

CNN을 이용한 Al 6061 압출재의 표면 결함 분류 연구

김수빈¹ · 이기안^{2, #}

Study on the Surface Defect Classification of Al 6061 Extruded Material By Using CNN-Based Algorithms

S. B. Kim, K. A. Lee

(Received July 18, 2022 / Revised July 26, 2022 / Accepted July 29, 2022)

Abstract

Convolution Neural Network(CNN) is a class of deep learning algorithms and can be used for image analysis. In particular, it has excellent performance in finding the pattern of images. Therefore, CNN is commonly applied for recognizing, learning and classifying images. In this study, the surface defect classification performance of Al 6061 extruded material using CNN-based algorithms were compared and evaluated. First, the data collection criteria were suggested and a total of 2,024 datasets were prepared. And they were randomly classified into 1,417 learning data and 607 evaluation data. After that, the size and quality of the training data set were improved using data augmentation techniques to increase the performance of deep learning. The CNN-based algorithms used in this study were VGGNet-16, VGGNet-19, ResNet-50 and DenseNet-121. The evaluation of the defect classification performance was made by comparing the accuracy, loss, and learning speed using verification data. The DenseNet-121 algorithm showed better performance than other algorithms with an accuracy of 99.13% and a loss value of 0.037. This was due to the structural characteristics of the DenseNet model, and the information loss was reduced by acquiring information from all previous layers for image identification in this algorithm. Based on the above results, the possibility of machine vision application of CNN-based model for the surface defect classification of Al extruded materials was also discussed.

Keywords: Convolution Neural Network, Surface Defect, Aluminum alloy, Extrusion, Deep Learning, Data Augmentation

1. 서론

알루미늄 합금은 경량화의 핵심 소재로 산업적 활용 분야가 매우 다양하다. 알루미늄계 소재와 관련하여 압출 공정은 주요 제조 공정들 중 하나이며, 압출 공정 제어를 통해 구조적 형상 제어 및 심미적 기능 부가도 가능하다 [1]. 최근 경량화를 통한 탄소 배출량 감소 이슈가 부각됨에 따라, 알루미늄계 압출재를 항공기, 자동차, 선박, 고속철도 부품[2] 등에 적용하려는 시도가 더욱 확대되고 있다.

압출재 부품에서는 Blister, Die-Line, Pick-Up, Weld Line, Crack, Back-defect 등의 표면 결함들이 나타날 수 있다 [3]. 본 연구에서는 이들 중에서 Blister, Crack, Back-defect 의 세 가지 대표적 표면 결함에 초점을 맞추어 진행하고자 한다. Blister 는 갇힌 공기 또는 휘발성 윤활제로 인해 발생하는 결함이다. 이것은 압출 시 빌렛과 컨테이너의 직경 차로 인해 공정 중에 일정량의 공기가 컨테이너에 갇히게 되고, 압축 시 Die 를 통해 나가면서 팽창하여 물집 형상으로 나타난다. Crack 은 압출 중 온도 상승과 밀접한 관련이 있는 것으로, Die 부근에서 발생하는 마찰 열이 고상점 이상의 국부적 온도 상승을 가져오고 용융을 발생시켜 생성될 수 있다 [4]. Back-defect 는

1. 인하대학교 신소재공학과, 대학원 박사과정

2. 인하대학교 신소재공학과, 교수

Corresponding Author : Department of Materials Science & Engineering, Inha university, E-mail: keeahn@inha.ac.kr

ORCID : 0000-0003-2149-3871

이송 불량으로 압출 제품을 취급하는 동안 흡집으로 인하여 발생하게 되며, 표면에서 일정 깊이의 선으로 관찰된다.

압출 과정에서 발생한 결함들은 제품의 물성 및 후공정시 품질에 해로운 영향을 미칠 수 있다. 또한, 원재료의 낭비를 증가시키고 제품 수명을 감소시켜 기업의 손실을 야기시킨다 [5]. 따라서 압출재 부품에서 불량제품을 분류하여 제품의 품질을 향상하는 품질 관리 공정은 필수적으로 요구된다. 기존의 제품 품질 관리를 위한 검사에는 많은 인력 자원을 필요로 한다 [6]. 이를 위해서 그 동안 검출자의 육안 검사를 바탕으로 주로 검사 진행해 왔으며, 그 결과 개인의 경험적, 감정적 상태에 따라 검사 결과의 변동성으로 객관적 판단에 어려움이 있었다 [7]. 또한 인력 기반의 검사는 오랜 시간이 소요되며 산업 생산 속도에 맞추기 어려운 단점도 존재한다. 이러한 문제의 해결은 스마트 팩토리(Smart Factory)의 한 분야인 머신 비전(Machine Vision)을 통해 이뤄질 수 있다.

머신 비전은 기계에 사람이 가지고 있는 시각과 판단 기능을 부여한 것으로 최근 향상된 성능을 기반으로 교통, 의료, 식품, 군사, 제조업 등으로 그 활용 범위가 더 넓어지고 있다 [8~10]. 제조업에서의 머신 비전을 이용할 경우, 생산의 통합 관리를 통해 불량을 사전에 검출하고 이에 대한 원인을 해결해 생산성을 향상시킬 수 있다. 관련하여 머신 비전을 이용한 결함 검출 기술 연구와 이를 활용한 생산 효율성과 생산 비용 절감, 제품 품질 향상 연구가 수행되고 있다 [11]. 최근 머신 비전에 적용하는 검출 기법 중 물체 인식에 탁월한 성능을 보일 수 있는 딥러닝 기법을 결함 검사에 적용하는 연구[12~17]들이 급격히 늘어나고 있다. 특히, 컴퓨터 비전 분야에서 뛰어난 성능을 나타내는 Convolutional Neural Network(CNN)[18]는 비전 검사에서 핵심적인 알고리즘으로 손꼽힌다. CNN은 1989년 LeCun[19]에 의해 처음 소개되었으며 필기체 인식에 있어 유의미한 결과를 도출하였으나 이를 범용화하기에는 부족한 수준이었다. 그 후 1998년 LeNet[20]이라는 Pooling, Padding, Activation, Fully Connected 로 연계되는 일련의 과정들이 적용되어, MNIST Dataset 을 기준으로 99.05% 정확도가 달성된 바 있다. 2003년 Behnke [21], Simard [22] 등에 의해 더욱 단순한 구조가 만들어졌으며, 이를 바탕으로 CNN 분야에 더 활발한 연

구가 촉발되었다. 근래에는 CNN 알고리즘을 기반으로 AlexNet[23], VGGNet[24], GoogLeNet[25], ResNet[26], DenseNet[27] 등 다양한 알고리즘들이 파생되어 제시되었으며 이후 현재까지 끊임없이 새로운 알고리즘들이 제안되고 있다.

CNN 기반의 알고리즘을 이용한 이미지 분류 성능에 대한 연구는 다양한 산업군에 걸쳐 이뤄지고 있다. Shoaib Karrem 등[12]은 유방암에서 유관 암종을 분류하기 위한 딥러닝을 진행했으며, 정확도, 정밀도, 재현율 등의 점수를 기반으로 DenseNet121, MobileNet V2, ResNet50 및 VGGNet 16 등의 CNN 모델들을 비교하고, DenseNet121 모델에서 89.25% 정확도가 제시되었다. Xie 등[13]은 DenseNet121, ResNet50, Inception V3, VGG16, VGG19의 5개 CNN 기법들을 이용하여 외관상 불량 당근을 인식하기 위한 딥러닝을 진행하였다. Tao 등[14]은 금속 표면 결함 데이터를 Gray Scale 변환을 통해 5개의 Convolutional Layer와 3개의 Max-Pooling Layer를 사용하여 시도된 CNN으로 결함을 분류하여 89.60%의 정확도를 보였다. 또한 Feng 등[17]은 열간 압연 강판 스트립 표면 결함 감지를 위해 X-SDD(XSteel Surface Defect Dataset)라는 열연 강판 표면의 결함 데이터에 대해 RepVGG 및 VGG, ResNet, SqueezeNet, EspNet 등의 다양한 알고리즘을 적용하여 그 성능을 비교·분석하였다. Wei 등[28]은 Multi-Scale Object Detection을 사용하여 알루미늄 프로파일 표면 결함 감지 및 분류를 진행하였다. 여기에서 10가지 유형의 결함을 포함하여 총 3,005개의 이미지 데이터가 수집되었으며, 결과적으로 여러 유형에 대한 AP(Average Precision)는 75.8%를 나타냈다. 그러나 현재까지 비전 분야에서 알루미늄 압출재에 대한 연구는 Profile 표면 결함에 대한 연구를 제외하고는 거의 전무한 실정이다. 특히 철강 재료와 달리 알루미늄 합금의 경우에는 균형 잡힌 대규모의 데이터 세트가 구축되어 있지 않아 연구를 수행함에 어려움이 존재한다.

본 연구에서는 CNN 기반의 알고리즘들을 이용하여 Al 6061 압출재의 소성 가공 중 발생하는 Back-defect, Blister, Crack 등의 결함들을 분류하고자 하였다. 이를 위해 데이터 수집 기준을 마련하고 확보하였으며, 얻어진 데이터의 불균형 및 작은 훈련 데이터 세트의 보완을 위해 데이터 증식 기법을 사용하였다. 이후 CNN 기반 VGGNet, ResNet, DenseNet 알고리즘들을 이용하여 최적의 조건을 도출하고 그

성능을 비교하였다.

2. Deep Learning Model

2.1 합성곱 신경망(Convolutional Neural Network, CNN)

CNN은 DNN(Deep Neural Network)에서 다차원 데이터를 처리함에 발생하는 문제점으로부터 출발하였다. DNN은 기본적으로 1D 형태의 데이터를 사용하며, Fig 1과 같이 2D 형태의 이미지가 입력될 때에는 이를 Flatten 시켜 벡터로 평면화해야 한다. 이 과정에서 이미지에 대한 많은 정보를 잃어버리게 되며, 특히 데이터 내의 공간적, 지역적 정보가 손실된다. 또한 추상화 과정 없이 바로 연산 과정으로 넘어가기 때문에 그 효율성이 저하될 수 있다.

이러한 문제점을 해결하기 위해 CNN 이 제시되었으며, CNN 에서는 이미지를 그대로 받아들임으로써 공간적, 지역적 정보를 유지한 채 특성들의 계층을 쌓아 올리게 된다. 기존의 DNN 에서는 Affine 으로 명시된 Fully-connected 연산과 ReLU 와 같은 비선형 활성화 함수의 합성으로 정의된 계층을 여러 층 쌓아 사용하였지만, CNN 에서는 Fig 2 와 같이 Convolution Layer 와 Pooling Layer 를 Fully-connected Layer 이전에 추가함으로써 이미지 데이터에 필터링 기법을 적용한 뒤, 필터링 된 이미지에 대해 분류 연산을 수행한다. 즉 Convolution Layer 에서 이미지에 필터링 기법을 적용하고, Pooling Layer 에서 이미지의 국소적인 부분들을 하나의 대표적 스칼라 값으로 변환하여 이미지의 크기를 줄이는 기능을 수행한다.

일반적으로 이미지는 H(Height) x W(Width) x C(Channel)의 3D Tensor 로 표현될 수 있다. Color Image 의 경우 각 픽셀은 RGB(Red, Green, Blue)로 구성되어 3 개의 Channel 을 갖는다. 한 개의 Convolution Layer 에는 입력되는 이미지의 채널 개수 만큼 필터가 존재하며, 각 채널에 할당된 필터를 적용함으로써 Convolution Layer 의 출력 이미지가 생성된다. Convolution Layer 의 작동 원리에 대해 간략한 예시를 위해 1 개의 채널을 가진 흑백 이미지를 이용하여 설명하였다. (H, W, C)가 5x5x1 인 Tensor 형태의 입력 이미지에 대해 3x3 필터를 적용하는 Convolution Layer 에서는 Fig 3(a)와 같이 3x3x1 Tensor 형태의 이미지가 생성된다. 하나의 필터를 이용하여

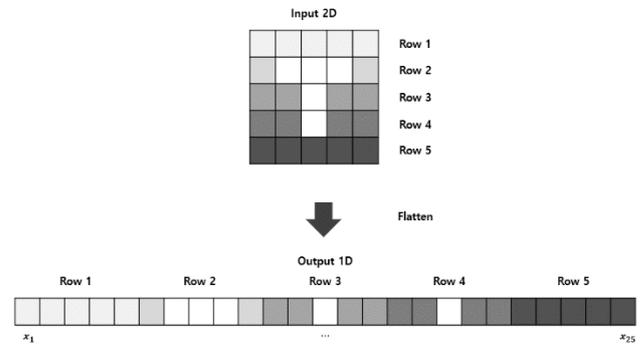


Fig. 1 Flatten works converting matrix to single array

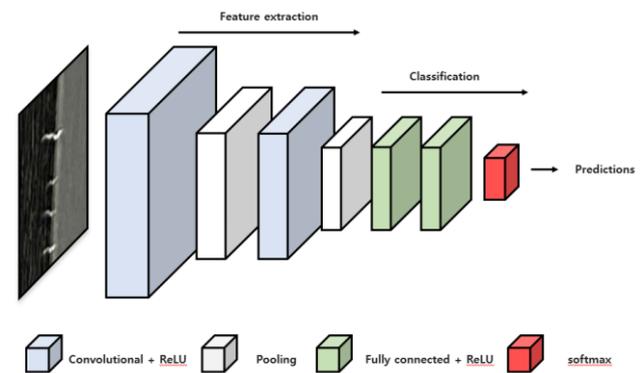


Fig. 2 Schematic diagram of convolutional neural network (CNN)

입력 값 전체를 포괄적으로 연산 처리를 함으로써 패턴을 찾아내는 것이 Convolution Layer 의 목적이다. Convolution 연산을 수행할 때는 이러한 이미지의 축소를 조절하기 위해 추가적으로 Padding 과 Stride 를 적용하기도 한다 [29]. 위 예시에서 5x5 Image 가 3x3 으로 축소됨을 확인할 수 있다. 이는 Image Data 의 손실이 발생함을 의미하며, 이러한 점을 해결하기 위해 Padding 이라는 방법이 사용될 수 있다. 이는 원래 이미지의 양옆에 0 또는 1 값을 갖는 Data 를 추가함으로써 필터가 적용된 후에도 원래의 크기와 동일한 이미지를 유지할 수 있다. 즉, 이미지의 가장 자리에 0 또는 1 로 구성된 테두리를 감싸주는 방식이다. Fig 3(b)를 보면 5x5 Image 는 Padding 을 통해 7x7 Image 가 된 것을 알 수 있으며, 이 상태에서 3x3 필터를 적용해줄 경우, 입력과 동일한 5x5 크기의 출력 값을 얻을 수 있다. 또한 Stride 를 적용하여 필터의 보폭을 조정해줄 수 있다. 이처럼 Convolution Layer 는 이미지를 분류하는데 필요한 특징 정보를 추출해내는 역할을 한다.

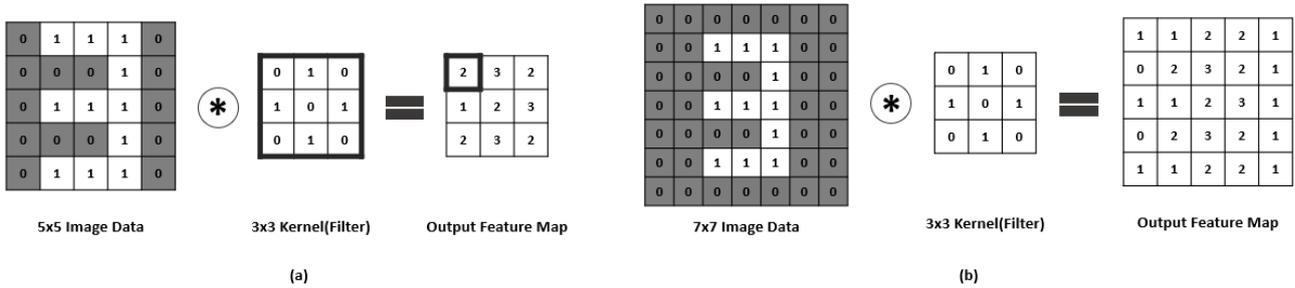


Fig. 3 Schematic diagram of the convolution operation; (a) a kernel size of 3 x 3 and (b) zero-padding

Pooling Layer에서는 Convolution Filter를 거친 결과를 Input으로 받아들인다. Convolution Layer에서 각 필터당 하나의 Feature Map이 형성되고 이 Map을 쌓아 두는데, 이는 필터의 개수가 많아지는 만큼 CNN의 차원이 매우 커짐을 의미한다. 고차원을 구현하기 위해서는 이에 상응하는 더 많은 파라미터가 요구되고 이는 과적합(Over-Fitting)을 초래할 수 있다. 따라서 Mapping된 공간 정보를 압축하는 방법이 필요한데 이러한 역할을 하는 것이 Pooling Layer이다. Pooling Layer는 Max-Pooling과 Average-Pooling 등의 방식이 있으며, Fig 4에 관련된 각 Pooling Layer의 역할을 도시하였다.

2.2 VGGNet

VGGNet[24]은 Karen Simonyan 과 Andrew Zisserman 이 만든 CNN 모델로 작은 필터로 네트워크층을 깊게 쌓아 네트워크 깊이가 모델의 높은 성능을 보이

는 데 중요한 역할을 할 수 있음을 보여주었다. 이 모델은 3x3 Filter, Stride 1, Zero-Padding 을 사용한 Convolution Layer 로 이루어져 있으며, Pooling Layer 는 2x2 크기의 Max Pooling 을 사용하여, Stride 2 로 설정된다. 입력 크기는 224x224의 컬러 영상을 사용하였으며, 하나 또는 복수의 Convolution 및 Max-Pooling Layer 가 반복되는 구조이고 마지막으로 Fully-Connected Layer 로 구성되어 있다.

2.3 ResNet

앞선 VGGNet CNN 모델을 이용한 연구 결과들에서 네트워크 층이 깊어질수록 성능 향상에 긍정적인 영향을 나타냈다. 그러나 여기에서 단순히 네트워크 층을 깊게 구성할 때, 일정 깊이에 이르면 기울기 소실(Gradient Vanishing) 및 증폭(Exploding) 현상으로 학습 속도가 느려지거나 학습이 제대로 되지 않는 딥러닝 성능의 한계점이 나타났다 [30]. ResNet 은 이러한 한계점을 극복하고자 제시되었으며, 잔차 학습(Residual learning) 방식을 도입하여 가중치가 소실되더라도 이전 층의 가중치로 다시 전환이 가능한 빌딩 블록을 만들어 사용한다.

Fig 5 는 일반적인 네트워크와 잔차 블록 방식에 대한 비교 모식도이다. (a)는 일반적인 네트워크를 나타내고 (b)는 잔차 블록 구조를 나타낸다. 잔차 블록 방식에서는 기존의 네트워크와 달리 입력 값을 출력 값에 더해주는 Short-Cut 연결이 존재한다. Short-Cut 연결은 잔차 블록의 입력 x 가 출력에 연결되므로 파라미터의 수에 영향이 없으며 덧셈을 제외하면 연산량의 영향도 없다. 또한 이 경우 SGD 나 오차 역전파에 의한 학습도 가능하다 [31].

ResNet 에서는 Layer 가 깊어질수록 학습에 걸리는 시간이 증가하고 파라미터의 수가 증가하는 문제를 해결하고자, 50, 101, 152 층의 네트워크에 대해 Fig 6

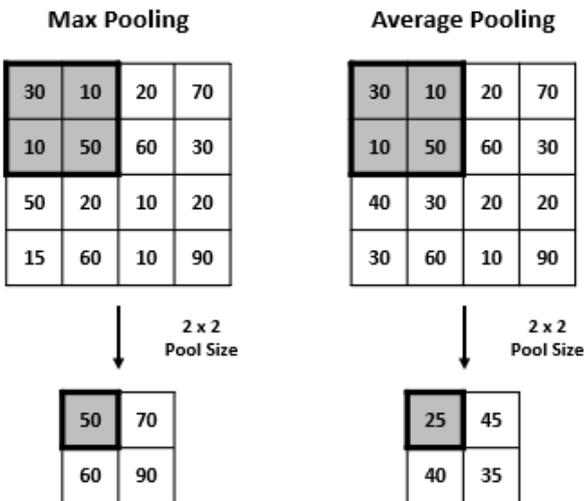


Fig. 4 Schematic diagram of pooling; left is max-pooling and right is average-pooling

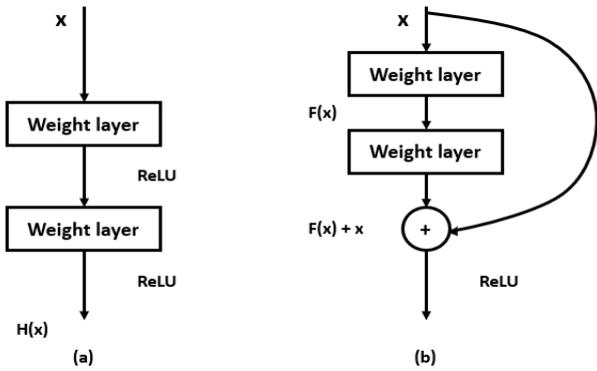


Fig. 5 (a) Plain network (b) Residual Block of ResNet

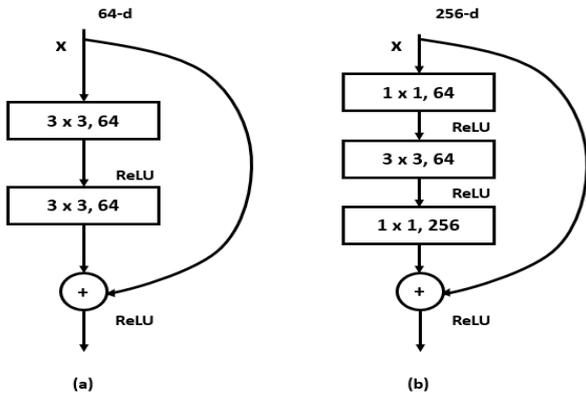


Fig. 6 Configuration of ResNet; (a) residual Block and (b) bottleneck

(a)의 잔차 블록 내 Convolution Layer 2 개를 (b)와 같이 1x1, 3x3, 1x1 의 3 개의 Layer 로 변형하였다. 여기에서 맨 처음 1x1 Convolution Layer 는 차원을 감소시키고 마지막 Layer 는 차원을 증가시키는 역할을 한다. 이 구조가 병목처럼 보여 일명 병목 구조라 부르며, 상기 과정을 통해 연산량을 절감시킬 수 있다. 즉, ResNet 은 잔차 블록과 병목구조를 도입해 네트워크를 구성하고 이러한 변화를 통해 네트워크 구조가 깊게 형성되었음에도 기울기 소실 문제의 해결과 빠른 속도의 이미지 분류를 구현할 수 있다.

2.4 DenseNet

DenseNet은 ResNet 모델을 부분 수용하여 개선한 모델로 ResNet보다 더 적은 파라미터 수를 가지지만 더 높은 성능을 나타낼 수 있도록 제시된 모델이다. ResNet에서는 이전 Layer에서 추출된 특징을 다음 계층에서 더해주는 방식을 통해서 거시적인 초기의

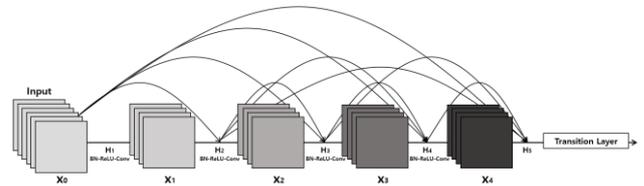


Fig. 7 Schematic diagram of Dense Block

정보를 잃지 않지만, 이 또한 직전 계층의 정보만을 받아오기 때문에 일부 특징 손실이 일어날 수 있다는 점이 문제될 수 있다 [32]. DenseNet의 경우 Dense Block 개념을 도입하여 이러한 문제를 해결하고자 하였다. Fig 7에 도시한 바와 같이 DenseNet 에서는 Dense Block내의 모든 레이어 간에 직접적 연결이 수행된다. 이러한 구조는 추출된 정보를 누적하는 전달 방식을 통해 이전 계층에서 추가된 입력에 대해 후속 계층으로 전달이 가능하다 [32]. 즉 깊은 모델의 후반 Layer까지 앞선 정보를 효율적으로 전달할 수 있도록 설계되었으며 이에 따라 더욱 정확한 분류가 가능해진다.

3. Experimental Method

3.1 데이터 수집

실제 알루미늄 결함 데이터의 경우 산업 분야의 제약 사항과 보안상의 문제로 인해 대규모의 데이터 셋을 확보하는 데 어려움이 있다. 또한 인터넷을 통해 데이터를 수집하는 경우에는 연구 목적에 맞는 데이터를 구하기 어렵다. 이에 본 연구에서는 AI 6061 합금의 압출 공정에서 발생하는 결함재를 직접 수집하여 초기 데이터 셋을 구축하였다.

알루미늄 압출재의 표면은 매끄럽고 높은 표면 반사율을 가진다. 이러한 특징으로 인해 빛과 그림자 영역을 가져와 회색 값이 일치하지 않아 잘못된 가장자리 감지 가능성이 높아진다 [33]. 촬영되는 알루미늄 압출재의 결함 데이터는 배경, 그림자 등 다양한 측면에서 분류 성능을 떨어뜨릴 수 있는 요소들을 가지고 있다. 그러므로 분류 정확도의 저해 요소를 제거하기 위해 다음과 같은 수집 기준을 마련하여 데이터 셋을 구성하였다. 먼저 알루미늄 압출재를 촬영함에 있어 동일한 위치 및 시간대에 압출해 촬영을 진행하였다. 이러한 방법은 다른 배경적 원인 요소를 제거하기 위한 것이다. 대부분의 산

업 공정은 공장의 내부에서 진행되기 때문에 외부 빛에 의한 영향이 거의 없다. 실제 데이터를 구성할 때도 이를 최대한 반영하여 외부 환경 조건의 영향을 줄이고자 하였다. 이와 함께 결함 사진 촬영자를 3명 이상으로 구성하여 각기 다른 카메라를 이용하여 기준에 맞게 촬영함으로써 더욱 객관적인 데이터 수집을 진행하고자 하였다.

본 연구에서 사용하게 될 결함 이미지를 준비하기 위해 알루미늄 압출재 생산에 있어 불량인 난제품들을 모아 각 결함을 촬영하였다. Fig 8 과 같은 장의 사진에는 다수의 결함이 포함되어 있으며, 먼저 230×230 크기로 결함 후보 영역을 찾아 분할하였다. 얻어진 이미지들에서 Fig 9 에 제시된 바와 같이 총 3 가지 주요 표면 결함들(Back Defect, Blister, Crack)이 제시되었다. 이를 기반으로 학습 데이터 1,417 개, 평가 데이터 607 개로 무작위 분류하였으며 총 2,024 개의 이미지로 데이터 셋을 구축하였다.

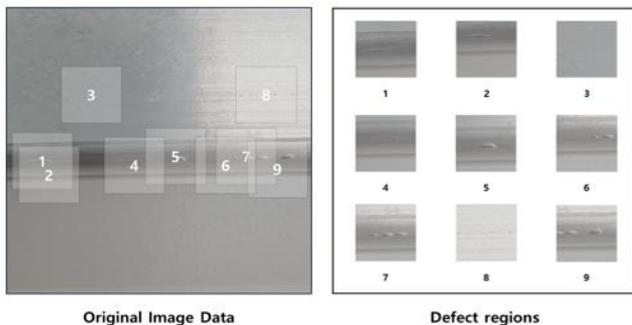


Fig. 8 Image defect area extraction procedure

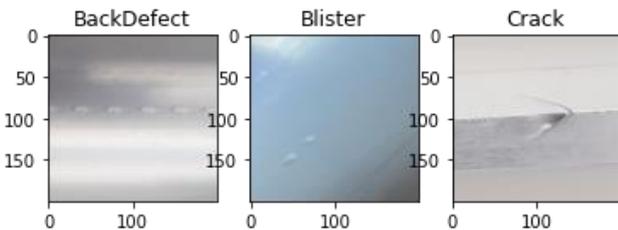


Fig. 9 Typical surface defects of aluminum 6061 extrusion products

3.2 이미지 증식(Image Augmentation)

딥러닝 학습에 필요한 데이터를 수집하기에 있어 산업 특성상 데이터가 부족하거나 각 레이블간 데

이터 불균형이 발생할 가능성이 매우 높다. 또한, 작은 데이터 세트를 이용하여 CNN 을 훈련하게 되면 과적합이 발생해 학습 성능이 크게 저하된다. 이와 같이 작은 데이터로부터 발생하는 문제점은 데이터 세트를 인위적으로 부풀리는 이미지 증식 기법을 사용하여 해결할 수 있다 [34,35].

본 연구에서는 그림과 같이 Rotation, Random Crop, Random Horizontal Flip 과 같은 Geometric 방법들과 Color Jitter를 사용한 Photometric 방법을 이용하여 데이터를 증식하였다. Geometric 방법은 이미지 내에서 표현되는 클래스의 기본 모양이 유지되지만 새로운 위치와 방향으로 변경시켜 이미지를 새롭게 만들어 낼 수 있다. Photometric 방법은 각 픽셀 값을 이동하여 RGB 채널을 변경하는 방식으로 이미지의 명도(Brightness), 색상(Hue), 채도(Saturation) 그리고 대비(Contrast) 등을 임의 변경하는 방식이다. 본 연구에는 Brightness 0.2, Hue 0.2, Saturation 0.2, Contrast 0.2 로 지정하여 무작위 생성하도록 수행하였다. 위 과정들을 통해 만들어진 결함 이미지들은 결함이 포함되지 않거나 결함이 잘려 극히 일부만 포함된 이미지를 생성하기도 한다. 이러한 이미지는 수작업을 통해 학습 데이터에서 제외하였다. 앞서 수행된 이미지 증식에서 적절한 파라미터의 선택은 이러한 수작업의 번거로움을 최대한 줄여줄 수 있었다.

3.3 CNN Network 및 학습 환경

학습에 사용한 Network 구조는 VGGNet-16, VGGNet-19, ResNet-50, DenseNet-121 으로 평가 데이터를 이용해 Network 각각의 성능을 비교, 분석하였다. 각 Network 에 대한 구성은 Table 1,2,3 에 자세히 제시하였다.

알루미늄 압출재 결함 분류 실험 진행을 위해 GPU는 NVIDIA Quadro RTX5000 을 사용하였으며, 딥러닝 프레임워크로 Pytorch 를 이용하여 학습을 진행하였다. 이미지 분류 실험은 계산할 요소가 많아 다량의 계산에 적합한 GPU 를 사용한다 [36]. Table 4 에 실험 환경에 대한 하드웨어 및 소프트웨어 성능을 자세히 나타냈다.

학습에 사용된 하이퍼 파라미터에 대하여 Table 5 에 정리 및 제시하였다. 하이퍼 파라미터는 딥러닝 성능에 큰 영향을 미치는 요소들 중 하나이다. 따라서 각 모델 구조에 맞는 파라미터들의 적절한 조합과 적용이 요구된다. 본 연구에서는 Batch Size,

Epoch 등의 파라미터를 다양하게 바꾸어 학습을 수행하였으며, 그 성능을 비교하였다. Learning rate 의 경우 일반적으로 임의 설정을 진행하게 되는데, 본 연구에서는 Learning rate Scheduler 를 사용하여 처음에는 큰 보폭으로 빠르게 Optimize 하고 최적 값에 가까워질수록 보폭을 줄여 미세 조정하였다.

Table 1 Diagram of VGGNet Structure

Layer	Output Size	16-layer	19-layer
Input(224 x 224 RGB Image)			
Conv	64	$\begin{bmatrix} 3 \times 3 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix}$	$\begin{bmatrix} 3 \times 3 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix}$
Pool		MaxPool	
Conv	128	$\begin{bmatrix} 3 \times 3 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix}$	$\begin{bmatrix} 3 \times 3 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix}$
Pool		MaxPool	
Conv	256	$\begin{bmatrix} 3 \times 3 \text{ conv} \\ 3 \times 3 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix}$	$\begin{bmatrix} 3 \times 3 \text{ conv} \\ 3 \times 3 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix}$
Pool		MaxPool	
Conv	512	$\begin{bmatrix} 3 \times 3 \text{ conv} \\ 3 \times 3 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix}$	$\begin{bmatrix} 3 \times 3 \text{ conv} \\ 3 \times 3 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix}$
Pool		MaxPool	
Conv	512	$\begin{bmatrix} 3 \times 3 \text{ conv} \\ 3 \times 3 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix}$	$\begin{bmatrix} 3 \times 3 \text{ conv} \\ 3 \times 3 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix}$
Pool		MaxPool	
FC	4096	-	-
FC	4096	-	-
FC	1000	-	-
softmax			

Table 2 Diagram of ResNet Structure

Layer	Output Size	50-layer
Conv 1	112 x 112	7 x 7, 64, Stride 2
Conv 2_x	56 x 56	3 x 3, MaxPool, Stride 2
		$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 253 \end{bmatrix} \times 3$
Conv 3_x	28 x 28	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$
Conv 4_x	14 x 14	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$
Conv 5_x	7 x 7	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1 x 1	Average Pool, 1000-d fc, softmax
FLOPs		3.8 x 10 ⁹

4. 결과 및 고찰

본 연구에 적용된 알고리즘의 분류 성능을 평가 지표로 얻어진 결과들 중 정확도와 손실 값들을 비교, 평가하였다. 정확도는 전체 데이터에 대한 예측 정답 비율이며, 손실은 실제 정답과 모델이 예측한 값의 사이의 거리(또는 오차)를 의미한다. 즉, 이 값들을 기반으로 잘못 예측한 경우 얼마나 오류를 범했는가를 정량화 할 수 있다. 각 Network 별 학습 결과들을 Fig 10 에 제시하였으며 여기에서 각 모델의 정확도 및 손실 값을 도출, 비교하였다.

Table 3 Diagram of DenseNet Structure

Layer	Output Size	121-layer
Conv	112 x 112	7 x 7 Conv, Stride 2
Conv	56 x 56	3 x 3, MaxPool, Stride 2
Dense Block (1)	56 x 56	$\begin{bmatrix} 1 \times 3 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
Transition (1)	56 x 56	1 x 1 Conv
	28 x 28	2 x 2, AveragePool, Stride 2
Dense Block (2)	28 x 28	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
Transition (2)	28 x 28	1 x 1 Conv
	14 x 14	2 x 2, AveragePool, Stride 2
Dense Block (3)	14 x 14	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$
Transition (3)	14 x 14	1 x 1 Conv
	7 x 7	2 x 2, AveragePool, Stride 2
Dense Block (4)	7 x 7	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 16$
Classification	1 x 1	7 x 7, Global Average Pool
		10000 fully-connected, softmax

Table 4 The conditions of experimental environment

OS	Windows 10, 64 bit	
CPU	Intel(R) Xeon(R) W-1250P@4.10 GHz	
RAM	32GB	
GPU	Name	NVIDIA Quadro RTX5000
	Core clock rate	1620 MHz
	Memory size	16384 MB
	Memory clock rate	1750 MHz
Toolkit	Cuda	10.0.130
	Cudnn	7.6.5
	Pytorch	1.7.1

Table 5 Network Training Hyper-parameters

	VGG-16	VGG-19	ResNet-50	DenseNet-121
Batch Number Size	16	16	16	16
Input Image Size	224 x 224	224 x 224	224 x 224	224 x 224
Number of Epochs	40	40	100	100
Learning rate	Learning rate scheduler			

4.1 VGGNet

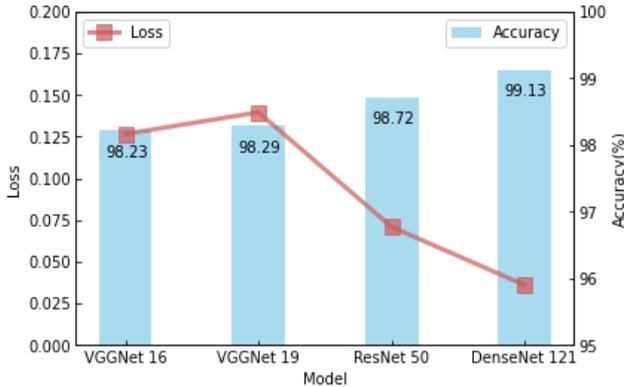


Fig. 10 Data classification rate results obtained in this study

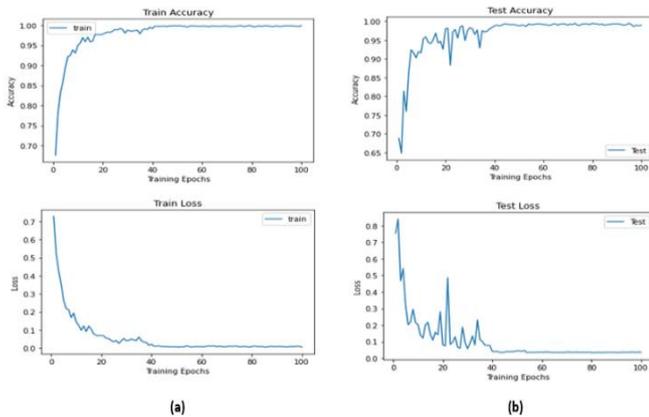


Fig. 11 Training result for DenseNet architectures; (a) training-set accuracy and loss and (b) test-set accuracy and loss

본 연구에서는 224×224 의 입력 이미지를 사용하여 VGGNet-16, 19를 이용한 학습을 진행하였다. 초기 VGGNet 학습에 있어 100회의 반복 학습을 설정하였으나, 45이상의 학습 이후 손실 값이 증가하는 흐름을 보였고 이는 학습의 최적점을 지나 과적합이 일어났음을 나타낸다. 이에 학습 횟수를 40으로 줄여 기존 현상에 대한 문제점을 해결할 수 있도록 하였고, 줄어든 학습 횟수에도 충분한 학습이 이뤄짐을 확인할 수 있었다.

Batch Size에 있어 64, 32, 16으로 비교 학습을 수행한 결과 64, 32 Batch Size에서는 매우 낮은 정확도가 나타났다. 일반적으로 Batch Size를 크게 하는 경우, 학습 도중 오차의 국부 최적에 의해 더 이상의 학

습이 진행되지 않는다. 반면 Batch Size를 작게 하는 경우 Noise로 인해 모델의 불안정한 수렴이 유발될 수 있는데, 이 때 약간의 Noise는 과적합을 방지하여 모델의 성능을 향상시키는 요인이 될 수도 있다. 그렇지만 Batch Size를 너무 작게 하는 경우에는 Noise가 더욱 커져 성능에 악영향을 미칠 수 있으므로 적절한 Batch size의 선택이 요구된다.

결과적으로 본 연구에서는 Batch size 16, Epoch 40, Learning rate Scheduler를 적용하여 최종 학습이 진행되었다. VGGNet-16의 경우 테스트 데이터에 의한 정확도는 98.23%로 나타났으며, 학습에 소요된 총 시간은 72.51 min으로 1 epoch 당 평균 1.81 min으로 나타났다. VGGNet-19의 경우 Accuracy는 98.29%를 보였으며, 학습에 소요된 시간은 67.83 min으로 1 epoch 당 평균 1.70 min으로 제시되었다.

4.2 ResNet

224×224 의 입력 Image를 사용하여 ResNet-50 학습을 진행하였으며, 그 결과 ResNet-50 알고리즘에 대한 분류 성능의 정확도는 98.72%로 나타났다. 학습에 소요된 총 시간은 50.80 min으로 1 epoch 당 평균 0.51 min이 소요되었다. 즉 ResNet은 VGGNet보다 더 향상된 정확도 값을 나타내는 것을 알 수 있었다. 이는 잔차 학습을 통해 이전의 가중치를 가져와 데이터 특징 손실을 줄인 결과로 이해될 수 있다.

4.3 DenseNet

DenseNet-121 알고리즘을 이용한 학습 진행 결과, 99.13%의 정확도를 보였으며 학습에 소요된 총 시간은 30.61 min으로 1 epoch 당 평균 0.306 min으로 나타났다. Fig 11에서 DenseNet-121의 평가 지표의 변화 수치를 확인할 수 있다. DenseNet의 경우 앞서 제시된 VGGNet, ResNet 알고리즘들과 비교할 때, 가장 빠른 학습 속도 성능을 나타냈으며, 손실 또한 가장 낮게 측정되었다. 또한 DenseNet 알고리즘을 사용할 때 가장 높은 정확도를 나타내어, 결함 이미지를 분류하는데 있어 최적의 Architecture라 판단되었다. DenseNet에서 얻어진 이러한 향상된 결과는 Dense Block을 통해 앞선 특징 값을 누적적으로 받아들여와 특징 값의 손실을 보상하였기 때문으로 설명될 수 있다.

5. 결론

본 연구에서는 직접 구축한 압출 결함 데이터 세트를 이용하여 VGGNet, ResNet, DenseNet 모델의 성능을 비교하였다. 연구의 결과는 다음과 같다.

(1) Al 6061 압출재에서 나타난 실제 결함을 기반으로 가시적 데이터 세트를 직접 구성하였다. 먼저 분류 정확도의 저해 요소를 제거하기 위해 적정 수집 기준을 마련하여 이에 따라 데이터 셋을 구성하였다. 얻어진 이미지들을 3 가지 주요 표면 결함들 (Back Defect, Blister, Crack)로 분류하였으며, 총 2,024 개의 결함 이미지를 구축하여 사용하였다.

(2) 본 연구에서는 Geometric 방법과 Photometric 방법을 이용하여 이미지 증식 과정을 통해 학습에 필요한 적절한 크기의 데이터 셋을 구축하였으며, 이를 통해 학습 데이터의 패턴을 도출함으로써 검 증 데이터의 정확도를 높일 수 있었다.

(3) DenseNet-121 네트워크가 99.13 %의 정확도와 0.037의 loss로 가장 우수한 분류 성능을 보임을 확인하였다. 한편 VGGNet-16의 경우 98.23 %의 정확도를, ResNet-50의 경우는 98.72 %의 정확도를 나타냈다. 학습 속도 또한 DenseNet에서 1epoch 당 평균 0.306 min으로 제시되어 3 가지(VGGNet-16 (1.81 min/epoch), VGGNet-19 (1.70 min/epoch), ResNet (0.51 min/epoch)) CNN 기반 알고리즘들 중 가장 빠른 학습 시간을 나타냈다.

REFERENCES

- [1] S. Barella, A. Gruttadauria, R. Gerosa, G. Mainetti, T. Mainetti, 2021, Predictive Tools for in-Line Isothermal Extrusion of 6xxx Aluminum Alloys, *Mater. Proc.*, Vol. 3, No. 1, pp.24-31.
<https://doi.org/10.3390/IEC2M-09239>
- [2] W. Zhou, Z. Shao, J. Yu, J. Lin, 2021, Advances and Trends in Forming Curved Extrusion Profiles, *Mater.*, Vol. 14, No.7, pp. 1603-1631.
<https://doi.org/10.3390/ma14071603>
- [3] A. Chondronasios, I. Popov, I. Jordanov, 2016, Feature selection for surface defect classification of extruded aluminum profiles, *JAMT*, Vol. 83, No.1, pp. 33-41.
<https://doi.org/10.1007/s00170-015-7514-3>
- [4] Z. Peng, T. Sheppard, 2014, Study of surface cracking during extrusion of aluminium alloy AA 2014, *Mater. Sci. Technol.*, Vol. 20, No. 9, pp. 1179-1191.
DOI: 10.1179/026708304225022016
- [5] J. Yang, S. Li, Z. Wang, H. Dong, J. Wang, S. Tang, 2020, Using Deep Learning to Detect Defects in Manufacturing: A Comprehensive Survey and Current Challenges, *Mater.*, Vol. 13, No. 24, pp. 5755-5777.
<https://doi.org/10.3390/ma13245755>
- [6] T. Wang, Y. Chen, M. Qiao, 2018, A fast and robust convolutional neural network-based defect detection model in product quality control, *JAMT*, Vol. 94, No. 9, pp. 3465-3471.
<https://doi.org/10.1007/s00170-017-0882-0>
- [7] M. K. Lee, K. S. Seo, 2018, Comparison of Region-based CNN Methods for Defects Detection on Metal Surface, *KIEE*, Vol. 67, No. 7, pp. 865-870.
<https://doi.org/10.5370/KIEE.2018.67.7.865>
- [8] J. Zhu, F. Shi, J. Li, 2021, IMCEC, IEEE, Manhattan, pp. 2026-2030.
DOI: 10.1109/IMCEC51613.2021.9482067
- [9] J. Uma, B. Sukanya, 2021, ICETEST, Elsevier, Amsterdam, pp. 1-5.
<http://dx.doi.org/10.2139/ssrn.3972999>
- [10] S. Qadri, T. Aslam, S. A. Nawaz, N. Saher, R. Abdur, M. U. Rehman, N. Ahmad, F. Shahzad, S. F. Qadri, 2021, Machine Vision Approach for Classification of Rice Varieties Using Texture Features, *IJFP*, Vol. 24, No. 1, pp. 1615-1630.
DOI: 10.1080/10942912.2021.1986523
- [11] J. P. Park, 2017, Analysis on Success Cases of Smart Factory in Korea: Leveraging from Large, Medium, and Small Size Enterprises, *J. Digit. Converg.*, Vol. 15, No. 5, pp. 107-115.
<https://doi.org/10.14400/JDC.2017.15.5.107>
- [12] S. K. Muhammad, Z. Muhammad, A. K. Hamid, H. J. Fawwad, 2022, Detection of Ductal Carcinoma in Breasts from DDSM Data using DenseNet-121 and Comparative Analysis, *EAI Endorsed Trans. Energy Web*, Vol. 1, No. 1, pp. 24-33.
DOI: 10.4108/EW9832.4242

- [13] W. Xie, S. Wei, Z. Zheng, 2021, Recognition of Defective Carrots Based on Deep Learning and Transfer Learning, *FABT*, Vol. 14, No. 7, pp. 1361–1374.
<https://doi.org/10.1007/s11947-021-02653-8>
- [14] X. Tao, D. Zhang, W. Ma, X. Liu, D. Xu, 2018, Automatic metallic surface defect detection and recognition with convolutional neural networks, *Appl. Sci*, Vol. 8, No. 9, pp. 1575-1589.
<https://doi.org/10.3390/app8091575>
- [15] J. P. Yun, 2020, Automated defect inspection system for metal surfaces based on deep learning and data augmentation, *JMS*, Vol. 55, pp. 317-324.
<https://doi.org/10.1016/j.jmsy.2020.03.009>
- [16] D. Praveen, M. V. Achuth Rao, R. Jordan, Y. J. Ahmad, 2021, ICAPAI, *IEEE*, Manhattan, pp. 1-6.
 DOI: 10.1109/ICAPAI49758.2021.9462060
- [17] X. Feng, X. Gao, L. Luo, 2021, X-SDD: A New Benchmark for Hot Rolled Steel Strip Surface Defects Detection, *Symmetry*, Vol. 13, No. 4, pp. 706-721.
<https://doi.org/10.3390/sym13040706>
- [18] D. E. Kim, B. K. Jeon, D. S. Kwon, 2018, 3D Convolutional Neural Networks based Fall Detection with Thermal Camera, *J. Kor. Robot. Soc*, Vol. 13, No. 1, pp. 45-54.
<https://doi.org/10.7746/jkros.2018.13.1.045>
- [19] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, L. D. Jackel, 1989, Backpropagation applied to handwritten zip code recognition, *Neural. Comput*, Vol. 1, No.4, pp. 541-551.
 DOI:10.1162/neco.1989.1.4.541
- [20] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, 1998, Gradient-based learning applied to document recognition, *Proc. IEEE*, Vol. 86, No. 11, pp. 2278-2324.
 DOI: 10.1109/5.726791
- [21] S. Behnke, 2003, Hierarchical neural networks for image interpretation, Vol. 2766, Springer, Heidelberg, pp. 1-244.
 DOI: <https://doi.org/10.1007/b11963>
- [22] P. Y. Simard, D. Steinkraus, J. C. Platt, 2003, *ICDAR*, *IEEE*, Manhattan, pp. 958-963.
 DOI: 10.1109/ICDAR.2003.1227801
- [23] A. Krizhevsky, I. Sutskever, G. E. Hinton, 2017, ImageNet classification with deep convolutional neural networks, *Commun. ACM*, Vol. 60, No. 6, pp. 84–90.
<https://doi.org/10.1145/3065386>
- [24] K. Simonyan, A. Zisserman, 2014, Very Deep Convolutional Networks for Large-Scale Image Recognition
<https://doi.org/10.48550/arXiv.1409.1556>
- [25] C. Szegedy, W. Liu, Y. Jia, 2015, *IEEE. CVPR. Conf, IEEE*, Manhattan, pp. 1-9.
 DOI: 10.1109/CVPR.2015.7298594
- [26] K. He, X. Zhang, S. Ren, J. Sun, 2016, Deep residual learning for image recognition, *Proc. IEEE conf. CVPR*, pp. 770-778.
<https://doi.org/10.48550/arXiv.1512.03385>
- [27] G. Huang, Z. Liu, L. V. D. Maaten, K. Q. Weinberger, 2017, Densely Connected Convolutional Networks, *Proc. IEEE conf. CVPR*, pp. 4700-4708.
<https://doi.org/10.48550/arXiv.1608.06993>
- [28] R. Wei, Y. Bi, 2019, Research on Recognition Technology of Aluminum Profile Surface Defects Based on Deep Learning, *Meter*, Vol. 12, No. 10, pp. 1681-1694.
 DOI: <https://doi.org/10.3390/ma12101681>
- [29] I. Kandel, M. Castelli, 2020, The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset, *ICT Express*, Vol. 6. No. 4, pp. 312-315.
<https://doi.org/10.1016/j.ict.2020.04.010>
- [30] H. G. Park, K. H. Bae, 2020, A Study on Detection and Resolving of Occlusion Area by Street Tree Object using ResNet Algorithm, *JKAIS*, Vol. 21, No. 10, pp. 77-83.
<https://doi.org/10.5762/KAIS.2020.21.10.77>
- [31] N. J. Kwak, H. J. Shin, J. S. Yang, T. S. Song, 2020, CNN Applied Modified Residual Block Structure, *KMMS*, Vol. 23, No. 7, pp. 803-811.
<https://doi.org/10.9717/kmms.2020.23.7.803>
- [32] Y. M. Park, S. Y. Gang, J. H. Chae, J. J. Lee, 2018, Classification Method of Plant Leaf using DenseNet, *KMMS*, Vol. 21, No. 5, pp. 571-582.

- <https://doi.org/10.9717/kmms.2018.21.5.571>
- [33] X. Fang, Q. Luo, B. Zhou, C. Li, L. Tian, 2020, Research Progress of Automated Visual Surface Defect Detection for Industrial Metal Planar Materials, *Sensors*, Vol. 20, No. 18, pp. 5136-5170.
<https://doi.org/10.3390/s20185136>
- [34] T. Luke, G. Nitschke, 2018, SSCI, IEEE, Manhattan, pp. 1542-1547.
- <https://doi.org/10.48550/arXiv.1708.06020>
- [35] C. Shorten, T. M. A. Khoshgoftaar, 2019, Survey on Image Data Augmentation for Deep Learning, *J. Big Data*, Vol. 6, No. 1, pp. 1-48.
<https://doi.org/10.1186/s40537-019-0197-0>
- [36] S. J. Kim, G. B. Kim, 2021, A Study on Surface Defect Classification Using Deep-Learning and Transfer-Learning, *KSDet*, Vol. 20, No. 1, pp. 64-69.