

다시점 영상 집합을 활용한 선체 블록 분류를 위한 CNN 모델 성능 비교 연구

전해명·노재규[†]
군산대학교 조선해양공학과

Comparison Study of the Performance of CNN Models with Multi-view Image Set on the Classification of Ship Hull Blocks

Haemyung Chon·Jackyou Noh[†]
Department of Naval Architecture and Ocean Engineering, Kunsan National University

This is an Open-Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

It is important to identify the location of ship hull blocks with exact block identification number when scheduling the shipbuilding process. The wrong information on the location and identification number of some hull block can cause low productivity by spending time to find where the exact hull block is. In order to solve this problem, it is necessary to equip the system to track the location of the blocks and to identify the identification numbers of the blocks automatically. There were a lot of researches of location tracking system for the hull blocks on the stockyard. However there has been no research to identify the hull blocks on the stockyard. This study compares the performance of 5 Convolutional Neural Network (CNN) models with multi-view image set on the classification of the hull blocks to identify the blocks on the stockyard. The CNN models are open algorithms of ImageNet Large-Scale Visual Recognition Competition (ILSVRC). Four scaled hull block models are used to acquire the images of ship hull blocks. Learning and transfer learning of the CNN models with original training data and augmented data of the original training data were done. 20 tests and predictions in consideration of five CNN models and four cases of training conditions are performed. In order to compare the classification performance of the CNN models, accuracy and average F1-Score from confusion matrix are adopted as the performance measures. As a result of the comparison, Resnet-152v2 model shows the highest accuracy and average F1-Score with full block prediction image set and with cropped block prediction image set.

Keywords : Multi-view image set(다시점 영상 집합), Convolutional Neural Network(CNN, 합성곱신경망), Ship hull block(선체 블록), Classification(분류), Data augmentation(데이터 확장), Transfer learning(전이학습)

1. 서론

대형조선소에서 건조하는 대형선박은 크기가 매우 크기 때문에 전체 선박을 수습에서 수백 개의 단위 블록으로 분할하여 생산 설계한다. 분할 설계된 선박 블록은 구성하는 부재들을 가공 후 소조립, 중조립, 대조립 및 도장 공정을 거쳐 대형블록으로 제작되며 도크에서 대형블록들을 탑재 후 진수한다 (Kim et al., 2013). 이러한 일련의 선박 건조 공정을 차질 없이 진행하기 위해 일정 계획을 세우고 실행하는데 공정의 진행 중 작업이 완성

되었거나 다음 작업이 할당되지 않은 블록들은 옥외 적치장에 임시 배치된다. 배치된 블록은 여러 요인에 의한 일정 변경으로 애초 계획했던 작업장과 같은 작업이 가능한 대안 선호 작업장으로 이동하여 작업이 진행되기도 한다. 이런 과정에서 하루에 적게는 수십 번부터 많게는 수백 번 블록들이 적치장에 반입, 반출하게 된다 (Nam et al., 2017). 이때 이동된 블록이 위치한 작업장 정보는 일정 계획을 위해 확보해야 한다. 대부분의 조선소에서는 PC에서 구동되는 시스템을 블록의 물류관리에 이용한다. 현장작업자가 하루의 작업내용을 수기로 기록하였다가 일과시간 후 물

류관리시스템에 입력하는 이중 작업에 의존하고 있다 (Cho et al., 2011). 그러나 블록 번호 및 작업장 위치와 같은 블록 정보를 입력하는 과정에서 작업자의 인지 오류나 오입력으로 인해 블록 번호 또는 작업장 위치 정보가 잘못 입력되는 경우가 발생할 수 있다. 이러한 경우에는 다음 작업을 위한 블록 위치를 파악하는데 문제가 발생할 수 있다 (Shin & Lee, 2006). 적치장에서 블록 정보 오류에 기인한 해당 블록을 찾기 위해 소모되는 시간이 수십 분에서 수 시간 정도 걸리기도 하여 생산성을 저해하는 요인이 되기도 한다. 이러한 블록 정보 오류를 방지하기 위해서는 블록 정보인 블록 번호와 위치 정보가 둘 다 정확하게 시스템에 입력되어야 한다.

조선소 적치장에 적치된 블록의 위치를 추적하는 연구는 꾸준히 수행되어왔다. Shin & Lee (2006)는 작업자가 블록 위치 이동 작업의 완료 후 항시 소지하는 PDA를 이용해 사용자가 위성항법 범시스템(Global Positioning System, GPS) 정보를 포함한 블록 정보를 수작업으로 입력해 서버에 실시간으로 전송 가능한 시스템을 제안하였다. Lee et al. (2008)은 위치 정도가 향상되고 일시적으로 전파가 단절되어도 위치 추적이 가능한 GPS/INS 시스템을 물류 운반 차량에 장착하여 휴대폰 무선통신망을 이용해 실시간으로 정보 전송이 가능한 시스템을 제안하였다. Kim et al. (2009)은 RFID 태그를 적치장에 설치하고 트랜스포터를 이용해 블록 운반 시 운반 블록의 적치 위치 정보를 해당 RFID 태그를 읽어 들어 시스템에 입력할 수 있는 운영관리시스템에 관한 연구를 진행하였다. Cho et al. (2011)은 작업 현장에서 현장 작업자의 이중 작업을 개선하기 위해 블록 위치 정보를 입력할 수 있는 안드로이드 OS 용 물류 관계 시스템의 개발을 제안하였다. Kang (2014)은 모바일기기를 이용하여 작업 현장에서 직접 블록의 위치에 대한 정보를 입력 후 무선통신망으로 전송하는 시스템을 개발하였다. 그리고 Mun (2019)은 위성항법시스템의 신호 음영 및 왜곡 문제 해결을 위해 범역항법위성시스템(Global Navigation Satellite System, GNSS)과 관성측정장치(Inertial Measurement Unit, IMU) 센서를 융합한 위치추적 장치를 개발하여 사용자가 모바일 기기와 컴퓨터로 입력할 수 있는 시스템을 개발하였다.

이와 같은 블록 정보 입력에 관한 선행연구들은 주로 수작업에 의한 블록 위치 정보 입력에 따른 오류를 극복하기 위한 실시간 자동 입력에 관한 연구가 주를 이루고 있음을 알 수 있다. 그런데 블록 번호가 인지 오류에 의하거나 오타로 인해 잘못 입력된 경우에도 적치장에서 찾고자 하는 블록을 입력된 적치 위치에서 찾지 못하게 되므로 결과적으로 적치 위치 오류와 같은 결과를 초래하게 된다. 그러나 블록 정보의 중요한 정보 중 하나인 블록 번호에 대한 오입력 방지 또는 자동 입력에 관한 연구는 거의 수행된 적이 없었다. 따라서 작업자의 블록 번호 수작업 입력에서의 오류를 줄일 수 있는 블록 자동 인식에 관한 연구가 필요하다.

이를 위해 위성항법시스템 또는 위치식별이 가능한 장치를 장착한 드론이나 지상 자율주행 로봇에 카메라를 부착하여 획득한 블록 형상 영상으로부터 적치된 블록의 형상을 인식하고 식별하

여 자동으로 블록 번호를 부여하고 적치 위치를 확인할 수 있는 시스템을 구현하는 것을 고려해 볼 수 있다. 이러한 시스템 구현에 필요한 자동 블록 인식에 관한 연구를 위해 영상을 통한 객체 분류에서 좋은 결과를 내는 것으로 알려진 딥러닝 기법의 하나인 CNN을 활용하는 방법도 하나의 대안이 될 수 있다.

딥러닝은 기계학습(machine learning)의 한 분야이다. 기계학습은 학습 방법에 따라서 지도학습(supervised learning)과 비지도 학습(unsupervised learning)으로 분류된다. 지도학습은 문제에 대한 답을 주고 학습한 후 새로운 문제에 대한 답을 구한다. 학습결과를 바탕으로 예측하고자 하는 대상의 연속성과 이산성에 따라 회귀(regression)와 분류(classification) 등으로 구분할 수 있다. 비지도 학습은 답을 주지 않고 문제만 학습시킨 후 유사한 성질을 가지는 그룹으로 나누는 학습 방법이다. CNN은 지도학습으로 분류된다.

CNN을 사용한 자동 블록 인식에 관한 연구를 수행하기 위해서는 다양한 CNN 모델 중에서 하나의 모델을 선택하여 적용하여야 할 것이다. 그런데 모든 영상 분류 문제에서 벤치마크 시험을 통해 검증된 성능이 현실의 모든 문제에 일반적으로 적용될 수 있다고 할 수는 없으며 최적인 방법은 실험을 통해 확인하는 수밖에 없다 (Park & Kim, 2018). 따라서 영상 집합을 이용하여 영상 인식 알고리즘의 성능을 평가하는 ILSVRC (Stanford University, 2018) 이미지 인식 경진대회에서 좋은 성능을 보여 검증받은 다양한 CNN 모델들을 대상으로 실험을 수행하여 선체 블록 인식에 가장 적합한 성능을 가지는 CNN 모델을 선정할 필요가 있다. 본 연구에서 성능 비교를 위해 사용한 CNN 모델은 VGGNet (Simonyan & Zisserman, 2014), GoogLeNet (Szegedy et al., 2014), ResNet (He et al., 2016), DenseNet (Huang et al., 2017) 그리고 NASNet (Zoph et al., 2018) 모델로 모두 5가지이다.

하지만 학습 및 식별을 위한 실제 선체 블록 영상 수집에 현실적 어려움이 있어 선체 블록의 특징을 그대로 가지고 있는 선체 블록 모형 제작을 통해 다시점 영상 집합을 수집하고 학습한 다음 CNN 모델들의 성능을 실험하였다. 다시점 영상 집합을 이용하는 학습은 3차원 형상을 다양한 방향에서 본 여러 장의 영상으로 투영시켜 획득한 영상 집합을 학습에 이용하는 방법이다 (Kim, 2018).

그런데 성능 비교를 위한 CNN 모델들이 개발 당시 학습에 사용한 영상 데이터의 크기는 본 연구에서 CNN 모델 학습에 사용하는 학습용 영상 데이터의 크기보다 아주 크다. 이렇게 학습용 데이터의 크기가 충분히 크지 않은 상황에서는 CNN 모델의 분류기(classifier)는 성능이 우수하지만 만족스러운 성능을 얻기 어려울 수도 있다 (Park & Kim, 2018). 본 연구에서는 이에 대한 성능 향상을 정량적으로 비교해 보기 위해 2가지 기법을 추가하였다. 하나는 원본 데이터를 변형시켜 학습용 데이터의 수를 늘리는 데이터 확장(data augmentation)이고 다른 하나는 이미 대단위 학습용 영상 데이터로 학습한 파라미터를 가져와 추가 학습에 도입하여 사용하는 전이학습(transfer learning)이다.

2장에서는 본 연구에서 비교한 CNN 모델에 대해 기술하고, 3장에서는 블록 인식 성능 향상을 위한 전이학습에 대한 설명을 기술하였다. 4장에서는 훈련을 위한 다시점 영상 집합과 실험환경 구성에 관하여 기술하며 5장에서는 실험에 따른 결과와 분석을 기술하고 6장에서는 결론과 향후 연구를 기술하였다.

2. CNN 모델

최근 CNN 기법은 이미지넷(ImageNet)과 같은 대규모 공용 영상을 저장소와 Graphics Processing Unit(GPU)을 이용한 고성능 연산을 통해 영상 및 동영상 인식에서 좋은 성과를 거두고 있다 (Simonyan & Zisseran, 2014). 손글씨와 같은 고차원의 패턴을 가진 데이터를 더 좋은 성능으로 분류하기 위해 기울기(gradient) 기반 학습 방식을 통해 만든 LeNet (LeCun et al., 1998)을 시작으로 영상 분야를 다루기에 최적화된 CNN 모델들은 다양한 영역에서 적용되고 있다.

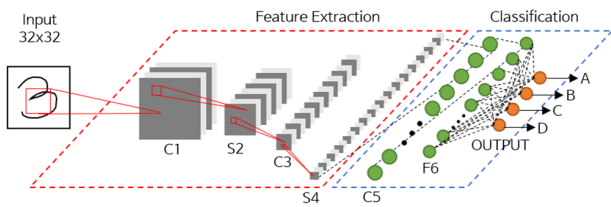


Fig. 1 Architecture of LeNet-5 (LeCun et al., 1998)

CNN 모델은 크게 입력층, 출력층과 두 층 사이의 많은 은닉층으로 구성되어 있으며 일반적으로 Fig. 1과 같이 특징추출계층(feature extraction)과 분류계층(classification) 구조를 가진다.

특징추출계층의 합성곱레이어(convolutional layer)에서는 합성곱필터(convolutional filter)가 영상 위를 돌아다니면서 합성곱 연산을 적용한 결과를 다음 레이어로 출력한다. 합성곱레이어는 합성곱필터의 개수, 사이즈와 스트라이드(Stride) 그리고 패딩(Padding)을 매개변수(parameter)로 갖는다. 합성곱 연산은 Fig. 2와 같이 입력데이터에서 합성곱필터가 스트라이드만큼 이동하며 대응되는 원소끼리 곱한 후 그 총합을 구한다. 편향(bias)은 필터를 적용한 후 값에 더해진다. 패딩은 합성곱 연산을 수행하기 전 영상 경계를 특정값으로 채워 확장한 크기이다.

합성곱레이어에서 합성곱필터를 통해 연산 되어 추출한 특징 맵(feature map)에는 활성화 함수(activation function)가 적용된다. 활성화 함수는 추출된 특징맵에서 정량적인 값으로 나오는 특징을 출력값으로 변환하여 활성화 여부를 판단한다. 출력값의 범위는 활성화 함수의 종류에 따라 다르다. 예로든 LeNet(Fig. 1)의 경우 합성곱 연산을 수행하는 C레이어에서 활성화 함수로 식 (1)의 시그모이드함수(Sigmoid function)를 사용하였다. 입력값이 작을 때의 출력값은 0에 가깝도록, 입력값이 커지면 출력값이 1에 가까워지도록 하는 역할을 한다 (Francois, 2018).

$$h(x) = \frac{1}{1 + \exp(-x)} \quad (1)$$

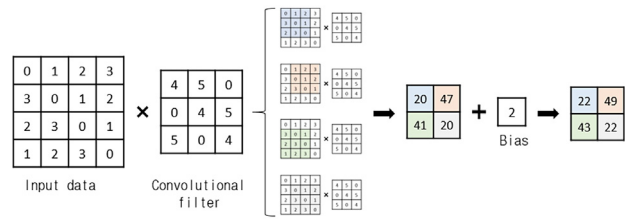


Fig. 2 Schematic diagrams of convolution operation (Stride = 1)

활성화 함수가 적용된 출력값은 풀링레이어(Pooling layer)로 전달된다. 풀링(Pooling)은 Fig. 3과 같이 차원(D)을 가지는 영상의 세로(H), 가로(W) 방향의 크기를 줄이는 연산이다. 예로 든 LeNet의 경우 S레이어에서 평균풀링을 통해 대상 출력값의 크기를 줄인다.

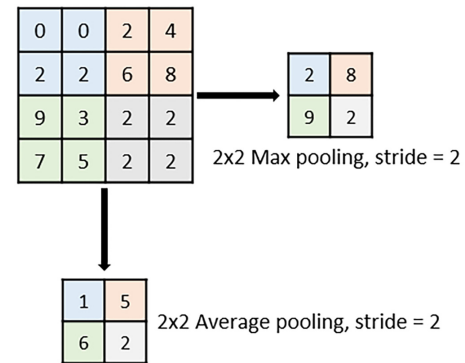


Fig. 3. Average pooling and max pooling

분류계층의 완전연결레이어(fully connected layer)는 합성곱레이어와 연결되며 결과를 출력한다. LeNet에서는 F6레이어에 해당하며 입력벡터(input vector)와 가중치벡터(weight vector)의 내적 후 편향을 더하여 연산한다. 최종적으로 출력(output)레이어에서 10개의 클래스(class)로 구분한다.

CNN 기법은 객체 검출과 인식 분야를 발전시키는 중이다 (Krizhevsky et al., 2012). 각 CNN 모델들은 상기 설명된 기본적인 형태는 같으나 구조의 깊이와 구성에 차이가 있으며 이전 CNN 모델에서 발생했던 문제를 해결하기도 한다. 다음은 본 연구에서 사용한 CNN들의 특징과 장단점에 대하여 기술하였으며 Table 1에 간략히 정리하였다.

2.1 VGGNet

VGGNet은 2014 ILSVRC에서 2위를 차지한 모델이다. 구조적으로 단순해서 변형이 쉬우므로 많이 사용된다 (Simonyan & Zisseran, 2014). VGGNet의 특징은 이전 모델들과 다르게 작은 크기의 3x3 합성곱필터를 사용했으며 16개, 19개의 레이어로 구성해 이전의 모델들보다 깊이(depth)가 깊다는 것이다. VGGNet은 파라미터의 수가 너무 많은 문제점을 가지고 있다. 본 연구에서 사용한 CNN 모델 VGG-19구조는 114만개의 파라미터를 가지고 있다.

2.2 GoogLeNet

GoogLeNet은 2014 ILSVRC에서 1위를 차지한 모델이다. 대용량의 데이터를 학습할 때 망이 깊고 층(layer)이 넓을수록 성능이 좋지만 많은 연산량과 파라미터 수에 따른 문제들이 발생하였다. 이를 해결하고자 인셉션모듈(inception module)을 사용하여 1×1 크기의 합성곱필터로 영상의 차원을 줄여 연산량과 파라미터수를 줄였다 (Szegedy et al., 2014). 하지만 구조가 복잡하기 때문에 구조를 변형시켜 사용하기 어려운 문제점이 있다. 본 연구에서는 GoogLeNet모델의 Inception-v3구조를 사용하였다 (Szegedy et al., 2016).

2.3 ResNet

ResNet은 2015 ILSVRC에서 1위를 차지한 모델이다. 모델의 깊이가 깊어지면 기울기 소실, 기울기 폭발(exploding gradient) 등의 문제가 발생한다 (Byeon & Kwak, 2018). ResNet은 깊이를 깊게 하면서도 학습 성과를 좋게 하도록 잔차학습방법(residual learning)을 사용하여 이 문제를 해결한다. ResNet은 잔차학습방법을 사용하기 위하여 빌딩블록을 여러 개 쌓는 구조를 가지고 있다. 이 구조로 인해 VGGNet보다 8배 깊으면서도 파라미터도 증가하지 않고 특징맵들이 더해져 연결되기 때문에 학습이 간단해진다. 따라서 깊은 네트워크를 최적화하며 높은 정확도도 얻을 수 있다 (He et al., 2016). 하지만 결국 모델의 깊이가 적절한 수준을 벗어나 더욱 깊어질 경우 기울기 소실문제가 발생한다는 문제점이 있다. 본 연구에서는 ResNet모델의 Resnet-152v2구조를 사용하였다.

2.4 DenseNet

DenseNet 이전의 CNN 모델들은 합성곱필터의 연산 결과인 특징맵이 다음 특징맵과 연결되는 기본적인 방식이거나, ResNet의 잔차 방식이다. Fig. 4와 같은 DenseNet은 특징맵이 연속하여 병합(concatenation)되는 연결 방식을 갖는 것이 이전의 모델들과 큰 차이점이다 (Haug et al., 2017). DenseNet의 구조는 신경망의 깊이가 깊어질수록 입력데이터(input data)에

Ⓢ: Channel-wise concatenation

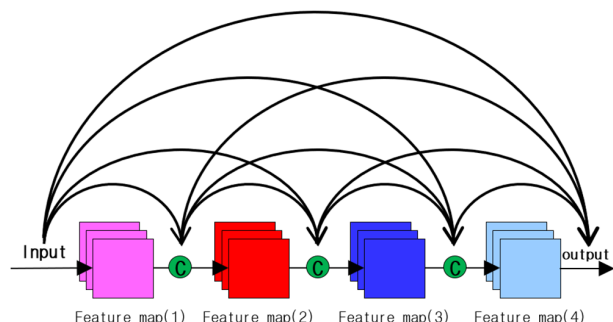


Fig. 4 DenseNet connectivity diagram (Haug et al., 2017)

대한 정보가 사라지는 문제(wash out)를 방지하기 위해 특징맵을 계속 쌓아간다. 이는 오차역전파(error backpropagation)의 효율을 극대화하며 학습(train)을 쉽게 해준다. 또한, 신경망의 깊이 대비 적은 합성곱필터를 사용하기 때문에 이전 CNN 모델들보다 파라미터 수가 적어 학습에 소모되는 시간을 단축시킬 수 있었다. 또한 파라미터가 많을 때의 문제점도 보완할 수 있다는 장점이 있다. 하지만 깊이가 깊어질수록 연산량이 증가하는 단점을 가지고 있다. 본 연구에서는 DenseNet모델의 Densenet-201 구조를 사용하였다.

2.5 NASNet

NASNet은 구글에서 만든 Auto Machine Learning (AutoML) 알고리즘을 이용한 CNN 모델이다 (Zoph et al., 2018). 기본원리는 RNN컨트롤러(Recurrent Neural Network Controller)가 합성곱레이어의 필터 크기, 스트라이드 등 변수를 신경망 층마다 임의로 설정하여 학습을 진행한다. 이렇게 구성된 신경망의 학습 정확도를 다시 RNN컨트롤러가 학습하여 더 좋은 성능을 가질 수 있도록 최적의 변수를 찾는다. NASNet은 CIFAR-10 데이터셋에 한하여 신경망이 구성되었음에도 이미지넷 데이터셋을 이용한 실험 결과 높은 정확도를 보여주었으며 여러 데이터셋들에 대해 유연하게 전이할 수 있다(transferable)는 장점이 있다. 하지만 RNN컨트롤러를 이용해 강화학습을 통한 최적의 변수를 찾기 위해서는 대량의 고비용 GPU들로 구성된 시스템이 필요한 단점이 있다. 본 연구에서는 Zoph et al.(2018)이 Neural Architecture Search (NAS)의 RNN컨트롤러를 이용해 최적의 값으로 출력한 NASNetLarge구조를 사용하였다.

3. 전이학습

사전에 구성된 CNN 모델을 사용할 때에는 적용하려는 영상 데이터가 CNN 모델의 설계에 사용된 학습 데이터(source data)와 유사성을 가져야 좋은 성능을 가질 가능성이 높다. 따라서 CNN 모델의 설계 사용된 학습 데이터와 겹치지 않는 다른 학습 데이터를 사용할 경우 성능이 좋지 않을 수도 있다. 이런 문제점을 보완하기 위해 전이학습(transfer learning)을 사용할 수 있다 (Pan & Yang, 2009). 전이학습에서 사전 훈련된 네트워크를 사용하는 방법으로 크게 특징추출(feature extraction)과 미세조정(fine tuning) 두 가지 방법을 사용할 수 있다. 본 연구에서는 미세조정방법을 사용하였다.

Fig. 5의 상단에 보이는 특징추출을 이용하여 사전 훈련된 네트워크를 활용하는 방법은 합성곱기반층(convolutional base)의 가중치를 동결시킨 상태에서 입력을 통과시키고 분류기(classifier)에서 그 출력을 사용하여 분류 훈련을 하는 방법이다 (Donahue et al., 2014).

미세조정을 통하여 사전 훈련된 네트워크를 활용하는 방법 중 하나는 본 연구에서 사용한 방법으로 특징추출계층의 합성곱기

Table 1 Comparison of CNN models

CNNs	Feature	Pros	Cons
VGGNet	- Simplicity and depth	- Easy network modification	- Too many parameters
GoogLeNet	- Inception module with dimension reductions - No use of fully connected layers	- Saving a huge number of parameters	- Difficult network modification due to complex architecture
ResNet	- Residual architecture - 8 times deeper than VGGNet, 152 layers	- Good learning performance without increase of the number of parameters	- Occurrence of vanishing gradient problem with deeper depth
DenseNet	- Connects each layer to every other layer in a feed-forward fashion	- Substantially reduce the number of parameters	- Increase of computational cost with deeper depth
NASNet	- The normal and reduction cells are searched by the RNN controller	- Can be transferred to ImageNet classification without much modification	- High-end GPUs for computation of the RNN controller

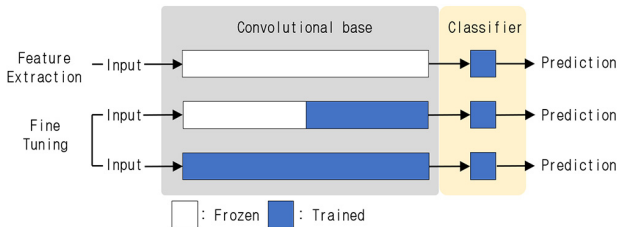


Fig. 5 Transfer learning strategies

반층 중 일부를 동결하지 않고 훈련하는 방법이다 (Yosinski et al., 2014). Fig. 5의 중앙에 보이는 것처럼 이 방법은 새로 학습할 입력에 대하여 합성곱기반층 중 동결되지 않은 가중치 초기값은 사전 훈련된 가중치를 사용하고 훈련에 따라 변화하게 된다. 본 연구에서는 대상 CNN 모델들의 합성곱기반층 중 입력층 쪽 5개 합성곱레이어의 가중치를 동결하였다.

또 다른 미세조정 방법으로는 Fig. 5의 하단에 보이는 것과 같이 합성곱기반층 전체를 학습시킬 수도 있다. 다만 학습 과정에서 학습률(learning rate)을 너무 크게 설정하거나 학습횟수(epoch)를 많이 설정하면 사전 학습 내용을 모두 망가뜨릴 수 있어 주의해야 한다.

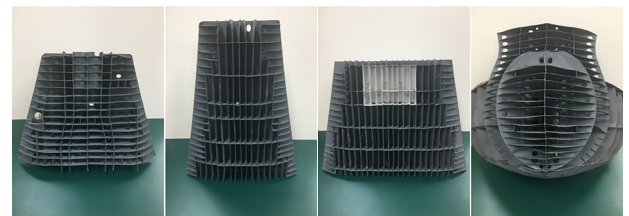
4. 다시점 영상 집합 및 실험환경 구성

4.1 다시점 영상 집합

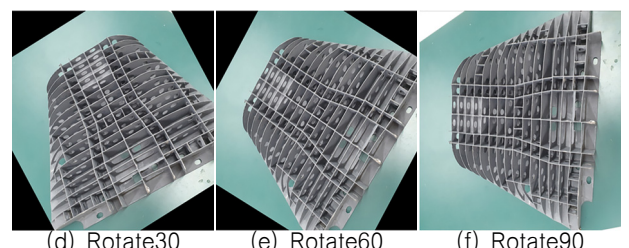
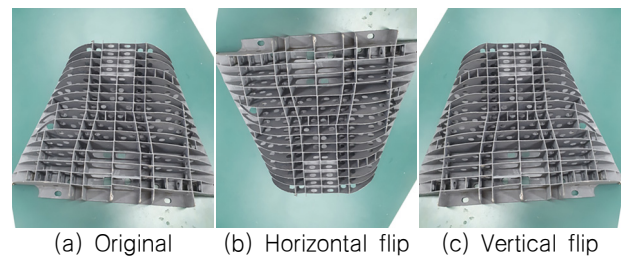
선체 블록 분류를 위해 직접 대형조선소의 블록 적치장에 있는 블록 영상을 활용하는 것이 가장 좋다고 판단되나 보안 및 현장 상황에 따른 영상 확보의 어려움이 있었다. 또한, 인터넷 영상을 사용하여 확보하는 것은 특정 블록의 다양한 방향에서의 모습을 획득하기에 한계가 있고 해당 블록에 대한 식별 영상을 확보하는 것도 문제가 있다. 따라서 본 연구에서는 실제 설계도를 기반으로 만들어진 선체 블록 모형을 이용하였으며 실험실 환경에서 직접 선체 블록 모형에 대해 다양한 방향에서의 영상을 획득하여 훈련 및 식별 영상 집합(prediction image set)으로 사용

하였다. 분류하고자 하는 선체 블록 모형은 Fig. 6와 같이 유사한 형상이지만 길이와 폭의 비가 다르고 내부적 특징이 다른 세 종류의 블록A, 블록B, 블록C와 전혀 다른 형상의 블록D로 구성 되어 있다. 네 개의 블록은 각 500장씩 총 2,000장의 영상을 획득하여 기본 영상 집합(original image set)으로 할당하였다.

CNN 모델의 학습정확도는 영상 집합의 크기와 비례할 수 있으므로 데이터 확장(data augmentation)을 적용해 영상 개수를 증가시켰다. 기본 영상 집합을 상하, 좌우로 반전시키고 30°, 60°, 90°씩 회전하였다. 1장의 영상을 6장으로 확장시켜 총 12,000장으로 이루어진 확장 영상 집합(augmented image set)을



(a) Ship block A (b) Ship block B (c) Ship block C (d) Ship block D
Fig. 6 Ship block models



(a) Original (b) Horizontal flip (c) Vertical flip
(d) Rotate30 (e) Rotate60 (f) Rotate90
Fig. 7 Data augmentation

Table 2 The number of original and augmented images

Class	Original		Augmented	
	train	test	train	test
A	400	100	2400	600
B	400	100	2400	600
C	400	100	2400	600
D	400	100	2400	600

만들었다. Fig. 7은 선체 블록 A의 기본 영상을 확장한 영상의 예시이다. 영상 집합은 8:2의 비율로 학습(train)에 80%를, 평가(Test)에 20%를 할당하였다. Table 2는 4개의 A, B, C, D 블록에 해당하는 클래스에 할당된 기본과 확장 영상의 개수이다.

4.2 실험 환경 구성

5가지 CNN 모델과 사용된 영상 집합의 종류 그리고 미세조정 전이학습의 적용 여부인 4가지 경우(case)를 조합하여 20가지의 실험환경을 구성했다. 일반적으로 데이터의 개수가 많지 않을 때는 데이터 확장한 영상 집합을 사용한 경우가 기본 영상 집합을 사용한 경우보다 학습의 완성도가 높아 분류 정확도가 더 높게 나온다. 따라서 데이터 확장 영상 집합으로만 학습하는 것이 효율적일 수 있다. 다만 본 연구에서는 향후 조선소에서 수천 개의 선체 블록별 영상 집합을 획득하는 상황을 고려하여 기본 영상 집합을 활용한 학습 결과와 데이터 확장 영상 집합을 활용한 학습 결과의 차이가 크지 않다면 더 많은 영상 집합을 구성해야 하고 더 긴 학습 시간이 소요되는 데이터 확장을 적용할 필요가 없을 수도 있다고 판단하여 두 경우의 정량적 차이를 확인하고자 기본 영상 집합과 데이터 확장 영상 집합의 두 경우 모두를 학습하여 결과를 비교하였다.

각 CNN 모델의 완전연결층(Fully Connected, FC) 중 마지막 완전연결층을 제외한 나머지 완전연결층은 전역평균풀링(Global Average Pooling, GAP)으로 대체하였다. 전역평균풀링을 사용하면 과적합 및 많은 계산량 문제를 해결할 수 있다 (Park & Kim, 2018). 또한 완전연결층을 사용하면 클래스의 위치 정보가 손실되지만 전역평균풀링은 클래스의 위치 정보를 유지할 수 있도록 한다 (Bolei et al., 2015). 이를 이용해 CNN 모델이 클래스의 특징을 추출할 때 어떤 영역에 주목하고 있는지 가시화할 수 있는 Class Activation Map (CAM)을 사용하여 실험 결과 분석에 사용하였다 (Zhou et al., 2016). 그리고 드롭아웃(Dropout)을 적용하여 CNN 모델이 과적합을 억제할 수 있는 정규화(regularization) 효과를 가질 수 있도록 했다 (Srivastava et al., 2014). 드롭아웃 비율로 0.4를 사용하였다. 이는 가중치(weight)의 동조화 현상(co-adaptation)을 피해 특징을 좀 더 선명하게 얻을 수 있도록 해준다 (Zeiler & Fergus, 2014). 손실함수(loss function)는 범주형 크로스 엔트로피(categorical cross entropy)를 사용하였고 최종 출력 활성화 함수는 소프트맥스(softmax)를 사용하였다. 매개 변수의 갱신은 실효값전파(Root Mean Square Propagation,

RMSprop)를 사용하였고 학습률은 $2e-5$ 로 설정하였다.

각 CNN 모델별로 사용한 영상 크기, 학습 횟수와 배치크기(batch size)는 Table 3과 같다. 워크스테이션에서의 메모리 초과(memory out) 문제를 해결하기 CNN 모델별로 배치크기를 다르게 하였다. 스텝당 에포크(Epoch per step)는 전체 훈련 영상 개수를 배치크기로 나눈 값을 사용해 모든 영상을 훈련할 수 있도록 했다. 사용된 소프트웨어와 하드웨어 사양은 Table 4와 같다.

Table 3 Image size, epoch and batch size (TL=With Transfer Learning)

CNN model	Image size	Epoch	Batch size
VGG-19	224×224	10	64
VGG-19_TL	224×224	1	64
Inception-v3	299×299	10	64
Inception-v3_TL	299×299	1	64
Resnet-152v2	224×224	10	32
Resnet-152v2_TL	224×224	1	32
Densenet-201	224×224	10	16
Densenet-201_TL	224×224	1	16
NASNetLarge	331×331	10	8
NASNetLarge_TL	331×331	1	8

Table 4 Software and hardware specification

	Version
OS(Operating System)	Ubuntu 16.04 LTS
TensorFlow	1.14.0
CUDA	10.0
Keras	2.3.1
CPU	intel Core i9-7900X 3.30 GHz
GPU	GeForce RTX 2080 TI × 4
RAM	128 GB (16 GB × 8)

5. 실험 결과 및 분석

5.1 실험 결과

본 연구에서는 전체 20가지 실험환경의 학습에 대한 파라미터를 각 CNN 모델별로 획득하였다. 각 CNN 모델별 분류 성능 평가를 위한 식별 영상 집합은 블록의 전체 형상이 포함된 식별 영상 집합으로 블록 당 30장씩 총 120장으로 구성하였다.

클래스를 예측한 결과는 식별 영상 집합의 각 영상별 실제 클래스(actual class)와 예측클래스(predicted class)가 Fig. 8과 같이 표현된다. Fig. 8(a)에서 보인 예제는 실제클래스가 A일 때 가장 높은 확률로 클래스 C로 예측한 경우를 나타낸 것이며 Fig. 8(b), (c), (d)는 실제클래스 B, C, D의 경우에 각각 가장 높은 확률로 클래스 B, C, D로 예측한 경우를 나타낸 것이다.

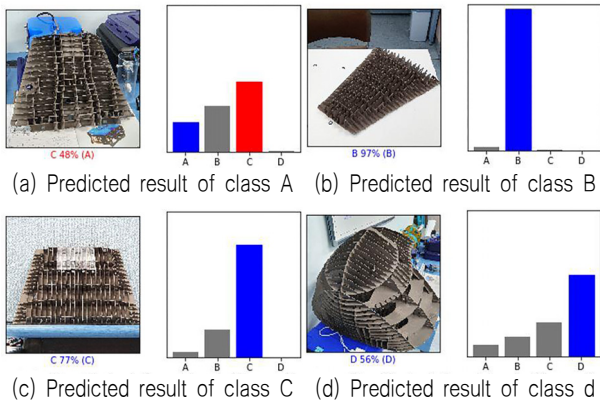


Fig. 8 Predicted results

n 개 클래스의 분류성능지표를 계산하기 위해 실제클래스에 대한 클래스 예측결과를 정리하여 혼동행렬(confusion matrix)을 Fig. 9와 같이 구성할 수 있다. 혼동행렬의 원소 x_{ij} 는 입력된 영상의 실제클래스가 i 번째 클래스이고 예측은 j 번째 클래스로 한 경우의 수를 의미한다. 본 연구에서는 분류 클래스가 4가지이므로 $n=4$ 의 경우에 해당한다. 그리고 혼동행렬은 전체 20가지의 실험환경에서 각각 한 개씩 가지게 되므로 전체 20개의 혼동행렬이 구성된다.

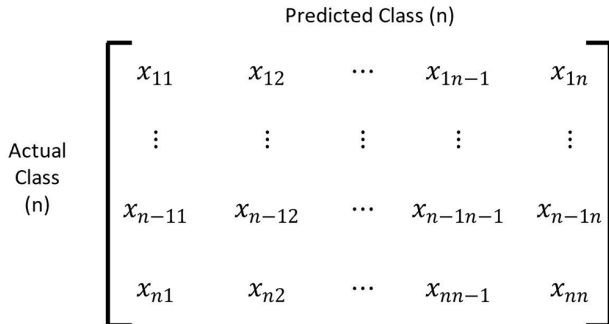


Fig. 9 Confusion matrix of multi-class classification

혼동행렬을 사용하여 n 개의 분류 클래스에 대하여 정확도(accuracy), 정밀도(precision), 재현율(recall)과 평균 F1-Score(Average F1-Score)의 4가지 분류성능지표를 산출할 수 있다. 4가지 성능지표는 각각 식 (2), 식 (3), 식 (4)와 식 (5)로 표현된다. 평균 F1-Score는 정밀도와 재현율의 조화평균으로 계산되므로 정밀도와 재현율의 특성을 모두 고려하는 지표라고 할 수 있다. 본 연구에서는 분류성능지표로서 정확도와 평균 F1-Score를 사용하였다.

정확도와 평균 F1-Score의 계산을 위해서는 혼동행렬을 이용한 TP(True Positive), FP(False Positive), TN(True Negative), FN(False Negative)의 계산이 필요하다. TP는 실제클래스가 i 일 때(Positive) 예측클래스 j 가 i 와 같은($j=i$) 참(True)인 경우로 식 (6)과 같이 표현된다. FP는 실제클래스가 i 일 때 예측클래스 j 가 i 가 아닌($j \neq i$) 거짓(False)인 경우로 식 (7)과 같이 표현된다. FN는 실제클래스가 i 가 아닐 때(Negative) 예측클래스 j 가 i 인($j=i$) 거짓(False)인 경우로서 식 (8)과 같이 표현된다. TN는 실제클래스가 i 가 아닐 때 예측클래스 j 가 i 가 아닌($j \neq i$) 참(True)인 경우로서 식 (9)와 같이 표현된다.

스 i 가 아닐 때 예측클래스 j 가 i 가 아닌($j \neq i$) 참(True)인 경우로서 식 (9)와 같이 표현된다.

20가지 실험환경에서의 전체 블록 형상이 포함된 식별 영상 집합을 이용한 정확도는 Table 5와 Fig. 10에 나타내고, 평균 F1-Score는 Table 6와 Fig. 11에 나타내었다. 실험 결과에 따르면 확장 영상 집합을 사용하고 전이학습을 적용한 Resnet-152v2 CNN 모델이 정확도와 평균 F1-Score가 각각 99.17%와 99.18%로 20가지 환경에서 가장 높았으며, 기본 영상 집합을 사용하고 전이학습을 적용하지 않은 VGG-19 CNN 모델의 정확도와 평균 F1-Score가 각각 15%와 9.61%로 20가지 환경에서 가장 낮았다.

$$Accuracy(TP_i) = \frac{\sum_{i=1}^n TP_i}{\sum_{i=1}^n \sum_{j=1}^n x_{ij}}, i = 1, 2, \dots, n \quad (2)$$

$$Precision(TP_i, FP_i) = \frac{TP_i}{TP_i + FP_i}, i = 1, 2, \dots, n \quad (3)$$

$$Recall(TP_i, FN_i) = \frac{TP_i}{TP_i + FN_i}, i = 1, 2, \dots, n \quad (4)$$

$$Avg F1 - Score = 2 \times \left(\frac{Avg Precision \times Avg Recall}{Avg Precision + Avg Recall} \right) \quad (5)$$

$$= 2 \times \left\{ \frac{\frac{1}{n} \left(\sum_{i=1}^n \frac{TP_i}{TP_i + FP_i} \right) \times \frac{1}{n} \left(\sum_{i=1}^n \frac{TP_i}{TP_i + FN_i} \right)}{\frac{1}{n} \left(\sum_{i=1}^n \frac{TP_i}{TP_i + FP_i} \right) + \frac{1}{n} \left(\sum_{i=1}^n \frac{TP_i}{TP_i + FN_i} \right)} \right\}$$

$$TP_i = x_{ii}, i = 1, 2, \dots, n \quad (6)$$

$$FP_i = \sum_{j=1}^n (x_{ji}) - x_{ii}, i = 1, 2, \dots, n \quad (7)$$

$$FN_i = \sum_{j=1}^n (x_{ij}) - x_{ii}, i = 1, 2, \dots, n \quad (8)$$

$$TN_i = \sum_{i=1}^n \sum_{j=1}^n x_{ij} - \sum_{j=1}^n x_{ij} - \sum_{j=1}^n x_{ji} + x_{ii}, i = 1, 2, \dots, n \quad (9)$$

전이학습을 적용하지 않고 기본 영상 집합으로 학습된 경우 정확도는 Inception-v3 CNN 모델이 34.17%로 가장 높고 평균 F1-Score는 Resnet-152v2 CNN 모델이 31.81%로 가장 높다. 전이학습을 적용하지 않고 확장 영상 집합으로 학습된 경우 VGG-19 CNN 모델이 정확도 38.33%와 평균 F1-Score 43.20%로 가장 높은 성능을 보여주고 있다.

전이학습을 적용하고 기본 영상 집합으로 학습된 경우에는 NASNetLarge CNN 모델이 정확도 70.00%와 평균 F1-Score 70.36%로 가장 높은 성능을 보여주고 있다. 전이학습과 기본 영

Table 5 Classification performance comparison of accuracy (NTL=Non Transfer Learning, TL=Transfer Learning, OI=Original Image set, DA=Data Augmentation)

Case \ CNN Model	VGG-19	Inception-v3	Resnet-152v2	Densenet-201	NASNetLarge
NTL+OI	15.00%	34.17%	30.00%	26.67%	25.00%
NTL+DA	38.33%	27.50%	32.50%	27.50%	33.33%
TL+OI	43.33%	33.33%	55.83%	61.67%	70.00%
TL+DA	97.50%	89.17%	99.17%	95.83%	98.33%

Table 6 Classification performance comparison of average F1-Score (NTL=Non Transfer Learning, TL=Transfer Learning, OI=Original Image set, DA=Data Augmentation)

Case \ CNN Model	VGG-19	Inception-v3	Resnet-152v2	Densenet-201	NASNetLarge
NTL+OI	9.61%	23.40%	31.81%	21.00%	10.00%
NTL+DA	43.20%	30.58%	21.79%	23.47%	29.42%
TL+OI	41.02%	40.44%	59.38%	62.20%	70.36%
TL+DA	97.61%	89.91%	99.18%	95.83%	98.33%

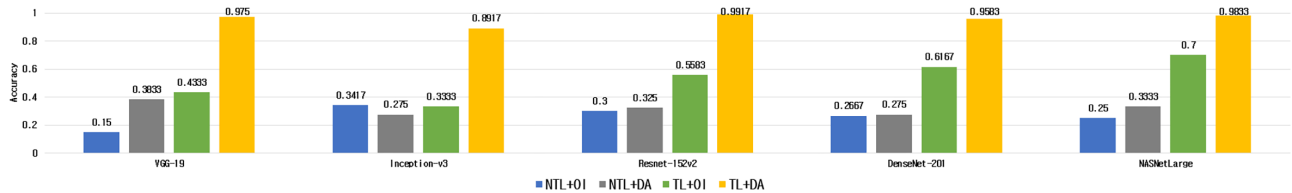


Fig. 10 Classification performance comparison of accuracy (NTL=Non Transfer Learning, TL=Transfer Learning, OI=Original Image set, DA=Data Augmentation)

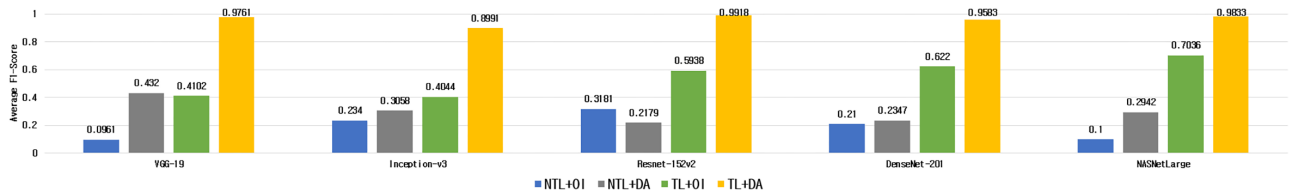


Fig. 11 Classification performance comparison of average F1-Score (NTL=Non Transfer Learning, TL=Transfer Learning, OI=Original Image set, DA=Data Augmentation)

상 집합으로 학습된 CNN 모델에서는 Resnet-152v2, Densenet-201와 NASNetLarge 순서로 정확도와 평균 F1-Score가 증가하는 경향을 보여주었다. 다만 전이학습과 확장 영상 집합으로 학습된 CNN 모델에서는 Inception-v3를 제외한 나머지 모델들 모두 정확도와 평균 F1-Score는 95% 이상으로 급격한 상승을 보여주었으며 Inception-v3, Densenet-201, VGG-19, NASNetLarge와 Resnet-152v2 순서로 정확도와 평균 F1-Score가 높아지는 경향을 보여주었다.

5.2 실험 결과 분석

5가지 CNN 모델들의 4가지 실험환경에서의 성능 차이의 원인을 분석하기 위해 CAM을 사용하였다. CAM은 분류의 기준이 되는 가장 차별적인 특징들을 가시화하여 영상의 어느 영역이 예측클래스 특징을 갖는지 보여준다. 이러한 CAM 결과를 이용하

면 동일한 식별 영상에 대하여 각각의 실험환경에서의 클래스를 예측하는데 있어 필요한 차별적인 특징들이 집중하고 있는 영역을 비교할 수 있다.

Fig. 12에서는 Table 5의 4가지 실험환경 중에서 낮은 정확도를 보여주고 있는 NTL+OI 환경(전이학습을 적용하지 않고 기본 영상 집합을 사용한 경우)에서의 CAM 결과를 나타내고 있다. Fig. 13에서는 Table 5의 4가지 실험환경 중에서 가장 높은 정확도를 보여주고 있는 TL+DA 환경(전체적으로 전이학습을 적용하고 확장 영상 집합을 사용한 경우)에서의 CAM 결과를 나타내었다.

NTL+OI 환경에서의 식별 결과는 TL+DA 환경에서의 식별 결과보다 정확도와 평균 F1-Score가 낮는데 이는 Fig. 12와 Fig. 13을 비교함으로써 추정해 볼 수 있다. Fig. 12의 각 NTL+OI 환경에서의 CNN 모델들의 CAM 결과를 보면 차별적인 특징을 가시화한 영역이 영상의 배경을 포함하거나 그 범위가 전체적으로 퍼져있는 양상으로 나타나고 있음을 알 수 있다. 이는 학습이 완

성되지 못했음을 의미한다고 볼 수 있다. Fig. 13은 Fig. 12의 CAM 결과와 비교해 볼 때 객체영역(Object region)에 차별적인 특징을 나타내는 영역이 집중되어 분포하고 있음을 확인할 수 있었다. 따라서 CNN 모델들이 TL+DA 환경보다 NTL+OI 환경에서 훈련 영상으로부터 선체 블록의 특징을 잘 학습하지 못하고 있다고 분석할 수 있다.

Fig. 14는 가장 높은 정확도를 보여준 Resnet-152v2 CNN 모델의 전이학습 적용 여부와 사용된 영상 집합의 종류를 고려한 4가지 경우에 대해 Table 5의 정확도가 증가된 순서로 CAM의 변화를 나타내고 있다. Fig. 14의 (a)부터 (d)로 갈수록 정확도가 증가하는 순서대로 나열한 것이다. 전이학습을 적용하지 않고 기본 영상 집합을 사용한 'NTL+OI Resnet-152v2'는 특징 추출 영역이 배경에도 흩어져 분포되어있다. 정확도가 증가하는 우측으로 갈수록 특징 추출 영역이 객체영역에 집중되는 것을 확인할 수 있다. Fig. 14의 (a)보다는 (b)가 (c)보다는 (d)에서 특징 추출 영역이 객체영역에 더욱 집중됨을 알 수 있다. 따라서 데이터 확장을 통해 영상의 개수를 증가시킨 확장 영상 집합을 사용하고 전이학습을 적용한 CNN 모델일수록 특징추출을 위한 영역이 객체영역에서 집중적으로 선정되므로 정확도가 증가하는 경향을 나타내게 되는 요인으로 파악할 수 있다.

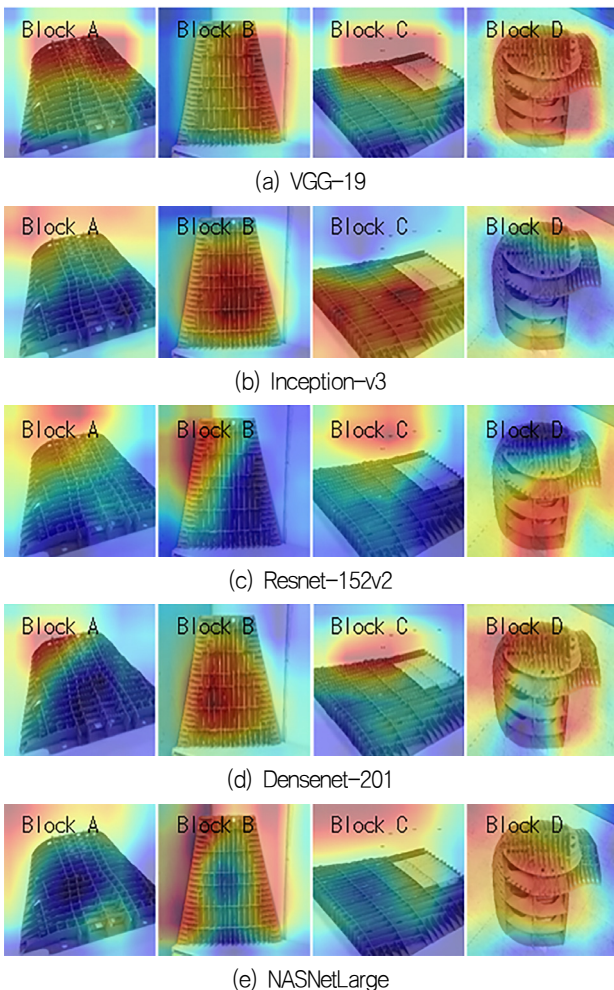


Fig. 12 CAM results of NTL+OI cases

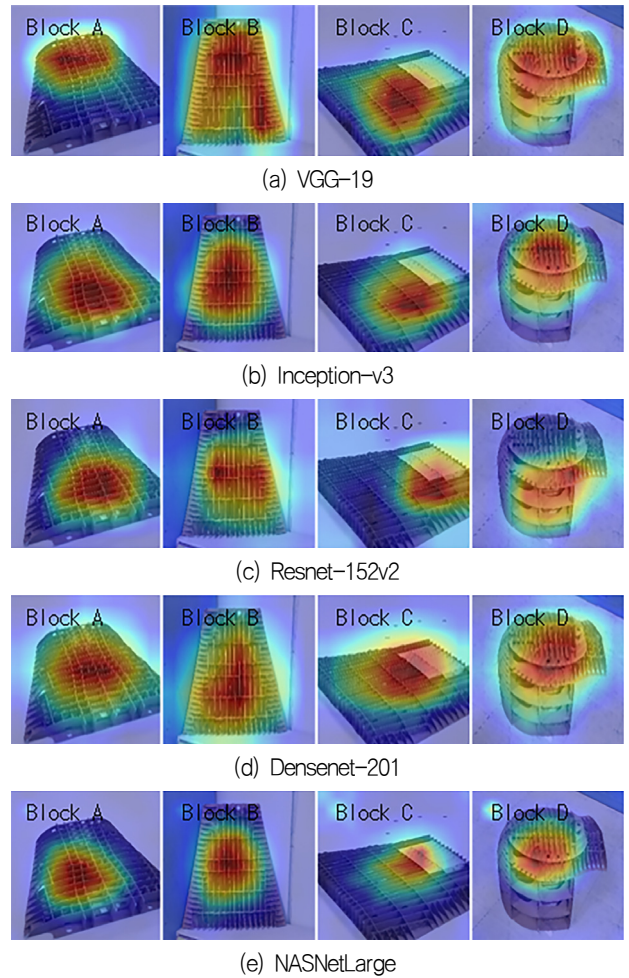


Fig. 13 CAM results of TL+DA cases

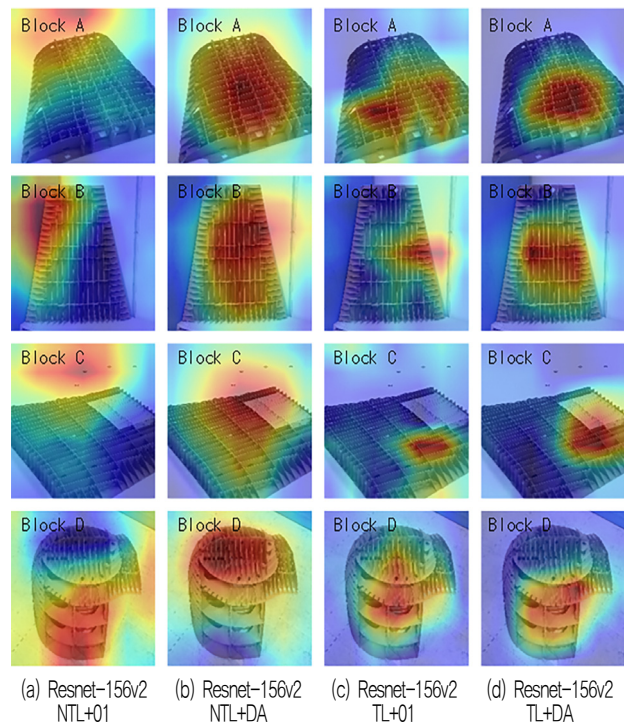


Fig. 14 CAM comparison of Resnet-152v2

현재 실험 결과에 의하면 CNN 모델의 학습에 적은 수의 블록 영상만으로 구성된 기본 영상 집합을 사용한 경우에는 파라미터가 우수한 분류 성능을 나타내는 값으로 학습되지 않았다고 할 수 있다. 다만 본 실험에서 사용한 블록 영상만으로 구성된 기본 영상 집합의 크기가 전이학습의 적용 시 사용하는 초기 파라미터 값을 구성하는 이미지넷 데이터셋 만큼 크지 않아 전이학습의 영향이 더욱 크게 작용하고 있다고 할 수 있다. 만약 블록 영상만으로 구성된 기본 영상 집합의 데이터 크기가 이미지넷 데이터셋 크기 정도로 크다면 각 CNN 모델의 파라미터가 충분히 학습되어 전이학습을 적용하지 않아도 높은 분류 성능을 보여 줄 수도 있다고 판단된다.

실험 결과인 정확도와 평균 F1-Score의 가장 높은 값은 각각 99.17%와 99.18%로 상당히 높게 산출되었다. 이는 선체 블록 모형의 식별 영상 집합은 전체 형상이 포착된 영상으로 구성되어 있기 때문에 볼 수 있다. 실제 조선소에서 획득할 수 있는 선체 블록 영상은 한 화각에 전체 형상을 포착하기 어려울 수 있다. 따라서 선체 블록 모형의 전체 형상이 포착되는 식별 영상 집합을 이용해 산출한 정확도와 평균 F1-Score가 실제 선체 블록을 분류하는 성능과 같을 것이라고 일반화하기 어렵다.

이러한 실제 획득 블록 영상과의 간극을 줄이기 위한 방안 중 하나는 블록 전체 형상이 아닌 블록 일부 형상으로만 구성된 영상 집합을 구성하고 이를 식별 영상 집합으로 사용하여 분류 성능의 변화를 살펴보는 것이다. 실제 조선소 적치장의 블록의 일부분만을 영상으로 획득하였을 때를 가정하여 기존 식별 영상에서 블록의 전체 형상이 드러나지 않도록 블록 영역의 일부분을 자른 식별 영상 집합을 만들었다. 이렇게 블록 일부만으로 구성된 식별 영상 집합을 이용하여 각 CNN 모델의 식별 성능에 대하여 평가했다. 다만 모든 실험환경에서의 평가보다는 가장 높은 정확도와 평균 F1-Score를 보여주는 TL+DA(확장 영상 집합을 사용하고 전이학습을 적용) 환경에서만 실험하였다.

Fig. 15의 (a), (b), (c) 와 (d)는 각 블록 A, B, C와 D의 식별 영상집합 영상 중 하나이며 (e), (f), (g)와 (h)는 식별영상에서 블록 영역의 일부분 만으로 구성되어 영상에 배경이 없고 블록의 격자 구조물만 특징으로 보이도록 잘라낸 것이다.

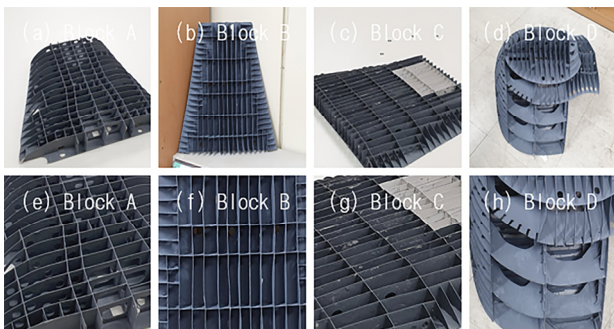


Fig. 15 Cropped prediction image set

블록 일부만으로 구성된 식별 영상 집합을 이용해 각 CNN 모

델들의 정확도와 평균 F1-Score를 산출한 결과는 Table 7에 나타내었다. 블록 일부만으로 구성된 식별 영상 집합을 이용한 분류 성능지표 산출 결과 Resnet-152v2 모델이 정확도 81.67%와 평균 F1-Score 83.29%로 가장 우수한 성능을 보여주었다. 정확도는 전체 형상 식별 영상 집합의 경우에 비교하여 가장 높은 정확도가 17.5% 감소하였다. 그 외에 VGG-19, Inception-v3, NASNetLarge, Densenet-201과 Resnet-152v2순서로 정확도가 높아지는 결과를 볼 수 있었는데 이는 식별 영상 집합을 사용했을 때의 정확도가 높은 순서와 비교해 볼 때 VGG-19와 Inception-v3의 결과가 바뀌었으나 나머지 모델들은 동일한 순서를 보여주고 있음을 알 수 있다.

Table 7 Classification performance comparison of accuracy and average F1-Score with cropped prediction image set

CNN model	Accuracy	Average F1-Score
VGG-19	58.33%	64.32%
Inception-v3	60.00%	66.52%
Resnet-152v2	81.67%	83.29%
Densenet-201	81.67%	82.43%
NASNetLarge	66.67%	73.71%

이는 본 실험에서 분류 성능이 가장 뛰어난 CNN 모델은 Resnet-152v2라고 할 수 있으나 실제 조선소의 블록 획득 영상에 따라 블록의 식별 성능의 차이는 존재할 수 있다는 것을 의미한다. 따라서 본 연구의 실험 결과가 실제 블록의 분류 성능에도 완전히 동일하게 일반화하여 적용할 수 있다고 확정적으로 결론을 내릴 수 있음을 의미하는 것은 아니다. 다만 CNN 모델들의 분류 성능에 관한 모든 벤치마크 실험은 학습에 사용한 데이터셋의 도메인과 식별 영상 집합의 도메인이 유사할 경우에는 실험 결과를 참고할 수는 있으므로 본 실험의 결과는 선체 블록 분류 도메인에서 참고할 수는 있을 것으로 생각된다.

6. 결론

본 연구에서는 블록 적치장에 적치된 선체 블록의 위치를 인식하고 식별할 수 있는 시스템에 적용하기 위한 기초연구로서 학습 영상 집합을 가지고 있는 경우를 가정한 CNN 모델 적용 시 높은 정확도를 가지는 블록 식별에 적합한 CNN 모델을 찾기 위한 다양한 CNN 모델의 식별성능에 관한 비교실험을 수행하였다. 학습 영상 집합을 위하여 선체 블록 모형을 제작하였고 이를 이용하여 다시점 영상 집합을 획득한 다음 학습을 통해 다양한 CNN모델들의 선체 블록 식별 성능을 비교하였다. 비교에 사용한 CNN 모델로는 VGGNet의 VGG-19구조, GoogLeNet의 Inception-v3 구조, ResNet의 Resnet-152v2구조, DenseNet의 Densenet-201구조, NASNet의 NASNetLarge구조가 사용되었다.

학습을 위한 기본 영상 집합을 선체 블록 당 500장씩 총 2,000장으로 만든 다음 기본 영상 집합만으로 이루어진 학습, 기

본 영상 집합을 데이터 확장한 확장 영상 집합을 적용한 학습 그리고 각 경우에 미세조정 전이학습을 적용한 학습을 수행하였다. 이는 각 CNN 모델별로 전이학습의 적용 여부와 학습에 사용된 영상 집합의 종류에 따라 총 4가지 경우의 성능을 평가하기 위함이다. 성능 평가는 선체 블록 당 30장씩 총 120장으로 이루어진 식별 영상 집합에 대해 정확도와 평균 F1-Score를 산출하여 평가지표로 사용하였다.

평가결과 ResNet의 Resnet-152v2 CNN 모델에 확장 영상 집합을 사용하고 전이학습을 적용한 결과가 정확도 99.17%, 평균 F1-Score 99.18%로 가장 우수하였다.

또한, 평가결과에 대한 분석을 위해 CAM을 사용하여 가장 정확도가 높은 CNN 모델과 낮은 CNN 모델 그리고 가장 정확도가 높은 CNN 모델에서 학습에 사용한 영상 집합 및 전이학습을 고려한 4가지 실험환경에 대해서 특징추출영역을 확인하였다. 정확도가 높은 CNN 모델일수록 그리고 영상의 개수가 많은 학습 데이터셋일수록 객체영역에 특징추출영역이 밀도 있게 집중하여 분포하는 것을 확인할 수 있었다.

그리고 적치장의 실제 획득 블록 영상과의 간극을 줄이기 위한 방안 중 하나로서 블록 전체 형상이 아닌 블록 일부 형상으로만 구성된 영상 집합을 구성하고 이를 식별 영상 집합으로 사용하여 분류 성능의 변화를 추가로 실험하여 살펴보았다. 결론적으로 가장 높은 분류 성능은 전체 형상 식별 영상 집합을 사용한 경우와 동일하게 Resnet-152v2가 가장 높은 정확도를 나타내었다.

본 성능평가 실험 결과를 모든 경우에 일반화할 수는 없으나 직선과 격자무늬가 많이 존재하는 선체 블록을 분류 및 식별하는 경우에는 ResNet의 Resnet-152v2 CNN 모델을 적용한다면 타 구조보다 더 나은 성능을 얻을 것으로 기대할 수 있다고 판단된다.

다만 본 연구에서는 학습 영상 집합을 얻기 위해서 선체 블록 모형을 제작하여 학습에 필요한 영상 집합을 획득하였으나 실제 시스템에서 적용하기 위해서는 모든 블록의 모형을 만들 수는 없으므로 다른 방법을 고려해야 할 것이다. 특히 이를 해결할 수 있는 대안 중 하나의 방법으로는 CAD 설계 데이터를 활용하여 3차원 다시점 영상 집합으로 이루어진 학습용 영상 집합을 얻는 것을 고려해 볼 수 있을 것이다. 향후 연구에서는 선체 블록 도면의 3차원 CAD 데이터를 영상화하여 학습용 다시점 영상 집합을 구성하고 데이터 확장 및 전이학습을 적용한 Resnet-152v2 모델을 통해 학습하고 식별하는 과정을 통해 좀 더 조소한 환경에서 실질적으로 적용 가능한 방법을 연구할 필요가 있다고 생각된다.

후 기

This work was supported by the Korea Institute of Energy Technology Evaluation and Planning(KETEP) and the Ministry of Trade, Industry & Energy(MOTIE) of the Republic of Korea (No.20183010025200).

References

- Byeon, Y.H. & Kwak, K.C., 2018. A transfer learning and performance comparison of deep learning models for pedestrian classification under automobile driving environment. *The Journal of Korean Institute of Information Technology*, 16(10), pp.83-92.
- Cho, D.Y., Song, H.C. & Cha, J.H., 2011. Block and logistics simulation. *Bulletin of the Society of Naval Architects of Korea*, 48(4), pp.24-29.
- Donahue, J. et al., 2014. Decaf: A deep convolutional activation feature for generic visual recognition. *In International conference on machine learning*, Beijing, China, pp.647-655.
- Francois, C., 2018. *Deep learning with Python*, Gilbut Publichin Co., Ltd.
- 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*, Las Vegas, United States of America, pp.770-778.
- Huang, G., Liu, Z., Van Der Maaten, L. & Weinberger, K.Q., 2017. Densely connected convolutional networks. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pp.4700-4708.
- Kang, J.H., 2014. *A study on mobile block logistics system for shipyard*. Master Thesis, Mokpo National University.
- Kim, B.C., 2018. Shape recognition of plant equipment from 3-D scanned point cloud data using a convolutional neural network. *Transactions of the Korean Society of Mechanical Engineers, A* 42(9), pp.863-869.
- Kim, J.O. et al., 2009. Development of real time location measuring and logistics system for assembled block in shipbuilding. *Korean Institute of Industrial Engineers*, pp.834-839.
- Kim, M.S., Cha, J.H. & Cho, D.Y., 2013. Determination of arrangement and take-out path in ship block stockyard considering available space and obstructive block. *Society for Computational Design and Engineering*, pp.433-438.
- Krizhevsky, A., Sutskever, I. & Hinton, G. E., 2012. Imagenet classification with deep convolutional neural networks. *In Advances in neural information processing systems*, pp.1097-1105.
- LeCun, Y., Bottou, L., Bengio, Y. & Haffner, P., 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), pp.2278-2324.
- Lee, Y.H., Lee, K.C., Lee, K.J. & Son, Y.D., 2008. Study on the positioning system for logistics of ship-block. *Special Issue of the Society of Naval Architects of Korea*, pp.68-75.
- Mun, S.H., 2019. *Real time block locating system for shipbuilding*

- through GNSS and IMU fusion. Ph.D. Thesis, Pusan National University.
- Nam, B.W., Lee, K.H., Lee, J.J. & Mun, S.H., 2017. A study on selection of block stockyard applying decision tree learning algorithm. *Journal of the Society of Naval Architects of Korea*, 54(5), pp.421–429.
- Pan, S. J., & Yang, Q., 2009. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10), pp.1345–1359.
- Park, S.W. & Kim, D.Y., 2018. Comparison of image classification performance in convolutional neural network according to transfer learning. *Journal of Korea Multimedia Society*, 21(12), pp.1387–1395.
- Saito, G., 2017. *Deep learning from scratch*, Harbit Media, Inc.
- Shin, J.G. & Lee, J.H., 2006. Prototype of block tracing system for pre-erection area using PDA and GPS. *Journal of the Society of Naval Architects of Korea*, 43(1), pp.87–95.
- Simonyan, K. & Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. *Published as a conference paper at ICLR 2015*, San Diego, United States of America, September 2014.
- Srivastava, N. et al., 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1), pp.1929–1958.
- Standford University, 2018. ILSVRC, URL : <http://image-net.org/challenges/LSVRC/> [Accessed 30 November 2019].
- Szegedy, C. et al., 2014. Going deeper with convolutions. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pp.1–9.
- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. & Wojna, Z., 2016. Rethinking the inception architecture for computer vision. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, Las Vegas, United States of America, pp.2818–2826.
- Yosinski, J., Clune, J., Bengio, Y. & Lipson, H., 2014. How transferable are features in deep neural networks?. *In Advances in neural information processing systems*, pp.3320–3328.
- Zeiler, M.D. & Fergus, R., 2014. Visualizing and understanding convolutional networks. *In European conference on computer vision*, Zurich, Swiss, pp.818–833.
- Zhou, B., Khosla, A., Lapedriza, A., Oliva, A. & Torralba, A., 2016. Learning deep features for discriminative localization. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2921–2929.
- Zhou, B. et al., 2014. Object detectors emerge in deep scene cnns. *In International Conference on Learning Representations*, San Diego, United State of America.
- Zoph, B., Vasudevan, V., Shlens, J. & Le, Q.V., 2018. Learning transferable architectures for scalable image recognition. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2018*, Salt Lake Coty, United States of America, pp.8697–8710.



전해명

노재규