올바르게 분류했다. 가장 낮은 재현율을 기록한 클래스는 81%의 사마귀였다. 사마귀 클래스는 180장 중 145장을 올바르게 분류했다. 사마귀 클래스는 전체 클래스에서 97%의 네 번째로 높은 정밀도를 기록했다.

Fig. 6은 Inception-V3 모델의 훈련 결과를 보여주는 그래프이다. 1 에폭때 동결 방법의 정확도는 78.80%, 재훈련 방법의 정확도는 36.11%로 42.69%의 성능 차이를 보였다. 재훈련 방법은 7 에폭에 도달해서야 동결 방법보다 0.71% 높은 87.60%의 정확도를 달성했다. 100 에폭에서 동결 방법의 정확도는 89.74%로 10.94% 상승세를 보였고 재훈련 방법의 정확도는 96.43%로 60.32%의 상승세를 보였다. 재훈련 방법은 동결 방법보다 정확도 상승폭이 49.38%더 컸고 최종 정확도도 더 높았지만 그래프가 심하게 굽어진 움직임 보여준다. 동결 방법은 재훈련 방법과 비교하면 지그재그 정도가 덜한 것을 알 수 있다.

손실률의 경우 100 에폭때 동결 방법은 0.6329%를 기록했고 재훈련 방법은 0.1548%를 기록했다. 1 에폭 손실률은 동결 방법이 0.7321로 재훈련 방법보다 5.2354% 더 낮았지만, 최종 손실률은 재훈련 방법이 0.2223%로 동결 방법보다 0.3438% 더 낮았다. In-

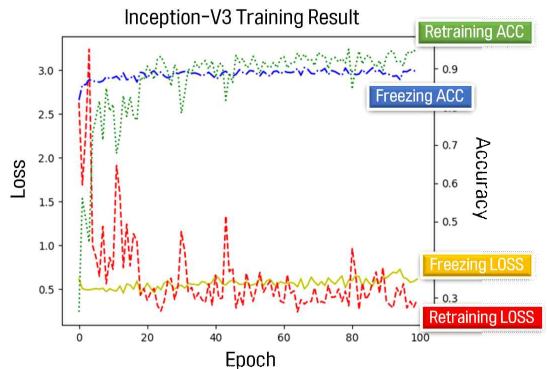


Fig. 6. Comparison of Accuracy and loss rates up to 100 epoch without early stopping in Inception-V3.

Table 7. Comparison of the two transfer learning results in DenseNet-121

Method	Test Accuracy[%]	Epoch	Trainable parameter	Lead Time[s]
Freezing	84.38	60	30,750	3715
Retraining	99.16	44	6,984,606	7463

Table 8. The precision, recall, and f-score of DenseNet-121

Method	Precision[%]		Recall[%]		F-Score[%]	
	Min	Avg	Min	Avg	Min	Avg
Freezing	49	89	38	84	55	85
Retraining	96	99	94	99	95	99

ception-V3 모델에서 재훈련 방법은 동결 방법보다 최종 정확도와 손실률 모두 더 높은 성능을 보여주었다.

4.3 DenseNet-121

Table 7의 재훈련 방법은 44 에폭에서 99.16%로 가장 높은 정확도를 보여주었다. 동결 방법은 60 에폭에서 84.38%로 재훈련 방법과 14.78%의 성능 차이를 보였다.

DenseNet-121 모델에서 동결 방법의 훈련 파라미터 수는 30,750개, 재훈련 방법의 훈련 파라미터 수는 6,984,606개로 227배가량 차이가 있다. Table 7에서 동결 방법은 학습 후 테스트까지 1시간 1분 55초가 소요됐으며 재훈련 방법은 2시간 4분 23초가 소요됐다. 동결 방법은 재훈련 방법보다 16 에폭이후 종료됐음에도 불구하고 학습 후 테스트까지 1시간 2분 28초 빨랐다.

Table 8에서 정밀도 가중치 평균은 동결 방법이 89%, 재훈련 방법이 99%로 10%의 차이를 보였다. 재현율과 F점수는 동결 방법이 각각 84, 85%, 재훈련 방법이 99%로 14~15%의 차이를 보였다.

동결 방법에서 가장 낮은 정밀도를 기록한 클래스는 49%의 메뚜기 클래스였다. 메뚜기 클래스의 재현율은 83%로 180장 중 150장을 올바르게 분류했다. 가장 낮은 재현율을 기록한 클래스는 38%의 꿀벌이었다. 꿀벌 클래스는 180장 중 68장을 올바르게 분류했다. 꿀벌 클래스는 전체 클래스에서 100%의 가장 높은 정밀도를 기록했다. 이처럼 다른 클래스가 꿀벌 클래스로 잘못 분류한 것이 없을 때 정밀도는 100%가 된다.

재훈련 방법에서 가장 낮은 정밀도를 기록한 클레

스는 96%의 귀뚜라미 클래스였다. 재현율은 95%로 180장 중 171장을 올바르게 분류했다. 가장 낮은 재현율을 기록한 클래스는 94%의 메뚜기였다. 메뚜기 클래스는 180장 중 169장을 올바르게 분류했다. 메뚜기 클래스는 전체 클래스에서 97%의 네 번째로 높은 정밀도를 기록했다.

Fig. 7은 DenseNet-121 모델의 훈련 결과를 보여주는 그래프이다. 1 에폭때 동결 방법의 정확도는 73.00%, 재훈련 방법의 정확도는 97.45%로 24.45%의 성능 차이를 보였다. 100 에폭에서 동결 방법의 정확도는 84.31%로 11.31% 상승세를 보였고 재훈련 방법의 정확도는 99.51%로 2.06%의 상승세를 보였다. 해당 모델은 동결 방법의 정확도 상승폭이 9.25% 더 컸다.

손실률의 경우 100 에폭때 동결 방법은 0.5116%를 기록했고 재훈련 방법은 0.0168%를 기록했다. DenseNet-121 모델에서 재훈련 방법은 동결 방법보다 최종 정확도와 손실률 모두 더 높은 성능을 보여주었다.

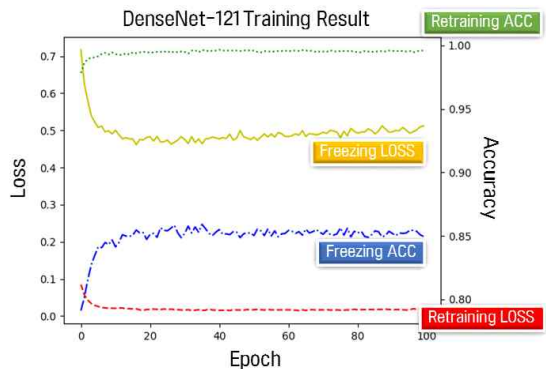


Fig. 7. Comparison of Accuracy and loss rates up to 100 epoch without early stopping in DenseNet-121.

5. 결 론

본 연구에서는 30종의 산림 곤충 데이터세트를, 3가지 CNN 모델(ResNet-50, Inception-V3, DenseNet-121)과 2가지 전이학습(동결, 재훈련) 방법을 통해 학습시켰다. 동결 방법은 훈련 속도가 재훈련 방법보다 1 에폭 기준 1분 40초~2분 11초가량 빨랐지만 나머지 평가 지표에서는 좋은 성능을 보여주지 못했다. 정확도, 손실률, 정밀도, 재현율, 그리고 f 점수에서는 재훈련 방법의 성능이 더 우수하게 나타났다. 실험 결과의 Fig. 5~Fig. 7은 재훈련방법이 제대로 동작하고 또 효과적임을 보여준다. 동결 방법의 경우 시각적 특징이 뚜렷하지 않거나 서로 유사한 다른 곤충 클래스에 의해 비교적 정확한 분류가 쉽지 않음을 알 수 있다. 다양한 평가 지표를 이용하여 통계적 유의성 검증을 시행한 결과, 전반적으로 재훈련 방법의 예측 성능이 동결 방법보다 우수했다. 정확도 측면만 놓고 보자면, 재훈련 방법의 성능이 동결 방법보다 ResNet에서 1.18배, Inception에서 1.09배, DenseNet에서 1.17배 더 높았다.

하지만 이러한 결론을 근거로 모든 영상 분류 문제에서 재훈련 방법이 동결 방법보다 더 효율적이라고 일반화할 수는 없다. 벤치마크 시험(Benchmark Test)을 통해 검증된 성능이 현실의 모든 문제를 대변할 수 없기 때문이다[21]. 이와 같은 이유로, 도전 과제의 최적인 방법은 실험을 통해 확인하는 수밖에 없다고 사료된다. 그러나 본 실험의 학습 데이터 세트와 화소값 분포가 비슷하거나 높은 벤치마킹 점수의 CNN을 사용하는 연구에서, 위 결과는 시간과 비용을 절약하는데 크고 작은 도움이 될 것으로 기대된다.

끝으로, 본 실험의 결과 성능은 하이퍼파라미터(Hyper-parameter) 값을 보다 정밀하게 조정하면 더 끌어올릴 수 있을 것이라 판단된다. 추후 연구과제는 실험에 사용된 방법들의 상하관계를 명확하게 밝힐 수 있는 연구와 객체 검출(Object Detection)[22], 객체 분할(Object Segmentation)[23]과 같은 분류보다 어려운 문제에서 위 전이학습 방법들이 어떤 영향을 미치는지 평가할 수 있는 연구가 필요할 것으로 사료된다.

REFERENCE

- [1] L. Yann, B. Yoshua, and H. Geoffrey, "Deep Learning," *Nature*, Vol. 521, No. 7553, pp. 436-444, 2015.
- [2] Y. Lecun, L. Bottou, and Y. Bengio, "Gradient-based Learning Applied to Document Recognition," *Proceeding of The IEEE*, Vol. 86, No. 11, pp. 2278-2324, 1998.
- [3] K. Alex, S. Ilya, and H. Geoffrey, "ImageNet Classification with Deep Convolutional Neural Networks," *Proceeding of Advances in Neural Information Processing System*, pp. 1097-1105, 2012.
- [4] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, et al., "DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition," *arXiv*, arXiv: 1310.1531, 2013.
- [5] A.S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN Features Off-the-shelf: An Astounding Baseline for Recognition," *Proceeding of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 512-519, 2014.
- [6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *arXiv*, arXiv:1512.03385, 2015.
- [7] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," *arXiv*, arXiv:1512.00567, 2015.
- [8] G. Huang, Z. Liu, L. v. d. Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," *Proceeding of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2261-2269, 2017.
- [9] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro et al., "TensorFlow: Large-scale Machine Learning on Heterogeneous Systems," *arXiv*, arXiv:1603.04467, 2015.
- [10] Keras, <https://github.com/fchollet/keras> (accessed Jun., 4, 2018).

[11] Y. Jeong, I. Ansari, J. Shim, and J. Lee, "A Car Plate Area Detection System Using Deep Convolution Neural Network," *Journal of Korea Multimedia Society*, Vol. 20, No. 8, pp. 1166-1174, 2017.

[12] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How Transferable Are Features in Deep Neural Networks?," *arXiv*, arXiv:1411.1792, 2014.

[13] K. Janocha and W.M. Czarnecki, "On Loss Functions for Deep Neural Networks in Classification," *arXiv*, arXiv:1702.05659, 2017.

[14] V. Nair and G. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines," *Proceeding of the 27th International Conference on Machine Learning*, pp. 807-814, 2010.

[15] M. Lin, Q. Chen, and S. Yan, "Network In Network," *arXiv*, arXiv:1312.4400v3, 2014.

[16] T. Tieleman and G. Hinton, *Rmsprop: Divide the Gradient by a Running Average of Its Recent Magnitude*, Coursera: Neural Networks for Machine Learning Technical Report, 2012.

[17] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the Importance of Initialization and Momentum in Deep Learning," *Proceeding of the 30th International Conference on Machine Learning*, Vol. 28, pp. 1139-1147, 2013.

[18] X. Glorot and Y. Bengio, "Understanding the Difficulty of Training Deep Feedforward Neural Networks," *Proceeding of the International Conference on Artificial Intelligence and Statistics*, pp. 249-256, 2010.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Delving Deep into Rectifiers: Surpassing Human-level Performance on Imagenet Classification," *Proceeding of International Conference on Computer Vision*, pp. 1026-1034, 2015.

[20] L. Prechelt, "Early Stopping-but When?," *Neural Networks: Tricks of the Trade*, pp. 53-67, 2012.

[21] Y. Bengio, "Practical Recommendations for Gradient-based Training of Deep Architectures," *arXiv*, arXiv:1206.5533, 2012.

[22] D. Jung, J. Son, and S. Kim, "Shot Category Detection Based on Object Detection Using Convolutional Neural Networks," *Proceeding of International Conference on Advanced Communication Technology*, pp. 36-39, 2018.

[23] S. Park, U. Park, and D. Kim, "Depth Image-based Object Segmentation Scheme for Improving Human Action Recognition," *Proceeding of International Conference on Electronics, Information, and Communication*, pp. 1-3, 2018.



박 성 옥

2018년 순천대학교 컴퓨터공학과 (공학사)
 2018년~현재 순천대학교 컴퓨터공학과 석사과정
 관심분야: 컴퓨터비전, 기계학습



김 도 연

1986년 충남대학교 계산통계학과 졸업(이학사)
 2000년 충남대학교 대학원 정보통신공학과 졸업(공학석사)
 2003년 충남대학교 대학원 컴퓨터공학과 졸업(공학박사)
 1986년~1996년 한국원자력연구원 선임연구원
 1997년~2008년 한국전력기술(주) 책임연구원
 2008년~현재 순천대학교 컴퓨터공학과 교수
 관심분야: 컴퓨터비전, 컴퓨터보안, 기계학습