공학전문석사학위 연구보고서

해석 가능한 인공지능 기반의 이미지 분류 정확도 향상 연구

The improving accuracy of classifying an image using interpretable machine learning (IML)

2021년 2월

서울대학교 공학전문대학원
응용공학과 응용공학전공
김 성 현

해석 가능한 인공지능 기반의 이미지 분류 정확도 향상 연구

The improving accuracy of classifying an image using interpretable machine learning (IML)

지도 교수 윤성로

이 프로젝트 리포트를 공학전문석사 학위 연구보고서로 제출함

2021년 2월

서울대학교 공학전문대학원

응용공학과 응용공학전공

김 성 현

김 성 현의 석사 학위 연구보고서를 인준함

2021년 2월

위원장 구윤모 (인)

위 원 곽우영 (인)

위 원 윤성로 (인)

초 록

인공 지능은 데이터에 편향이 없다면, 인과 관계가 아닌 데이터 내의 패턴에 있어서 가장 높은 확률을 보여준다. 왜냐하면 대부분의 기계 학습 알고리즘은 데이터에 의존하고 있기 때문이다. 이 논문은 복잡성과 신뢰성 사이의 균형을 기반으로 인간의 해석 가능성을 다루는 것을 목표로 한다. 그러기 위해서, IML 또는 XAI 기술들 중에서 몇 가지 이론적 관점을 소개하고, 그 접근 방식의 이해를 바탕으로 실제 데이터 샘플을 사용하여 모델을 구현하고, 그 시각화 결과는 기계 학습 모델의 해석에서 일반화 오류에 대한 더 나은 이해를 제공함을 확인했다. 결과적으로 모델의 해석은 데이터에 잘못된 답이나 질문이 있는지 분석가의 편견을 명확히 하고, 전이 학습에 기반한 미세 조정의 근거를 제시한다.

주요어 : IML, XAI, 시각화, 전이학습, 해석가능성

학번 : 2019-22523

목 차

그 림 목 차

표 목 차

제 1 장

서론

2014년 개봉한 트랜센던스(transcendence)는 조니 뎁(Johnny Depp, 1963) 주연의 공상과학(SF, Science Fiction) 영화로, 트랜센던스는 초월이라는 뜻이다. 이 영화에서, 수억 년에 걸쳐 이루어낸 인류의 지적능력을 초월하고 목적의식과 같은 자각능력까지 갖춘 슈퍼 컴퓨터 '트랜센던스'의 완성을 목전에 둔 채, 천재 과학자인 윌은 반 과학단체에 의해 살해당한다. 이후, 그의 아내는 그의 두뇌를 컴퓨터에 성공적으로 업로드하게 되고, 슈퍼 컴퓨터가 된 윌은 온라인을 장악하고, 세상을 지배한다는 것이 대략적인 내용이다 [8]. 이러한 영화들을 접하면서 많은 사람들은 인공지능(AI, Artificial Intelligence)이 욕망을 가질 수 있다 여기고 막연히 두려워하는 것인지도 모른다. 또한 그러한 두려움 속에서, 연구자들은 인공지능의 판단 기준 또는 그 근거를 오랫동안 알고 싶어했고 그 연구가 이제 매우 현실적으로 다가오고 있다.

1.1 연구 동기

과연, 인공지능은 지혜를 가질 수 있는가? 괴델(Kurt Gödel, 1906-1978)은 인간의 지혜조차 한계가 있음을 증명했다 [9]. 그렇다면, 인간은 인공지능이 추론하는 지혜의 불완전성이 두려운 것일까, 아니면 단순히 그 통찰력의 메커니즘을 알고싶은 것일까. 매슬로우의 욕구단계 [10]에서 알 수 있듯이, 지혜는 고통에서 욕망으로 가는 매개체로서 인공지능의

지혜 또한 고통과 욕망이 전제되어야 할 것이다.

XAI(Explainable Artificial Intelligence)는 이러한 대중적인 호기심과, 특히 상업적 관심을 유도하는 “설명 가능(Explainability)” 이라는 표현으로 인해, 최근 몇 년간 많은 관심을 끌어왔다 [11]. 그러나, 학계에서는 XAI 보다는 IML(Interpretable Machine Learning)이라는 용어를 더 선호한다 [12]. 왜냐하면, 인공지능 모델의 ”해석 가능(Interpretability)” 에 대한 수학적 정의가 현재까지는 없으므로, 여기서의 ”설명 가능”은 인간에 의해 수행되는 이해를 바탕으로 한 ”해석 가능”을 의미한다 [13]. 따라서, 해석과 설명에 대한 요구가 광범위하게 연구되었지만, 일반화의 오류는 여전히 문제로 남아있고, 많은 특징(Features)을 찾아내고 더 많은 데이터를 준비해야 한다 [14].

1.2 연구 범위

IML이란, Interpretable Machine Learning 의 약자로서, 예측 결과를 들여다보고, 사람이 직접 해석할 수 있는 형태의 추가적인 정보를 제공할 수 있는 머신 러닝(ML, Machine Learning) 알고리즘을 연구하는 분야이다 [13]. 이는 보다 설명 가능한 모델을 만들기 위해 어려운 선택과 결정이 필요한 도전적인 영역이다. 앞서 서술했듯이, 아직까지 IML, 즉 XAI는 사람의 의사결정에 부가적인 해석을 보완해주는 분야로서, 해당 모델이 판단을 내리는데 긍정적인 요소와 부정적인 요소를 세그멘테이션(Segmentation)으로 부각시켜 시각적 설명을 제공하게 된다 [14]. 본 논문에서는 긍정적인 요소(True Category)와 부정적인 요소(False Category) 사이의 특징을 추출하고, 그 결과를 해석하여 일반화 오류를 줄이기 위한 방법으로 전이학습(Transfer Learning)을 수행하고, 데이터 주변의 희소 선형 결합

해석(LIME, Locally Interpretable Model-agnostic Explanations)과 클래스 활성화 맵(CAM, Class Activation Mapping), 그리고 Grad-CAM(Gradient-weighted CAM)을 사용하여 특징 맵(Feature Map)의 중요한 부분을 시각화하여 해석함으로써 합성곱 신경망(CNN, Convolutional Neural Net) 기반의 이미지 분류 모델의 성능 개선 효과를 확인한다. 또한 전이 학습 기반의 미세 조정(Fine Tuning)을 위해, 계층(Layer)별 특징을 Grad-CAM의 히트 맵(Heat Map)으로 확인하고 그 결과를 검증한다.

1.3 논문의 구성

본 연구 보고서는 전체 5개의 장으로 구성되어 있으며, 각 장은 다음의 내용을 포함한다. 제 1 장에서는 이 연구를 수행하게 된 배경과 목적을 설명하고, 그 범위에 대해 간략히 소개하였다. 제 2 장에서는 이 보고서를 이해하는데 필요한 몇 개의 설명 가능한 인공지능 기술을 소개하고, 이미지 분류 모델링 네트워크 및 최적화 기술 경향과 관련 연구를 살펴본다. 제 3 장에서는 본 연구에서 실제 적용하고자 하는 문제의 정의 및 데이터 예측 모델링과 최적화 수행 방법, 그리고 그 평가 방안에 대한 이론적 기반을 설명한다. 제 4 장에서는 실험 데이터와 환경 구성 및 다양한 실험을 통해 연구 진행 내역과 실험 결과에 대한 비교 분석을 수행한다. 마지막으로, 제 5 장에서는 이 연구 보고서의 결과 및 성과에 대해 요약하고, 부가적으로 필요한 사항을 제시함으로써 향후 연구의 방향성을 고민하고 마무리할 것이다.

제 2 장

관련 연구

이 챕터는 본 연구의 배경 지식에 관한 사전 연구로써, 머신 러닝 모델의 설명 또는 해석 가능성에 관련된 연구들에 대한 기본적인 개요를 제공한다. 먼저 제 1 절에서 설명가능한 인공지능에 대해 전반적으로 개괄한다. 이어서, 제 2 절에서는 데이터 주변의 희소 선형 결합을 활용한 모델 해석 방법인 LIME에 대해서 살펴본 후, CNN 모델의 특징 추출 시각화 기술인 CAM과 Grad-CAM에 대해 제 3 절에서 들여다 볼 것이다. 제 4 절에서 이 연구의 모델링 개발에 적용된 이미지 데이터 분류 방법인 CNN에 대하여 설명하고, 제 5 절에서는 데이터 부족 문제를 해결하기 위해 이 실험에 적용된 전이학습과 제 6 절에서 전이학습 기반의 미세 조정 기법에 대해 확인하고, 마지막 제 7 절에서 최상의 정확도(Accuracy)를 내기 위한 최적화 방법과 관련된 연구 경향을 소개할 것이다.

2.1 해석가능한 인공지능 (IML)

인공지능은 이제, 현대 정보 기술의 최첨단(state-of-the-art)을 넘어서 여러 분야에서 매우 중요한 핵심요소가 되었다 [15]. 그 중에서도 DNN(Deep Neural Net)과 같은 딥러닝(DL, Deep Learning) 모델은 최근 매우 활발하게 연구 및 활용되고 있지만, 그 거대한 파라미터(매개변수, Parameter) 집체적인 성질로 인해 대표적인 블랙박스(Black-Box) 모델로 불리운다 [16]. 이러한 블랙박스 모델이 작동하는 메커니즘(Mechanism)에 대한 이해를

찾고, 현재의 인공지능 모델의 효과성의 한계를 피하기 위해, 학습 성능을 유지하면서도 설명이 가능한 모델을 생성할 필요성이 대두되었다 [17]. 따라서, 설명가능한 인공지능이란, 인공지능을 이해하고 적절하게 신뢰하여 효과적으로 관리할 수 있도록, 모델 메커니즘과 예측에 대한 이해, 모델의 규칙을 시각화하거나 그에 대한 힌트를 제공하는 일련의 기술들을 말한다 [18]. 여기서 한가지, 앞서 서론에서 밝힌 바와 같이, 설명가능한 인공지능은 이제 더이상 낯선 기술이 아님에도, 아직 많은 커뮤니티에서 XAI와 IML이 혼용되고 있고, 특히 학계에서 IML이라는 용어를 선호하는 이유가 무엇인지 설명가능성과 해석가능성이라는 용어의 정리가 먼저 필요하다 [12]. 해석 가능성은 주어진 모델이 인간에게 의미있는 수준의 설명을 제공하는 모델의 수동적 특성을 나타낸다 [13]. 대조적으로, 설명가능성은 모델의 능동적 특성으로 볼 수 있으며, 인간과 의사결정모델 사이의 인터페이스로서의 설명이라는 개념으로 인간이 이해할 수 있는 의사결정모델의 적절한 조치나 절차에 가깝다 [19].

그렇다면, 우리는 왜 높은 성능을 보여주는 모델을 신뢰하지 못하고, 해석하고자 하는 것일까. 문제는 분류 정확도와 같은 성능지표는 대부분의 실제 작업에 대한 설명으로는 불완전하기 때문이다. 실생활과 밀접한 분야에 대해 점차적으로 인공지능 기술이 적용되면서, 공정성, 정확성, 인과성에 기반한 신뢰할만한 머신 러닝은 필수적이다. 알고리즘이 특정 결정을 내리는 근거를 이해할 수 있다면, 정의에 대한 실질적인 합의가 없더라도 IML은 모델의 타당성을 높이고 규제 요구 사항(예: GDPR 및 CNIL, 프랑스 행정 규제 기관)을 충족하며 인간에게 결정을 설명하고 기존 모델을 개선 할 수 있다 [20].

2.1.1 IML 기술 분류 (Taxonomy)

IML 기술을 나누는 몇가지 분류 방법이 있으나, 널리 받아 들여지는 분류는 크게 투명한(transparent) 모델과 외부 XAI 기술로 설명할 수 있는 사후 설명 가능성(post-hoc explainability)으로 나누는 방법이다 [19].

투명한 모델은 설계부터 자체적으로 어느 정도의 해석가능성을 전달한다. 즉, 그 자체로 이해할 수 있는 경우 투명한 것으로 간주된다. 이 범주에 속하는 모델은 해석 가능한 영역, 즉 알고리즘의 투명성, 분해 가능성 및 시뮬레이션 가능성 측면에서도 접근할 수 있는데, 이러한 측면을 고려해서 1) 입력에서 출력으로의 매핑이 사용자에게 보이지 않는 불투명한(opaque) 시스템, 2) 사용자가 매핑을 수학적으로 분석 할 수있는 해석 가능한(interpretable) 시스템, 3) 모델이 매핑의 근거를 이해하는 데 도움이 되도록 특정 기호나 규칙을 출력하는 이해 가능한(comprehensible) 시스템과 같은 더 넓은 구분도 있다. 선형회귀분석(Linear Regression), 의사결정나무(Decision Trees), K근접이웃(K-Nearest Neighbors), 규칙기반학습(Rule-based Learners), 일반가법모델(Generalized Additive Models)과 베이지안 모델(Bayesian Models)과 같은 통계학적 기반의 알고리즘들이 그 예이다 [21].

사후 설명 가능성은 텍스트 설명, 시각적 설명, 지역적 설명, 예에 의한 설명, 단순화에 의한 설명 및 특징 관련 설명 기술과 같은 해석 가능성을 높이기 위해 다양한 수단을 사용하여 설계상 쉽게 해석할 수 없는 모델을 대상으로 한다. 즉 이미 개발된 모델의 주어진 입력에 대해 예측을 생성하는 방법에 대한 이해 가능한 정보를 전달하는 것이 사후 설명 가능성 또는 모델링 후 설명 가능성 기술의 목적이다. 이 기술은 다시 모델 불가지론적(Model-agnostic) 기술과 특정 모델지향적(Model-specific)

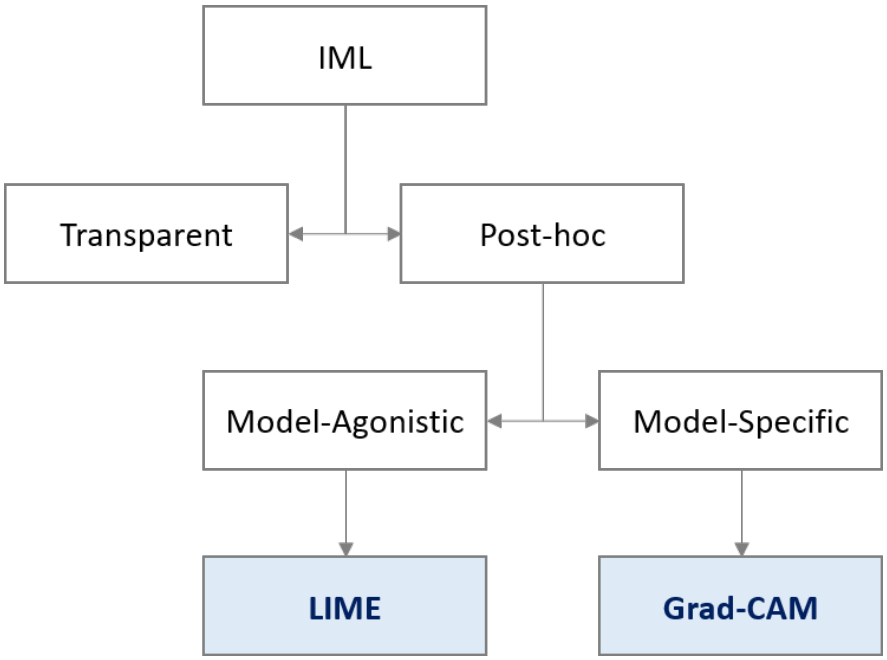


그림 1: 해석가능성 기반 IML 분류도.

기술로 나뉜다. 모델 불가지론적 기술은 설명하고자 하는 모델의 내부 처리 또는 내부 표현을 무시하고 모든 머신 러닝 모델에 원활하게 적용할 수 있는 기술로서, 예측 절차에서 일부 정보를 추출 할 목적으로 모든 모델에 연결되도록 설계되었으며, 다루기 쉽고 단순화 기술을 사용하여 복잡성을 줄인다. 대표적으로 LIME이 있으며, 다음 절에서 살펴볼 것이다. 특정 모델지향적 기술은 다시 얕은(shallow) 머신 러닝 모델의 사후 설명성 기술과 깊은(deep) 딥러닝 모델의 사후 설명성 기술로 나뉘는데, 랜덤 포레스트(Random Forests)나 그래디언트 부스팅(Gradient Boosting) 같은 트리 앙상블(Tree Ensembles)이나 서포트 벡터 머신(SVM, Support Vector Machines)을 활용할 때, 어떤 입력 변수가 실제로 관련 출력 데이터와 관련이 있는지 변수 중요도를 보여주는 머신 러닝 모델의 사후 설명적인

방식은 이미 일반적으로 활용되고 있다. 사후 설명 및 특징 관련 기술은 점차적으로 DNN, CNN과 RNN(순환 신경망, Recurrent Neural Network)과 같은 딥러닝 네트워크를 설명하는 데 많이 채택되고 있다. 특히, 인간은 시각적 데이터에 대한 이해를 선호하기 때문에 CNN의 설명가능성 기술은 다른 유형보다 매우 활발히 연구가 이루어지고 있다. CNN이 학습한 내용을 이해하는 기술은 크게 입력 공간에서 출력을 다시 매핑하여, 입력의 어떤 부분이 출력에 대해 차별적인지 확인하는 기술(예. Deconvnet)과 CNN의 네트워크가 중간 계층에서 어떻게 외부 세계를 내다보는지를 해석하는 기술(예. Grad-CAM)로 나뉠 수 있다 [22].

이 논문에서는 이미지 데이터를 활용한 결함 분류 모델을 구축하여, 그 분류 성능 지표인 정확도를 개선하고자 IML기술을 활용한다. 따라서, 모델의 정확도를 유지하면서 모델의 예측을 들여다보기 위해 사후 설명 가능성 기술에서 두가지 기법을 선택한다. 바로, 모델 불가지론적 기술에서 시각적 설명에 있어서 가장 간단한 접근법인 LIME과, 특정 모델지향적 기술에서 네트워크의 중간 계층에 대한 시각적 설명으로 해석하는 Grad-CAM이다.

2.2 데이터 주변의 희소 선형 결합 해석

LIME은 머신 러닝 모델의 예측을 해석하기 위해 활용된다고 널리 알려진 IML의 사후 설명 가능성 중 모델 불가지론적 알고리즘 중에 하나이다. [1] 이 알고리즘은 해석하고자 하는 모델을 전체가 아닌 지역적으로 근사함으로써 분류(Classification)나 회귀(Regression) 모델의 예측을 설명해주는 기술이다. 또한, 의사 결정 트리(Decision Trees), 신경망(Neural

[1] 저자는 제목조차도 “Explaining the Predictions of ‘Any Classifier’” 라고 했다.

Network), 로지스틱 회귀분석(Logistic Regression), 랜덤 포레스트(Random Forest)나 SVM 등 어떠한 예측 모델에서도 활용 가능하고, 일반적인 정형 데이터 외에도, 이미지 또는 텍스트와 같은 다양한 비정형 데이터에도 적용할 수 있다는 장점이 있다 [1].

해당 논문은 모델의 개별 예측 값을 설명하기 위한 LIME 알고리즘과 모델 자체의 신뢰 문제를 풀기 위해 서브 모듈(Sub-module)을 선택하는 SP(Submodular Pick)-LIME 알고리즘으로 나뉘는데, 여기서는 이미지 분류 모델의 결과에 대한 시각적 설명을 표현하기 위한 LIME 알고리즘과 관련한 논문의 내용을 요약해 보기로 한다.

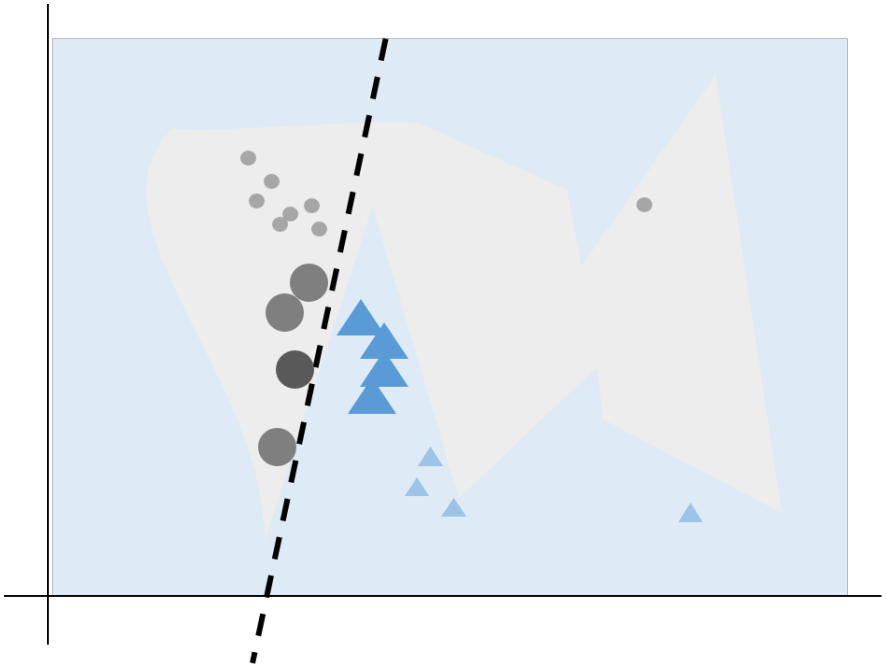


그림 2: LIME의 지역적 설명을 보여주는 예시 [1].

저자는 LIME은 추출된 데이터 주변의 인스턴스(Instance)를 샘플링(Sampling)하고 그 샘플 데이터와의 근접성에 따라 가중치를 부여한다고

설명한다. 위 그림의 점선은 지역적이고 선형적으로 학습된 예측 결과에 대한 설명 그 자체를 나타낸다고 볼 수 있는데, LIME을 한마디로 이해하기 위한 키워드는 지역성(Locality)과 선형성(Linearity)이라고 한다. 우선, LIME은 지역적으로 즉 국지적인 탐색을 하기 위해 샘플 데이터 주변의 인스턴스를 샘플링 한다. 이 때, 알고리즘의 설명을 특정 모델에 구애 받지 않도록 샘플 데이터의 함수에 대해 어떤 가정도 하지 않고 그 손실(Loss)을 최소화하면서 그 함수의 지역적인 패턴을 학습하기 위해 인스턴스에 가중치를 적용하여 근사(Assumption) 한다. 그리고, Lasso[2]로 K개의 특징을 선택하여 최소 제곱을 통해 인스턴스와의 근접성에 따라 적용된 가중치를 학습하여 대략적으로 계산한다. [3] 따라서, 샘플과의 거리가 가까운 인스턴스는 가중치가 높고, 거리가 먼 인스턴스는 가중치가 낮아지게 된다. 이로써 원래 모델이 복잡해서 전체적인 설명을 할 수 없더라도, LIME은 샘플 데이터에 의해 포착된 인스턴스의 주변 영역에 충실한 설명을 제공할 수 있게 된다 [1].

Algorithm 1 `Sparse Linear Explanations using LIME` [1]

Require: Classifier f, Number of samples N
Require: Instance x, and its interpretable version x'
Require: Similarity kernel π_z, Length of explanation K
  $Z \leftarrow \{\ \}$
  for $i \in \{1,2,3, ..., N\}$ **do**
    $z'_i \leftarrow$ sample around(x')
    $Z \leftarrow Z \cup (z'_i, f(z_i), \pi_z(z_i))$ sample around(x')
  end for
  $w \leftarrow$K-Lasso(Z, K) $\triangleright$ with z'_i as features, $f(z)$ as target
  return w

[2]통계 및 머신 러닝으로 생성된 통계 모델의 예측 정확도 및 해석성을 향상시키기 위해 변수 선택 및 정규화를 모두 수행하는 회귀 분석 방법이다 [23][24].

[3]잠재적으로, K는 사용자가 처리 할 수 있는 크기로 조정되거나 인스턴스마다 다른 값을 가질 수 있다. 최적의 K값에 대한 논의는 해당 논문에서 별도로 논의하지 않는다.

물론, LIME의 치명적인 단점을 지적하는 연구들도 찾아볼 수 있다. 이미지 데이터를 활용한 모델의 설명은 특정 클래스에 대해 양(Plus)의 가중치를 가진 슈퍼 픽셀(Super-pixel)을 강조함으로써 픽셀 단위로 표시하게 된다. 결과적으로 이미지 분류 모델의 예측 결과를 설명할 때, LIME의 출력은 이미지 데이터 샘플의 예측 결과에 대한 픽셀 단위의 기여도(True or False)를 보여준다. [4] 일반적으로, LIME은 인스턴스 구성 요소 간의 관계에 대한 양적 이해(Quantity understanding)가 아닌 질적 이해(Qualitative understanding)를 제공하기 위해 그에 관한 해석을 앞서 설명한 바와 같이, 시각적인 결과물로 표시하기 때문에 직관적인 설명이 확인 가능한 장점이 있지만, 학습된 모델에 대한 LIME의 설명자(Explainer)는 로컬에서(locally) 모델 예측 결과의 좋은 근사치(Approximation)를 보여줄 뿐, 글로벌(globally) 근사치는 아니라는 한계가 있다 [25][13].

2.3 CNN 모델의 특징 추출 시각화

IML의 모델지향적 기술에서 네트워크의 중간 계층에 대한 시각적 설명으로 해석하는 Grad-CAM은 CAM의 일반화 기술로서, 이미지 분류 모델의 해석을 위해 CNN의 관심 영역을 확인하는 방법인 특징 추출 시각화(Feature extraction visualization) 분야에서 최근에 많이 사용되는 방법이다. CAM은 CNN의 해석을 위해 GAP(Global Average Pooling) Layer를 사용하여 시각적인 설명을 제공하고, 이후에 나온 Grad-CAM은 그래디언트(Gradient)를 이용하여 GAP Layer에 의존하지 않는 방법을 제안한다 [26] [27]. 여기서는 두가지 솔루션의 구현 방법과 활용 방안을 비교해 본다.

[4] 이미지의 해석 가능한 표현은 이진(Binary) 형태로 1이 원래 슈퍼 픽셀을 나타내고 0은 회색(Grayed out)으로 표시된다.

2.3.1 클래스 활성화 맵 (CAM)

CAM은 CNN을 해석하고자 하는 생각에서 비롯되었다. 특정 범주에 대한 CAM은 해당 범주를 식별하기 위해 CNN에서 GAP을 사용하여 추출된 이미지 영역을 나타낸다. 이러한 특징 맵을 생성하는 절차는 간단하다. 일반적인 CNN 네트워크의 대부분은 컨볼루션(Convolution, 합성곱) 계층으로 구성되어 있으며 최종 출력 계층(Output Layer)이나 분류의 경우엔 소프트맥스(Soft-max) 바로 전에 완전 연결(FC, Fully Connected) 계층에서 이미지의 공간적인 정보가 사라진다. 이 문제를 해결하기 위해 CAM은 완전 연결 계층 대신에 GAP Layer를 사용한다 [26].

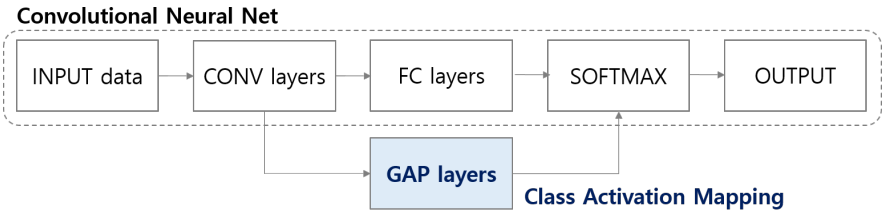


그림 3: GAP을 사용한 CAM 절차

우리는 전통적으로 머신 러닝 모델에서 정확도와 해석 가능성 사이에는 트레이드오프(Trade-off)가 존재한다는 것을 알고있다. CAM의 위와 같은 네트워크 아키텍처의 변화는 정확도의 감소를 필연적으로 가져온다. 해당 논문에서는 GAP Layer의 전단에 컨볼루션 계층을 추가하여 성능의 보전했다.

2.3.2 Gradient-weighted CAM

CAM은 시각적 설명을 통해 CNN을 더 투명하게(transparently) 만들어 주지만, 해석하고자 하는 모델의 아키텍처를 변경해야 하고, 변경된

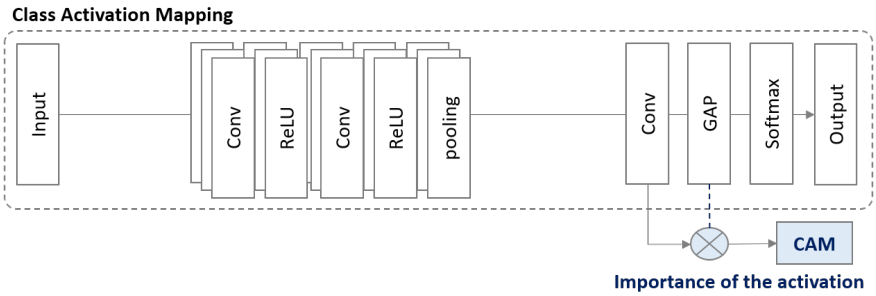


그림 4: CAM Architecture

네트워크로 재 학습(Re-training) 하게 되면 필연적으로 성능 저하를 유발한다. 이러한 문제점을 해결하기 위한 방안이 Grad-CAM이다. 이 기술은 네트워크의 모든 계층에서의 활성화를 설명하는 데 활용할 수 있다는 점에서 CAM에 비해 일반적이다. 왜냐하면, Grad-CAM에 의한 시각화는 머신 러닝 모델 개발자가 이해하기 힘든 모델의 예측이 어느 정도 합리적인 설명력을 가지고 있음을 보여주고, 데이터 세트(Data set)의 편향(Bias)을 식별하여 모델의 일반화를 달성하는 데 도움이 될 수 있기 때문이다 [27].

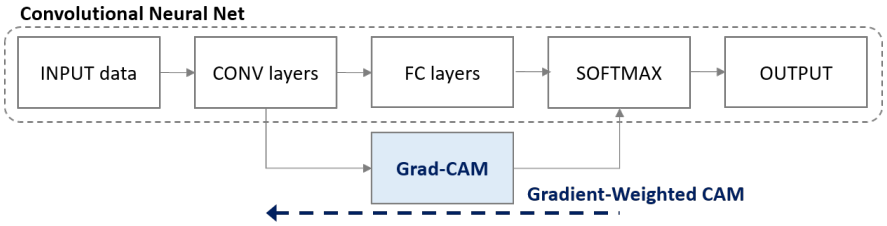


그림 5: Grad-CAM Architecture

CNN의 컨볼루션 레이어는 네트워크의 층이 깊어질수록 더 높은 수준의 시각적 표현이 가능하고, 또한 자연스럽게 그 이미지의 차원 정보를 유지할 수 있지만 완전 연결 계층에서는 손실되므로, 마지막 컨볼루션 레이어가 높은 수준의 의미와 세부 공간 정보 사이에서 최상의 절충안을

가질 것으로 기대된다. 이러한 계층의 뉴런은 각 이미지 부분에서 의미를 가지는 클래스의 유용한 정보를 찾아낼 수 있다. Grad-CAM은 이러한 CNN의 마지막 컨볼루션 레이어로 흐르는 그래디언트 정보를 사용하여 특정 클래스의 관심 유무를 결정할 수 있게 된다.

2.3.3 CAM vs. Grad-CAM

CAM은 완전 연결 계층을 GAP Layer으로 바꾸어야 하며, 가중치(Weight)를 구하기 위해 재 학습시켜 소프트맥스에 넣음으로써 성능의 감소를 유발하는 단점이 있다. 그에 비해, Grad-CAM은 모델의 아키텍처를 변경할 필요가 없고, 모델을 다시 훈련할 필요가 없기 때문에 모델의 성능에 영향을 미치지 않는다. 왜냐하면, Grad-CAM은 이미 학습한 모델의 소프트맥스 입력 이전 특징 맵의 그래디언트 값을 통해 구하기 때문에 재학습이 필요 없다 [27].

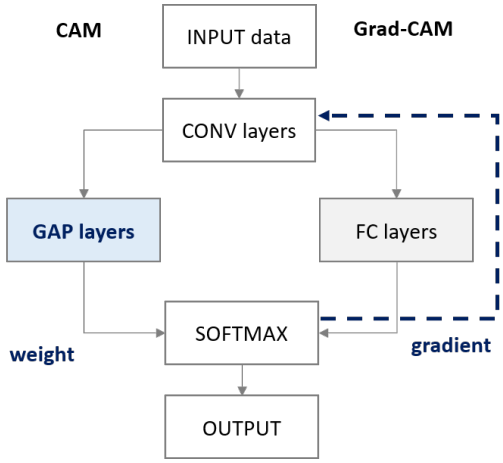


그림 6: CAM vs. Grad-CAM Architecture

2.4 합성곱 신경망 (CNN)

지금부터, 이제까지 소개한 설명가능한 인공지능을 활용하여 사후 설명을 해석하게될 모델과 관련한 연구들을 알아보자. CNN은 딥러닝에서 시각적 영상을 분석하는 데 사용되는 여러 개의 계층을 가지는 인공신경망의 한 종류로서 대량의 이미지 또는 비디오 인식 분야에서 많이 응용된다 [28].

CNN을 살펴보기 전에 먼저, 이미지를 인식하는 분야인 컴퓨터 비전의 일반적인 작업을 알아보면 다음과 같다.

1. 이미지 분류(Image Classification): 주어진 이미지 안에서 클래스(Class) 또는 레이블(Label) 이라고 불리는 컨텍스트(Context)의 패턴(Pattern)을 인식(Recognition)하는 것이다 [29].

2. 객체 인식(Object Recognition): 컴퓨터 비전 분야에서 이미지나 비디오 시퀀스 내의 사전에 지정되거나 학습된 객체를 인식하는 기술을 일컬으며 보통 객체 검출과 같은 의미로 쓰인다. 종종 객체 검출이 객체 인식보다 더 작은 의미로 해석되는 경우도 있다 [30].

3. 객체 검출(Object Detection): 관심 대상이 이미지 내에 있는지 여부를 판단하는 분류 뿐만 아니라, 모든 인스턴스의 위치를 찾는 객체 위치인식이 동시에 수행된다. 모델의 학습 목적에 따라서 특정 객체만 검출하는 경우도 있고, 여러 개의 객체를 검출하기도 한다 [31].

4. 객체 위치인식(Object Localization): 모델이 주어진 이미지 내에 목표 객체가 이미지에서 어느 위치에 있는지를 위치 정보로서 출력해 주는 것으로, 주로 바운딩 박스(Bounding box)를 많이 사용하며 바운딩 박스의 네개의 꼭지점 좌표를 출력하는 것이 아니라 주로 좌상(Left-Top), 우하(Right-Bottom)의 픽셀 좌표를 출력하는 것이 일반적이다 [32].

5. 객체 세그멘테이션(Object Segmentation): 객체 검출을 통해 검출된 인스턴스의 형상을 따라서 그 영역을 표시하는 것이다. 보통 이미지의 각 픽셀을 분류해서 결과 값을 도출한다 [2]. 아래에 추가로 나열된 세그멘테이션 기술은 그 종류와 활용이 다양하여 흥미로운 분야이지만, 이 논문에서는 다루지 않으므로, cs231n 강의에서 간략히 설명된 내용으로만 다음과 같이 정리한다.

6. 이미지 세그멘테이션(Image Segmentation): 이미지 세그멘테이션이란 이미지를 영역으로 분할, 즉 나누어 준다. 이런 분할된 영역들을 적당한 알고리즘을 사용해 합쳐서 객체 세그멘테이션을 수행한다 [2].

7. 시멘틱 세그멘테이션(Semantic Segmentation): 시멘틱 세그멘테이션은 객체 세그멘테이션을 하되 같은 클래스인 인스턴스들은 같은 영역 혹은 색으로 분할하는 것이다 [2].

8. 인스턴스 세그멘테이션(Instance Segmentation): 시멘틱 세그멘테이션에서 한발 더 나아가서, 같은 클래스이더라도 서로 다른 인스턴스들을 구분해주는 것이다 [2].

본 논문에서는 두가지 작업을 수행하게 되는데, 실제 산업 현장에서 수집된 이미지 데이터의 결함을 CNN으로 분류[5]하는 작업과 그 분류 모델의 사후 설명가능성 기술을 활용하여 모델의 특징을 추출하여 이미지 세그멘테이션으로 시각화하는 작업이다.

합성곱 신경망이라는 이름은 네트워크가 합성곱이라는 수학적 연산을 사용함을 나타낸다. 즉, 합성곱은 특수한 종류의 선형 연산으로, CNN 내의 합성곱 계층은 일반 행렬 곱셈 대신 합성곱 연산을 사용한다. CNN은 입력(Input) 및 출력(Output) 계층과 여러 개의 또는 깊은(deep) 숨겨진(hidden) 은닉 계층으로 구성되는데, 은닉 계층은 일반적으로 곱셈 또는

[5] 이미지 분류를 말한다.

다른 내적(Inner Product)과 합성곱을 하는 일련의 합성곱 계층으로 구성된다. 활성화 함수는 일반적으로 ReLU Layer이며 그 뒤에는 풀링 계층(Pooling Layer), 완전 연결 계층 및 정규화 계층과 같은 추가 계층이 따른다 [4].

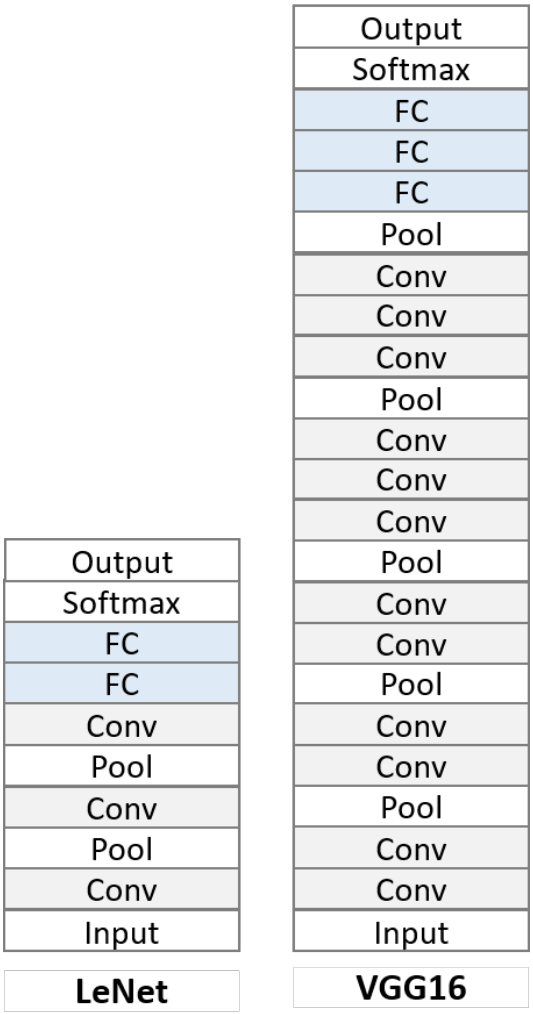


그림 7: LeNet vs. VGGNet Architecture [2].

이 연구에서는 다양한 CNN 알고리즘 중에서도 구성이 간단하여 응용에 용이한 VGGNet을 활용한다 [32]. 옥스포드 대학의 연구팀 VGG에 의해 개발된 모델인 VGGNet은 2014년에 ILSVRC로 불리는 이미지넷 데이터를 활용한 이미지 인식 경진대회(ImageNet Large-Scale Visual Recognition Challenge)의 검출(Localization) 부문에서 우승, 분류(Classification) 부문에서 준우승한 모델이다. 객체 분류 부문에서 7.4%의 오류 비율을 달성하며 준우승한 모델이긴 하지만, 앞서 언급한 활용하기 쉬운 그 간단한 구성과 성능(속도) 때문에, 우승한 모델인 구글의 GoogLeNet보다 많은 머신 러닝 모델 개발자들에게 인기를 얻어왔다. VGG-16, VGG-19와 같은 모델명으로 불리는 VGGNet은 각각 16개 또는 19개의 층으로 구성된다. VGG는 합성곱 계층과 풀링 계층, 그리고 완전 연결 계층으로 구성되는 CNN 모델이다 [33]. 특히, VGG 연구팀이 초록에서 밝히는 바와 같이, 그들이 연구한 핵심은 바로 네트워크를 깊게 만드는 것이 모델의 성능을 어떻게 변화시킬 수 있는지를 확인하고자 하는 것이었다. 따라서, VGGNet부터 네트워크의 깊이가 확연하게 깊어지기 시작했다.

VGGNet의 구성을 간단히 살펴보자. 깊이의 영향만을 최대한 확인하고자 컨볼루션 필터 커널(Filter Kernel)의 사이즈는 가장 작은 3 x 3으로 고정하고, 풀링 계층을 두어 크기를 절반으로 줄여주도록 처리했다. 하지만, 이러한 VGGNet 모델도 단점이 있다. 바로 파라미터의 개수가 너무 많다는 것이다. 그 이유는 위의 그림에서 볼 수 있듯이, VGGNet 또한 AlexNet과 마찬가지로 최종 출력 계층 전에 3개의 완전 연결 계층을 두는데, 이 부분에서 파라미터의 개수가 엄청나게 많아진다. 반대로 우승 모델인 GoogLeNet은 이러한 완전 연결 계층이 없으므로 파라미터의 개수를 확연하게 줄일 수 있었다 [34].

2.5 전이 학습 (Transfer Learning)

전이 학습이란, 사전에 학습된 모델(Pre-trained model) 즉, 내가 학습할 문제와 유사하면서도 대용량의 데이터로 기존에 학습이 되어 있는 모델을 활용하는 것을 말한다. 이러한 대량의 데이터로 모델을 학습시키는 것은 긴 시간과 고 사양의 하드웨어가 필요하지만, 다행스럽게도 이제 대부분의 머신 러닝 모델 개발자들은 이미지 넷 이미지 인식 대회 등에서 공개되어있는 모델(VGGNet, GoogLeNet 등)들을 이용할 수 있다 [35]. 아래 그림은 전통적인 학습과 전이 학습 기법의 차이점을 보여주는데, 전통적인 머신 러닝은 처음부터 학습을 시작하는 반면, 전이 학습 기술은 이전 작업의 지식(Knowledge)을 대상 작업에 전달(Transfer)하는 것을 볼 수 있다 [3].

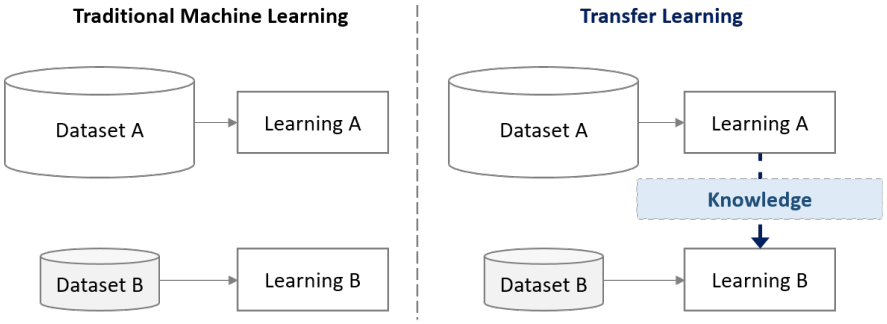


그림 8: 전이 학습 이란 [3].

CNN 네트워크를 어플리케이션 도메인(Application Domain)에 적용하려면 아래 그림에 표시된 두 가지 다른 CNN 기반 방법으로 수행할 수 있다. 첫 번째 방법은 연구된 데이터 세트에 적합한 CNN 네트워크가 생성되고, 해당 데이터 세트에 대해 완전히 훈련될 때, ”밑바닥부터 학습(Learning from Scratch)”이라는 용어를 사용한다. 두 번째 방법은 다른

데이터 세트에 의해 훈련된 일부 사전 훈련된 아키텍처에서 가중치 및 편향값 이라는 지식을 이전하는 것이다. 사전 훈련된 아키텍처의 이러한 지식이 작업 특화 데이터 세트으로 이어지는 것이 바로 전이 학습의 핵심 키워드이다 [36].

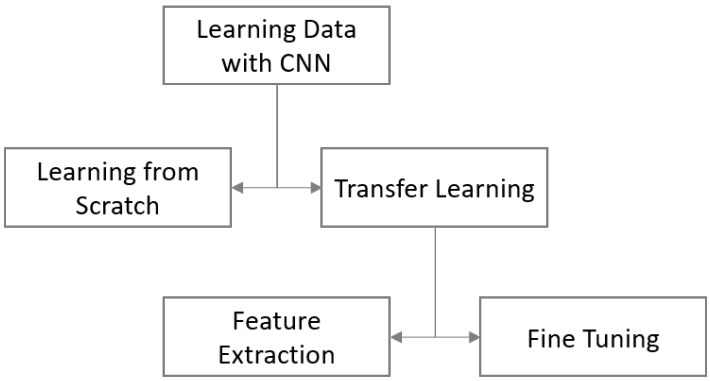


그림 9: 전이 학습 분류 [4].

지식의 전이 학습은 또한 고정 특징 추출기(Fixed Feature Extractor)와 미세 조정(Fine-Tuning)의 두 가지 방식으로 적용될 수 있다. 고정 특징 추출기는 네트워크를 유지할 필요없이 특정 작업에 전달된 사전 훈련된 가중치 및 편향을 직접 사용한다. 사전 훈련된 가중치 및 편향을 전송하여 초기화된 가중치 및 편향이 있는 작업 별 데이터 세트을 사용하여 네트워크의 일부 부분에서 네트워크를 재 훈련해야 한다. 즉, 기존 가중치 및 편향은 그대로 놔둔 뒤, 새로운 레이어를 추가해서 이를 학습하고 최종 결과를 내게끔 학습할 수 있다 [4]. 그 외에 새로운 데이터를 추가하여 처음부터 다시 시작하는 조인트 훈련(Joint Training)과 새로운 데이터로 가중치를 세밀하게 조정하되, 기존 데이터 분류 결과 또한 개선 가능하다고 주장하는 학습 방법(Learning without Forgetting) [37]도 있지만, 여기서는 앞으로 사용할 미세 조정의 방법과 전략 및 목표에 대해서만 알아본다.

2.6 미세 조정 (Fine Tuning)

전이 학습으로 신경망을 미세 조정하는 것은 좀 더 복잡한 지도 도메인 적용(Supervised domain adaptation) 기법으로, 분류 또는 회귀의 최종 레이어를 대체할 뿐만 아니라 이전 레이어 중 일부를 선택적으로 다시 훈련시킨다. 새로운 데이터로 다시 한번 가중치를 세밀하게 조정하도록 학습하고 기존 데이터는 기존대로 분류한다. 이로 인해 네트워크에서 레이어를 고정하여 특징 추출기로 사용해야 하는지, 아니면 어떤 절차로 레이어를 미세 조정해야 하는지 의문이 생긴다. 딥 러닝 모델에서 첫번째 계층은 "일반적인(general)" 특징이 추출되도록 학습하게 되는 반면에, 모델의 마지막 계층에 도달하면서 특정 데이터 세트 또는 특정한 문제에서만 나타날 수 있는 "구체적인(specific)" 특징을 추출해 내도록 하는 고도화된 훈련이 이루어진다. 따라서 첫번째 계층을 포함하여 전반부의 계층들은 다른 데이터 세트의 이미지들을 학습할 때에 재사용될 수 있지만, 뒤로 갈수록 후반부의 계층들은 새로운 문제를 만날 때마다 그에 맞는 훈련을 해야한다. 이러한 딥 러닝의 모델의 훈련 특성을 요약해보면, 첫번째 계층에서 추출된 특징이 일반적이고 마지막 층에서 추출된 특징이 구체적이라면, 네트워크 내의 어딘가에 일반적인 수준에서 구체적인 수준으로 넘어가는 전환점이 분명히 존재할 것이고, 그 레이어를 기준으로 가중치를 고정하고 업데이트(Update) 한다면 더 나은 성능을 확보할 수 있을 것이다. 또한 이러한 성질이 바로 전이 학습으로 신경망을 미세 조정하는 기술의 핵심이다[37].

전이 학습에서 컨볼루션 레이어와 풀링 레이어의 스택을 컨볼루션 베이스(Convolution base)라고 하고 출력 클래스를 예측하는 완전 연결 계층을 분류기(Classifier)라고 하는데, 사전 훈련된 CNN에서 컨볼루션 기반

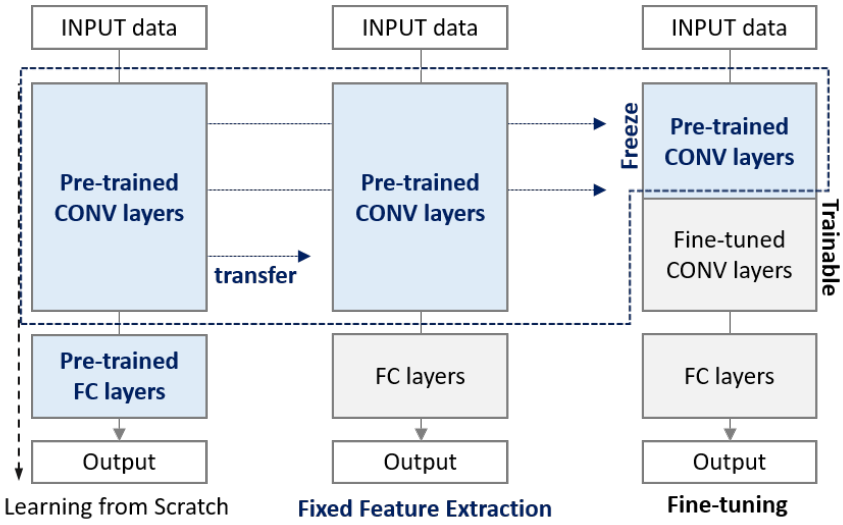


그림 10: 전이 학습 방법 [5].

의 처음 몇 계층은 입력 이미지에서 일반 기능을 학습할 가능성이 더 높기 때문에 이러한 계층을 우리는 다른 문제에 재사용할 수 있다. 그러나 컨볼루션 베이스의 마지막 몇 계층과 분류기는 특정 기능을 학습한다. 문제에 대해 사전 훈련된 CNN을 사용자 기준으로 정의하려면 분류기를 적절히 새 것으로 교체하고 마지막 몇 개를 미세 조정해야 한다. 요약하면 다음과 같다.

1. 사전 훈련된 CNN 모델을 선택
2. 사전 훈련 된 분류기를 새 분류기로 교체
3. 컨볼루션 베이스를 고정하고 추가된 분류자를 훈련
4. 컨볼루션 베이스의 마지막 몇 개 레이어를 고정 해제
5. 고정되지 않은 레이어와 추가된 분류기 모두 미세 조정

미세 조정 전에 추가된 분류기를 한 번, 미세 조정 중에 작은 학습률(Learning Rate)로 한 번 훈련한다. 이렇게 하면 이러한 계층을 통한 오

류 신호의 역전파로 인해 컨볼루션 계층이 학습한 특징의 현저한 왜곡을 방지할 수 있다. [38] 특히 데이터의 양이 적은 문제를 해결할 때, 데이터 세트의 크기와 학습된 데이터 세트과 사전 학습된 모델 간의 유사성을 고려하여 컨볼루션 베이스의 일부 계층을 고정하거나 재 학습할 계층을 적절히 조정할 수 있는 몇가지 전략을 취할 수 있다.

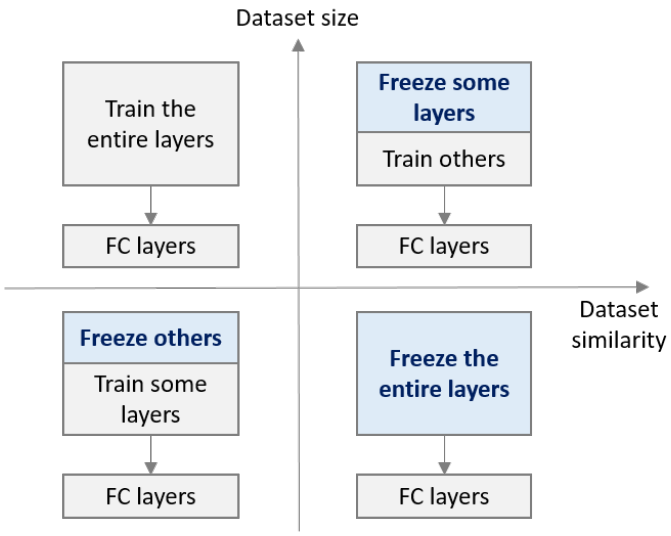


그림 11: 전이 학습 전략 및 목표 [5].

데이터 세트의 크기는 작지만 데이터 유사성은 매우 높은 경우, 데이터 유사성이 매우 높으므로 모델을 유지할 필요가 없다. 미리 학습 된 모델을 특징 추출기로 사용하여 출력 레이어를 문제에 다라 정의하고 수정한다. 데이터의 크기가 작고 데이터 유사성이 매우 낮은 경우, 사전 학습된 모델의 초기 레이어를 고정해야 한다. 또한 동결(freezing)이 완료되면 나머지 레이어를 다시 훈련한다. 그러나 최상위 레이어는 새 데이터 세트에 맞게 사용자 정의하고, 초기 레이어는 더 작은 크기로 사전 훈련 된 상태로 유지된다. 데이터 세트의 크기는 크지만 데이터 유사성은 매우 낮

은 경우, 데이터가 크기 때문에 신경망 훈련이 더 효과적이다. 또한 중요한 것은 우리가 사용하는 데이터가 다르므로 데이터에 따라 신경망을 처음부터 훈련시키는 것이 가장 좋다. 마지막으로, 데이터의 크기가 크고 데이터 유사성이 높은 최종적이고 이상적인 상황일 경우, 사전 훈련된 모델이 더 효과적이다. 또한 이 모델을 아주 좋은 방식으로 사용할 수 있다. 우리는 모델의 아키텍처와 모델의 초기 가중치를 유지하기 위해 모델을 사용해야 한다. 또한 사전 훈련된 모델에서 초기화 된 가중치를 사용하여 이 모델을 재 훈련할 수 있다 [4].

2.7 최적화 (Optimization)

딥러닝과 관련된 문제 중에서 가장 어려운 것이 신경망 훈련이다. 신경망 훈련 문제를 해결하기 위해 수백 대의 기계를 사용하여 며칠에서 몇 달의 시간을 투자하는 것은 매우 일반적이다. 이 문제는 매우 중요하고 비용이 많이 들기 때문에 이를 해결하기 위해 전문화된 일련의 최적화 기술들이 개발되었다 [4]. 이 장에서는 신경망 훈련을 위한 이러한 최적화 기술에 대해 알아본다.

가장 먼저 알아볼 최적화는 모든 기술의 기본이 되는 경사 하강법(GD, Gradient Descent)이다. 경사 하강법 함수의 기울기를 구하여 기울기가 낮은 쪽으로 계속 이동시켜서 최적값에 도달할 때까지 반복시키는 최적화(Optimization) 알고리즘이다 [39]. 다음은 경사 하강법의 확률적 근사치로 목적 함수를 최적화하기 위한 반복적 방법인 확률적 경사 하강법(SGD, Stochastic Gradient Descent)이다. SGD는 손실 함수의 기울기를 계산하여 이 기울기에 학습률을 계산하여 기존의 가중치를 업데이트하는 최적화 기술이다 [40].

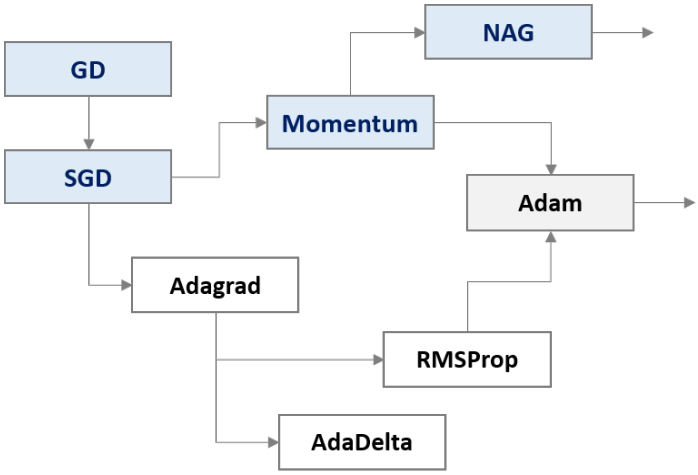


그림 12: 모멘텀 방식과 어댑티브 방식의 최적화

손실 함수의 기울기에 가속도(모멘텀)의 개념을 추가하여 속도가 커질수록 기울기를 크게 업데이트하는 방법으로 모멘텀(Momentum)이 있다. 함수의 경사 방향으로 운동량을 증가시키고, 기울기가 변경하는 차원에 대한 업데이트는 줄이게 된다. 결과적으로 수렴이 빨라지고 진동이 감소한다 [41] [42]. 네스테로프(NAG, Nesterov Accelrated Gradient)는 파라미터의 다음 위치에 대한 근사치를 제공한다 [43]. 파라미터의 위치를 대략적으로 알 수 있으므로, 불필요한 이동을 줄일 수 있는 최적화 기술이다. 모멘텀이 먼저 현재 기울기를 계산한 다음, 업데이트된 누적의 경사 방향으로 이동하는 동안, NAG는 먼저 이전에 누적된 경사 방향으로 이동한다. 기울기를 측정한 다음, 수정을 수행하여 완전한 NAG를 업데이트한다. 이는 업데이트가 너무 빨리 진행되는 것을 방지하고 응답성을 증가시켜 성능을 크게 향상 시킬 수 있다 [44].

학습률을 파라미터에 맞게 조정함으로써 더 작은 업데이트를 수행하는 Adagrad(Adaptive Gradient)는 자주 등장하거나 변화를 많이 한 변

수들의 경우 작은 크기로 이동하면서 세밀한 값을 조정하고, 적게 변화한 변수들은 더 크게 이동한면서 파라미터를 업데이트한다. 이러한 이유로 희소 데이터를 처리하는 데 적합하다. Adagrad의 주요 이점 중 하나는 학습률을 수동으로 조정할 필요가 없다는 것이다. 대부분의 구현은 기본값 0.01을 사용하고 그대로 둔다. Adagrad의 주요 약점은 분모에 제곱된 기울기가 누적된다는 것이다. 추가된 모든 항이 양수이므로 누적 합계는 훈련 중에 계속 증가한다. 이로 인해 학습 속도가 줄어들고 결국에는 알고리즘이 더 이상 추가 지식을 획득 할 수 없는 극히 작아지게 된다 [45]. 이러한 Adagrad의 학습 속도가 급격히 감소하는 문제를 해결해야하기 때문에 RMSprop와 Adadelta는 거의 동시에 독립적으로 개발되었다. RMSprop은 제프리 힌톤이 코세라(Coursera) 수업의 6e번째 강의에서 제안한 미공개 적응 형 학습률 방법이다 [46]. RMSprop는 학습률을 지수적으로 감소하는 평균 제곱 기울기로 나눈다. Hinton은 0.9로 설정할 것을 제안하지만 일반적으로 학습률에 대한 좋은 기본값은 0.001로 알려져 있다 [47].

Adam(Adaptive Moment Estimation)은 심층 신경망 훈련을 위해 특별히 설계된 적응형 학습률 최적화 알고리즘이다. Adam은 확실히 딥 러닝을 위한 최고의 최적화 알고리즘 중 하나이며 그 인기는 매우 빠르게 증가하고 있다. Adam이 각 반복에서 취한 실제 단계 크기는 대략 단계 크기 하이퍼 매개 변수에 제한된다. 이 속성은 이전의 직관적이지 않은 학습률 하이퍼 파라미터에 직관적인 이해를 추가한다. Adam 업데이트 규칙의 단계 크기는 기울기의 크기에 따라 달라지지 않으므로 작은 기울기가 있는 영역 (예: 안장 지점[6] 또는 협곡[7])을 통과 할 때 많은 도움이 된다. 이러한 영역에서 SGD는 빠르게 탐색하는 데 어려움을 겪고 있다. Adam은

[6] Saddle Point
[7] Narrow Long Valley

희소 그라디언트와 잘 작동하는 Adagrad와 온라인 설정에서 잘 작동하는 RMSprop의 장점을 결합하도록 설계되었다. 이 두 가지를 모두 가지고 있으면 더 넓은 범위의 작업에 Adam을 사용할 수 있다. Adam은 RMSprop에 SGD와 모멘텀의 조합으로 볼 수도 있다 [48]. 마지막으로, NAG에서 사용했던 방식대로 현재 위치에서 다음 위치로 이동할 그래디언트와 모멘텀 값을 구하는 것이 아닌 모멘텀 값으로 이동한 뒤에 그래디언트 값을 구해보자는 기술인 NAdam(Nesterov-accelerated Adaptive Memoment Adam)이 있다 [49].

Adam 최적화 알고리즘은 좋은 결과를 빠르게 얻을 수 있기 때문에 딥 러닝 분야에서 널리 사용되는 알고리즘이다. 지금까지는 Adam이 전체적으로 가장 좋은 선택일 수 있다 [50]. Adam은 매우 훌륭한 최적화 기술이며, SGD보다 훨씬 빠르고, 기본적으로 하이퍼 파라미터(Hyper Parameters)의 업데이트는 일반적으로 잘 작동하지만 자체적인 함정도 있음을 간과할 수 없다. 바로 Adam의 수렴 문제인데, 이로 인해, 많은 연구들이 SGD와 모멘텀을 융합한 최적화 기법이 훈련 시간이 길어질수록 더 잘 수렴될 수 있음을 밝히면서, 여전히 SGD는 애용되고 있다 [51][52][53].

제 3 장

연구 방법

본 섹션은 이 연구에서 구현하고자 하는 모델의 문제를 정의하고, 그 문제를 해결하기 위한 연구 범위와 실험 계획을 간략히 개관한다. 따라서, CNN 기반의 이미지 데이터를 활용한 결함 분류 모델을 생성함에 있어, 데이터 부족 문제를 해결하고 최적의 모델을 설계하여 분류 정확도를 향상시킬 수 있는 방법을 설계한다.

3.1 이미지 분류 모델링 개발

철강 공장에서 생산되는 대부분의 제품들의 결함은 크게 표면 결함과 내부 결함으로 나눌 수 있다. 표면 결함은 압력을 가하거나 롤링(Rolling) 작업 과정에서, 롤과 제품 사이에 발생하는 결함을 말하고, 내부 결함은 금속 자체적인 응고 과정에서 여러가지 원인으로 인한 비금속 물질의 균일하지 않은 내부 분포로 인해 검출된다. 표면 결함에 대해서는 많은 검출기나 인식 장치가 존재하고, 그 화상 이미지를 활용하여 머신 러닝 기반의 실시간 자동 표면 결함 검사를 위한 CNN 기법 연구도 여러 산업 현장에서 활발히 진행되고 있다. 이에 반해, 내부 결함 검출을 위한 대표적인 비파괴 검사에는 방사선 투과 검사 또는 초음파 탐지 검사가 있다. 초음파 검사는 제품 내부의 불연속을 초음파를 이용하여 검출하는 것으로, 결함의 위치와 크기 등을 측정할 수 있고, 두꺼운 대상의 내부 균열이나 결함 검출에 많이 활용된다 [54][55]. 특히, 초음파 검사로 검출되는 결함의 이미지를

활용할 때, 불량 발생 원인의 체계적 분석을 위한 결함 이미지의 패턴에 대한 분류가 필요하고, 그 분류를 위해 결함이 발생하는 패턴 별로 이미지를 자동으로 분류함으로써 근본 원인을 규명하고 결함의 발생 비율을 낮추어 개선이 가능하다. 수작업으로 분류할 때는 작업자 별로 다른 기준을 가지고 판단하므로, 원인 인자를 도출하는데 어려움이 있고, 불량률을 개선하는데 한계가 있다. 그러나, 자동으로 판정된 이미지를 후속으로 검증하게 된다면 결함 분류에 소요되는 시간을 줄여 생산성의 향상에 기여하고, 인적 편차를 해소하여 조업에 영향을 미치는 요소를 정량적으로 분석하고 관리할 수 있다.

3.2 성능 개선 문제의 해결방안

이렇게 초음파 검사로 검출되는 결함의 이미지를 수집하고 수집된 이미지를 활용하여 결함 분류 모델을 개발함에 있어 가장 적합한 학습 방법은 합성곱 신경망 모델을 사용하는 것이다. 합성곱 신경망은 이미지와 관련된 문제를 해결하기 위한 표준 형태의 신경망 아키텍처이다. 우선 특별한 기법을 사용하지 않고, 가장 기본적인 네트워크를 활용하여 처음부터 학습시켜 수집된 이미지의 분류 문제를 해결해 볼 것이다. 소규모 네트워크라 할지라도 검증된 기법을 활용하기 위해 1990년대에 CNN을 처음 고안한 Yann LeCun이 제시하고, 이후 Behnke에 의해 일반화 되고, Simard에 의해 단순화된 CNN의 가장 기본적인 아키텍처인 LeNet-5를 써서 간단하게 학습의 흐름을 파악하고 정확도를 확인한다 [7][56][57]. 일반적으로, 이미지 데이터 세트에 딥러닝을 적용하는 효과적인 방법은 대규모 데이터 세트에 이미 학습된 기존 네트워크를 활용하는 것이다. 사전 훈련된 네트워크는 이미지를 파악하기에 유용한 특징을 추출하는 법을 이

미 배워두었기 때문에, 앞선 사전 연구들에서 소개한 전이 학습이나 미세 조정으로 좋은 성능의 모델을 만들어 볼 것이다.

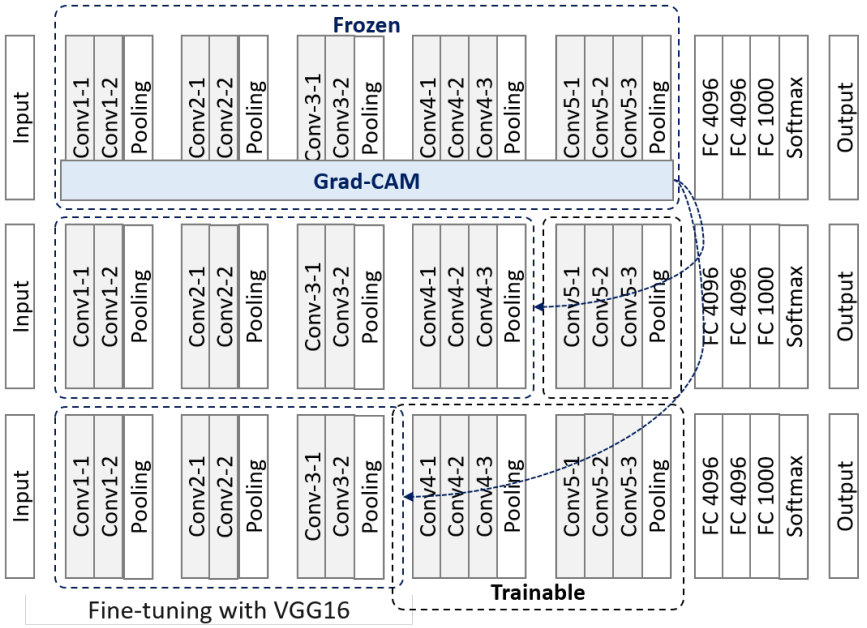


그림 13: VGG를 활용한 이미지 분류 모델 설계

서론에서 밝힌 바와 같이, 이 프로젝트의 목적은 모델 성능의 극대화가 아니라, 모델의 성능을 개선하기 위해 어떻게 모델의 내부를 들여다보고 그 결과를 활용하는 미세 조정의 과정에 있기 때문에, 구성이 간단하여 활용성이 뛰어나고, 학습 속도와 성능이 탁월한 VGG 모델을 기반으로 여러가지 방법을 적용해 볼 것이다. VGG-16 네트워크를 활용한 결과를 Grad-CAM을 통해 들여다봄으로써, 일부 블록을 조정하여 최적의 레이어를 찾아본다.

3.3 성능 평가 지표

정확도는 분류 모델을 평가하기 위한 측정지표 중 하나로, 모델의 예측이 얼마나 정확한지를 나타낸다 [6]. 그 정의는 다음과 같다.

표 1: 분류 성능 평가 - 혼동 행렬 [6].

		True Condition	
Predicted		True positive (TP)	False positive (FP)
		False negative (FN)	True negative (TN)

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \tag{3.1}$$

그러나, 데이터의 클래스 간 불균형으로 인해 정확도 지표 하나만으로 모델을 평가하기엔 한계가 있다. 클래스 불균형 문제를 가지는 모델을 평가하는 데 활용되는 지표로는 정밀도와 재현율이 있다. 정밀도는 바르게 확인된 양성의 결과에 바르게 확인되지 않은 결과를 더해 모든 양성 결과로 나눈 값이고, 재현율은 바르게 확인된 양성 결과를 나눈 값이다.

$$Precision = \frac{TP}{TP+FP} \tag{3.2}$$

$$Recall = \frac{TP}{TP+FN} \tag{3.3}$$

이진 분류의 통계 분석에서 F1 점수는 테스트 정확도의 척도로 정밀도와 재현율의 조화 평균으로 계산할 수 있다 [58][59]. F1 점수의 가능한 가장 높은 값은 완벽한 정밀도와 재현율을 나타내는 1이고 정밀도 또는 재현율이 0 인 경우 가능한 가장 낮은 값은 0이다.

$$F1Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{3.4}$$

특히, F1 점수는 세 개 이상의 클래스 (다중 클래스 분류)로 분류 문제를 평가하는데 사용된다. 이 설정에서 최종 점수는 마이크로(Micro) 평균화[1] 또는 매크로(Macro) 평균화[2]로 얻을 수 있다 [60]. 이번 실험의 분류 문제는 정상제품과 불량제품을 분류하는 문제와 같은 중요도와 빈도의 차이가 현저히 발생하는 문제와는 달리, 이미 결함이 있는 제품으로 분류된 데이터에서 결함의 종류를 분류하는 것이다. 그러나 각 클래스 단위로 빈도 차이는 존재하기 때문에, 모든 클래스를 똑같이 중요한 것으로 간주하되 약간의 가중치를 부여하는 가중화된 평균화 F1 점수(Weighted-average F1-score)를 정확도 외에 추가적인 성능 지표로 포함하기로 한다.

[1] 클래스 빈도에 따라 편향될 수 있다.
[2] 모든 클래스를 똑같이 중요한 것으로 간주한다.

제 4 장

실험 및 결과

이번 장에서는 제 3 장의 설계와 방법론으로 구성한 아키텍처로 모델을 생성하고 그 결과를 비교, 검증한다. 제 1 절은 본 연구의 실험 내용을 간략하게 요약하고, 제 2 절에서 실제 실험을 수행한 환경을 설명한다. 제 3 절은 수집된 이미지 데이터를 입력으로 한 비정형 데이터의 결함 패턴 분류 모델링을 처음부터 학습하는 스크래치 방법론부터 전이학습 방법론까지 실험한 결과를 살펴본다. 제 3 절에서 학습한 모델이 어떻게 예측하는지 제 4 절은 수퍼 픽셀로 표현한 모델링 결과 분석으로 제 5절은 CAM과 Grad-CAM 두가지 기술의 특징 데이터 시각화를 활용한 모델링 결과 분석을 통해 알아볼 것이다. 제 6 절은 5 절의 분석을 기반으로 미세 조정을 시행한 성능 개선을 확인하고, 마지막 제 7 절과 제 8 절은 모델의 검증 및 평가 그리고 모델간 성능 비교를 통해 실험 결과를 분석하고 정리한다.

4.1 실험 개요

본 실험에서 사용한 이미지 데이터는 철강 공장에서 생산되는 제품의 내부 품질 검사에 활용되는 초음파 탐상 장치로 검출된 결함 이미지 데이터 세트를 활용하였고, 제품 결함을 분류하기 위해 11개의 범주로 나누어 분석하였다. 2019년에 발생한 결함 중에서 총 6708개의 샘플을 수집하여 90%의 학습 데이터와 10%의 검증 데이터로 분리했다. 최종적으로 2020년도 1월에 발생한 84개의 샘플로 모델의 정확도를 평가했다.

표 2: 실험 데이터

데이터	학습	검증	평가
수량	5819	589	84

본 연구는 2014년 ILSVRC에서 준우승을 한 CNN 기반의 VGG-16 네트워크를 사용하여 결함 분류 모델을 설계했다. VGG-16은 3x3 컨볼루션 필터(Convolution Filters)를 사용하고, 13개의 컨볼루션 계층과 3개의 완전 연결 계층에 마지막 출력 계층은 분류를 위한 소프트맥스 함수를 사용한다 [33]. 여기서는 클래스의 수를 11로 바꾸고, 사전 훈련된 파라미터를 활용하여, VGG-16 모델을 훈련시켰다. 그리고, 앞서 설계한 모델로 학습을 진행한 후, LIME과 CAM, 그리고, Grad-CAM을 적용하여 특징 맵을 시각화하였다. LIME은 모형 자체가 어떠한 방식으로 작동하는지 이해하는 대신, 해석하고자 하는 예측 값의 근방에서 모형이 어떻게 작동하는지를 설명한다 [1]. CAM은 마지막 출력 계층에 앞서, CNN 특징 맵에 완전 연결 계층 대신에 GAP Layer를 적용하여 모델링한다 [26]. 마지막으로, Grad-CAM은 CAM의 일반화 모델로써, 각 컨볼루션 계층의 그래디언트를 사용해서 중요한 영역을 표현한다 [27].

전처리는 비즈니스 요건에 의해 사전에 처리된 것으로 이 연구에서 다루지 않는다. 대략적인 실험 사이즈를 파악하기 위해 토이 프로젝트로 간단한 컨볼루션 레이어를 몇개만 얹어서 실험해 보기로 했다. 네트워크 설계의 객관성을 확보하기 위해 최초의 CNN 모델인 LeNet-5를 가지고 우선 학습을 시킨 후, 일반적으로 성능향상에 도움이 된다고 알려져 있는 활성화 함수나 최적화 기술을 적용하여 성능을 개선해 보았다. 그런 다음, 이미지넷으로 사전 훈련된 모델을 전이학습으로 훈련시켜서 결함 이미지의 시각적 특징을 앞서 2장에서 논의한 사전 연구된 세 가지 기술을

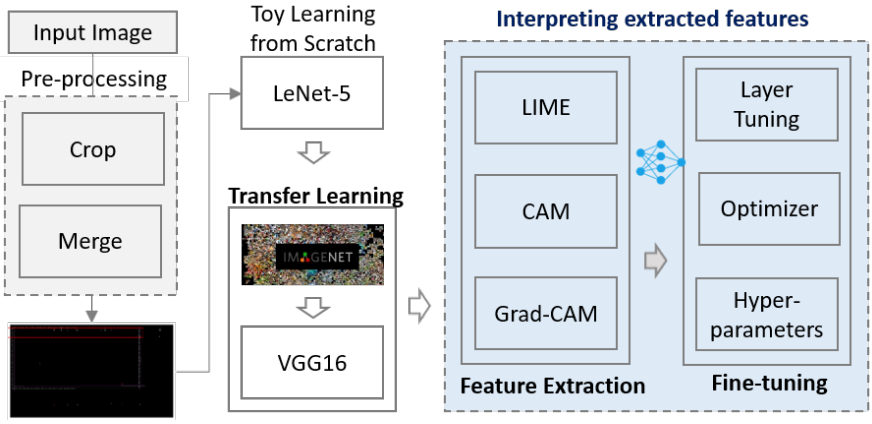


그림 14: 연구 흐름도

활용항 추출해보고 동결시킬 계층과 훈련시킬 계층을 나누어 학습시켜 보았다. 시각화 기술에서 해석한 모델의 설명력을 통해 최적의 계층 조정을 찾아낸 후, 최적화 및 초모수들에 대한 미세조정으로 최종적인 정확도를 확인했다. F1 점수를 보조지표로 포함한 정확도 향상을 위한 이 논문의 연구 흐름도는 위와 같다.

4.2 실험 환경

모델의 학습 및 성능의 평가를 위해 HOST 운영체제로 Linux RHEL 7.2를 기반으로 한 NVIDIA GPU Computing Server 환경에서 진행하고, NVIDIA TESLA M40 24G Driver의 안정적 할당 및 환경의 분리를 위해 Nvidia-Docker Engine 2.0 을 활용했다. 컨테이너 가상화 기술로 구축된 개별 분석 환경의 운영체제는 CentOS 7이고, 병렬 연산 모델은 각각 CUDA 9.0, cuDNN v7.1이다. 개발 언어는 Python 3.6.5를 기반으로 하여, Anaconda 5.2.0 기준의 파이썬 라이브러리 배포판을 적용했다. 따라서, 기본적인 데이터의 처리를 위해 numpy 1.14.3, pandas 0.23.0 라이브러리를 사

용하고, 데이터 정규화 및 학습 데이터와 평가 데이터 분리는 scikit-learn 0.19.1 라이브러리를 사용하게 된다. 시각화를 위해 각각 opencv-python 3.4.4.19 와 matplotlib 2.2.2 가 사용되었다. LeNet-5와 VGGNet의 학습을 검증하기 위하여 tensorflow-gpu 1.8.0과 tensorflow를 백엔드(Back-end)로 하는 Keras 2.2.4를 프레임워크로 사용하였다. 나열한 실험 환경 정보를 요약하면 다음과 같다.

표 3: 실험 환경 정보

세부 사양	
딥러닝 F/W	tensorflow-gpu 1.8.0 (Keras 2.2.4)
개발언어	Python 3.6.5
가상환경	Nvidia-Docker 2.0
병렬연산모델	CUDA 9.0, cuDNN v7.1
GPU Driver	NVIDIA TESLA M40 24G Driver
운영체제	Linux RHEL 7.2 (CentOS 7)

4.3 비정형 데이터의 결함 패턴 분류 모델링

수집된 이미지 데이터는 0-255의 RGB 계수로 구성되는데, 이 값은 모델을 효과적으로 학습시키기에 너무 크다. 그래서 이를 1/255로 스케일링하여 0-1 범위로 변환시켰다. 이는 다른 전처리 과정에 앞서 가장 먼저 적용되고, 적은 데이터 세트에서 최대한 많은 정보를 뽑아내서 학습할 수 있도록 이미지 데이터의 경우 데이터 증량(Augmentation)의 사례는 매우 일반적이다. 이미지를 사용할 때마다 임의로 변형을 가함으로써 마치 훨씬 더 많은 이미지를 보고 공부하는 것과 같은 학습 효과를 낼 수 있기 때문이다. 그러나, 증량에도 한계가 있고, 단순 기하학적인 변형으로 이미지의 특성이 크게 달라지진 않을 뿐더러 비즈니스적으로 제한되는 경우도

많기 때문에, 다른 기술들을 활용하여 학습하도록 한다.

4.3.1 스크래치

우선, 컨볼루션 레이어 3개를 쌓아놓은 간단한 형태의 LeNet-5를 사용하여 이미지 분류 모델을 생성하였다. 분류 대상은 총 11개 클래스이고, 전체 이미지는 6천여장의 데이터가 수집되었지만, 클래스 밸런스가 균일하지 않고 결코 큰 규모의 데이터라고 보기 어렵기 때문에 과적합 문제가 예상된다. 이러한 문제의 해결을 위해 이번 연구에서는 소규모의 네트워크인 LeNet-5를 적용했다.

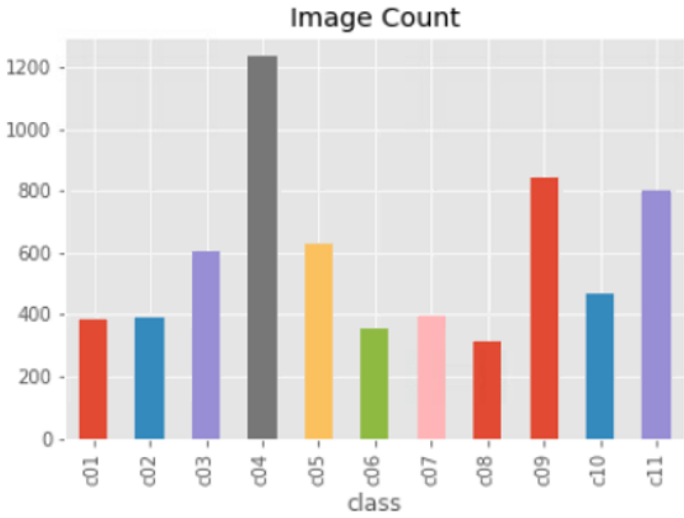


그림 15: 이미지 데이터 클래스 빈도

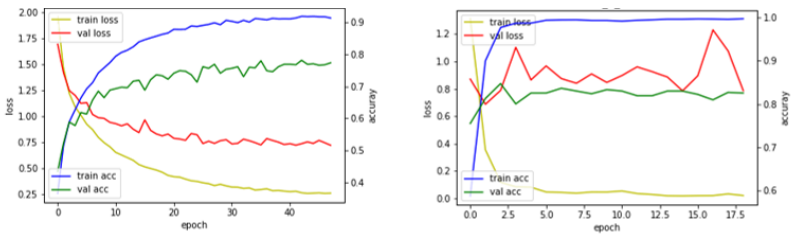
LeNet-5 는 7 개의 층으로 구성되며, 그 구성은 3 개의 컨벌루션 레이어, 2 개의 서브 샘플링 레이어(Subsampling Layer) 및 1 개의 완전 연결 레이어로 구성되어 있다. 참고로 서브 샘플링은 평균 풀링(Average Pooling)을 활성화 함수로는 하이퍼볼릭 탄젠트(tanh, Hyperbolic Tangent)를 사용

한다 [7]. LeNet-5 의 상세한 레이아웃은 다음 표에서 확인할 수 있다.

표 4: LeNet-5 레이아웃 [7].

Layer Type	Output(Shape)	Param #
INPUT (Data)	(None, 224, 403, 3)	0
C1 (Conv2D)	(None, 220, 399, 6)	456
S2 (AveragePooling2D)	(None, 219, 398, 6)	0
C3 (Conv2D)	(None, 215, 394, 16)	2416
S4 (AveragePooling2D)	(None, 107, 197, 16)	0
C5 (Conv2D)	(None, 103, 193 , 120)	48120
flatten (Flatten)	(None, 2385480)	0
F6 (Dense)	(None, 84)	200380404
OUTPUT (Dense)	(None, 11)	935
Total params: 200,432,331		
Trainable params: 200,432,331		
Non-trainable params: 0		

위와 같은 내용으로 가장 기본적인 CNN 모델을 활용한 학습한 결과, 각 에폭(Epoch)[1] 당 약 200초 정도 학습한 결과, 50에폭 이내로 0.75-0.83 의 검증 정확도(Validation Accuracy)를 달성했다.



(a) (b)

그림 16: LeNet-5 학습결과 (a) tanh-SGD (b) ReLU-Adam

[1]한 번의 에폭은 인공 신경망에서 전체 데이터 세트에 대해 forward/backward pass의 과정을 거친 즉, 전체 데이터 세트에 대해 한 번 학습을 완료한 상태를 말한다.

4.3.2 전이학습

앞 절의 LeNet-5를 활용했을 때, ReLU와 Adam과 같은 몇가지 기법을 추가하여 실험한 결과, 5%의 향상으로 0.83의 검증 정확도를 보였다. 일반적으로 여기서 성능을 향상하려면 신경망을 더 깊게 쌓게 되는데, 효과적인 학습을 위해 대규모 데이터 세트로 이미 학습되어 검증된 기존의 네트워크를 활용한다. 이를 위해, 3장에서 계획한 데로, 대규모 이미지넷 데이터 세트로 미리 학습된 VGG-16 네트워크를 사용하였고, 약 20회 정도의 학습만에 NAG 최적화 업데이트로 0.90의 검증 정확도에 도달했지만, 평가 정확도가 0.60 이하로 성능 차이가 나는 것으로 보아, 소규모 데이터에 과적합이 확인된다. 따라서, 좀 더 정교한 학습을 위해 VGG-16의 가중치로 초기화하여 학습하고, 0.91의 검증 정확도에 0.70의 평가 정확도를 확보했다. 이제 기존 네트워크만 활용하는 것 뿐만 아니라, VGG가 학습한 내용까지 활용하는 전이학습의 과정으로 진행한다.

표 5: VGG-16 학습결과

Architecture	Loss	Val. Accuracy	Test Accuracy
randomize weight	0.544	0.908	0.569
feature extraction	0.554	0.813	0.625

우선, VGG-16의 전체 컨볼루션 베이스를 동결시키고, 완전 연결 계층만 학습하여 성능을 평가하는 특징 추출 모델을 학습시켰고, LeNet-5와 유사한 결과를 얻었다. 이미지넷의 데이터와 실험 데이터의 특징이 확연히 다르기 때문에, 과연 VGG가 이미지넷에서 학습한 내용을 새로운 문제 상황으로 옮겨올 수 있을지, 그것이 가능하다면 기존 네트워크의 어느 정도까지 활용하여 학습하고자 하는 데이터를 훈련시킬지 몇가지 IML의 기법으로 확인해보았다. 다음 두 절에서 VGG가 학습한 내용으로 새로운

문제를 어떻게 해결할지 살펴보자.

4.4 수퍼 픽셀로 표현한 모델링 결과 분석

이제 앞 절에서 학습된 모델들 중에서 가장 좋은 성능을 특징 추출 모델로 하여 이미지 데이터의 특성을 파악하기 위해, 우선 LIME을 활용하여 이미지 특징을 추출해 보았다.

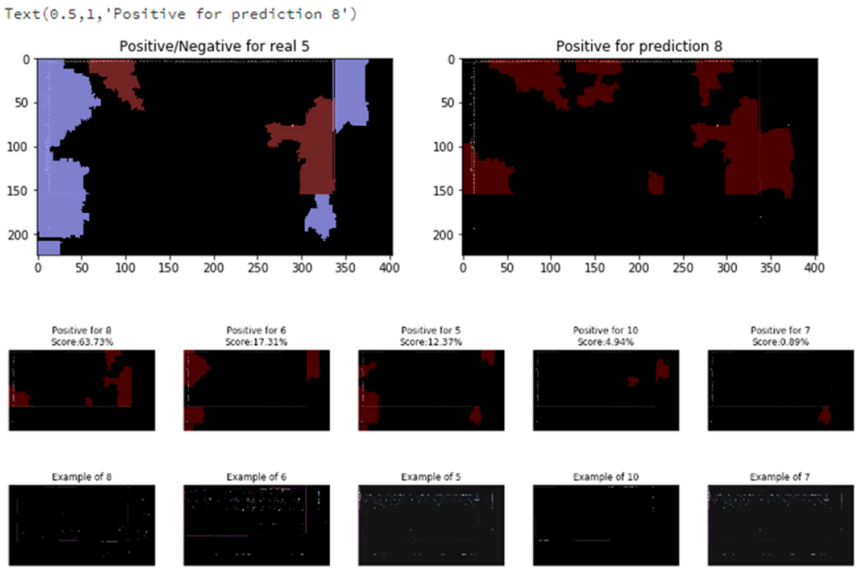


그림 17: LIME을 활용한 이미지 특징 추출

위 그림을 보면, LIME의 특징 맵은 긍정과 부정의 특징을 각각 볼 수 있고 다른 클래스들의 긍정 특징의 비율까지 확인할 수 있다는 장점은 있지만, 최종 출력 계층의 특징만을 추출하고 그 고유의 지역성과 선형성으로 인해 [13], LIME이 보여주는 각 예측의 긍정과 부정의 특징은 현장의 작업자가 바라보는 관점과는 다소 차이를 보였다.

4.5 특징 시각화를 활용한 모델링 결과 분석

이번 절에서는 CAM과 Grad-CAM의 특징 추출 맵을 확인해 보았다.

4.5.1 CAM

CAM의 특징 추출 결과는 모델이 가장 핵심적으로 보는 픽셀을 중심으로 한 히트맵 방식으로 현업의 관점과 유사점을 발견할 수는 있었다.

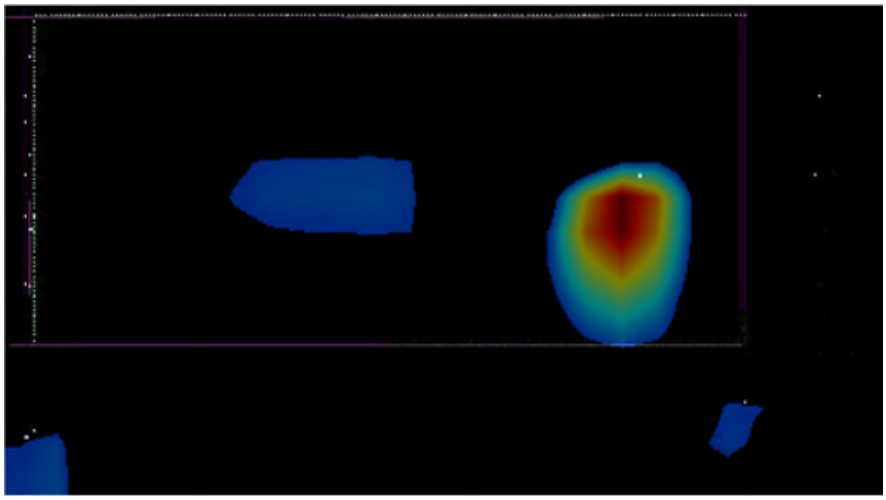
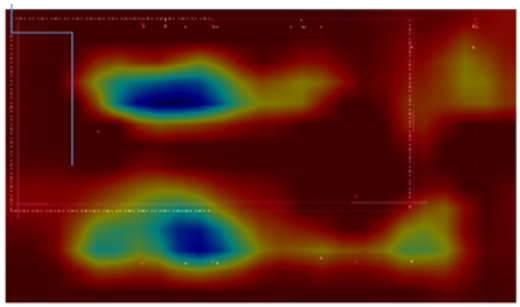


그림 18: CAM을 활용한 이미지 특징 추출

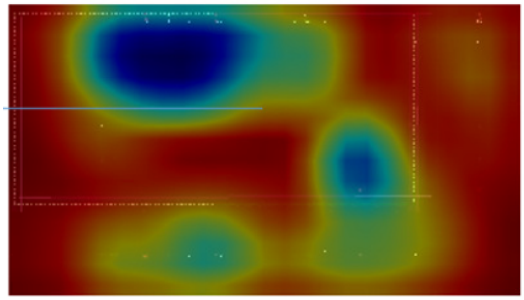
위 그림은 아일랜드형 결함의 특징을 잘 잡아낸 예시이다. 최종 결과 판단 유무의 예측에 대한 설명력을 얻는 데는 도움이 될 수 있었다. 추후 주기별로 수집되는 데이터를 추가하여 재학습시에 이미지의 증량 등에 활용할 수 있을 것이다. 그럼에도 불구하고, 역시 LIME과 마찬가지로 출력 계층의 특징만을 보기에 추후 재 학습을 할 때 각 클래스 별로 부족한 부분들을 보완하는 등의 활용성은 있겠지만, 즉시 모델의 성능 개선을 위해서는 한계를 보였다.

4.5.2 Grad-CAM

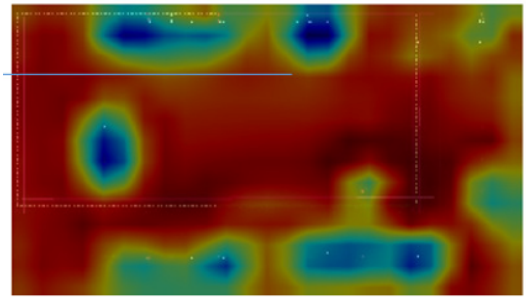
CNN의 앞쪽 레이어에서 이러한 특징을 추출하고, 뒤쪽 레이어에서 그 특징들을 가지고 이미지를 분류한다.



(a)



(b)



(c)

그림 19: Grad-CAM 결과 (a) Block3 (b) Block4 (b) Block5

여기까지의 학습은 앞쪽 레이어를 통과시켜 높은 성능의 모델을 만들었고, 그 결과로 추출된 층별 특징을 위의 그림과 같이 Grad-CAM으로 확인했다. VGG 모델이 학습한 이미지넷의 데이터 세트는 이 실험의 데이터 세트와는 여러가지 측면에서 매우 다른 형태로 컨볼루션 레이어의 많은 부분을 활용할 수 없을 것으로 생각할 수 있지만, Block3 정도까지의 특징이 각 결함의 형상을 잘 포착하는 것으로 보였고, Block4 이후로는 과적합이 의심된다. 따라서 이후의 실험은 Block3 까지의 특징을 최대한 활용할 수 있도록 진행하였다.

4.6 미세 조정에 따른 모델링 결과

기존의 학습 방식으로는 전이학습을 활용한다 하더라도, 다른 이미지에서 학습한 내용을 확인하기 어렵기 때문에, 모든 레이어의 trainable 세팅을 조절해가면서, 최적의 레이어를 찾아야 했을 것이다. 이는 하이퍼파라미터 튜닝보다도 더 고통스러운 작업이 될 수 있다. 하지만, 이 논문에서는 Grad-CAM을 통해, 각 레이어별 특징을 히트맵으로 확인함으로써, 고민없이 새로운 데이터에 적합한 레이어를 찾을 수 있었다.

표 6: VGG-16 Block1-3 freezing 모델의 학습 결과

Optimizer	Loss	검증 정확도	평가 정확도	F1 score
SGD	0.361	0.920	0.694	0.676
Momentum	**0.327**	**0.922**	0.764	0.742
NAG	0.347	0.917	**0.792**	**0.778**

모델이 새 데이터에 수렴되면 기본 모델의 전체 또는 일부를 고정 해제하고 매우 낮은 학습률로 전체 모델을 다시 훈련할 수 있다. 이것은 잠재적으로 점진적 개선을 제공할 수 있는 선택적 마지막 단계이지만 또한

빠른 과적합으로 이어질 수 있다. 기본 모델을 고정 해제하고 낮은 학습률로 전체 모델을 종단 간 학습하는 미세조정을 진행했다. 10 에폭의 학습 후 미세 조정을 통해 개선 효과를 얻을 수 있었다. 미세조정으로 학습한 결과는 다음과 같다.

표 7: 최종 모델의 미세조정 결과

성능지표	Loss	검증 정확도	평가 정확도	F1 score
결과	0.347	0.917	**0.792**	**0.778**

VGG-16 네트워크의 Block3까지 동결하여 가장 좋은 성능을 찾은 다음, 기존 네트워크의 가중치로 동결된 Block1-Block3의 네트워크에 미세한 가중치 업데이트를 시킴으로써 이 실험의 데이터에 좀 더 최적화시키는 미세 조정을 수행하여, 최종적으로 0.79와 0.78의 평가 정확도와 F1 점수를 확인했다.

4.7 성능 평가

각각의 기법은 앞에서 설명한 바와 같이 각 논문에서 설명된 방식으로 적용하여 특징 맵을 시각화하였다. 그 결과에 대한 해석을 기반으로, trainable Layer를 조정함으로써, VGG-16 네트워크의 성능을 유지하면서 특징 맵의 변화를 확인할 수 있었다. 이 실험에서는 Grad-CAM의 결과를 해석하여, 레이어 튜닝(Layer Tuning)의 수준을 결정한 후, 해당 모델을 미세 조정하여 최적의 성능에 도달했다.

또한, 뒤에서 자세히 논하겠지만, 각 블럭 단위로 추가적인 학습을 통해, Grad-CAM의 결과에 대한 해석이 적중했음을 성능지표를 통해 검증했다. 물론, 다양한 성능 개선의 방법이 있기 때문에, 예를 들어, 드롭

아웃[61]같은 간단한 앙상블 기법을 추가해서, 좀 더 나은 성능의 확보가 가능할 수도 있을 것이다. 하지만, 여기서는 이전의 데이터에서 확보한 학습 정보를 최대한 활용하면서, 또한 그 최대한의 지점을 찾는데 있어서 절차적 편의성을 찾는데 목적이 있다.

표 8: 실험 결과(정확도)

Architecture	학습	검증	평가
Fine Tuning	0.998	0.917	**0.792**
Transfer Learning (Block3)	0.999	0.919	0.778
fixed Feature Extraction	0.850	0.814	0.653
Randomize weight	0.996	0.908	**0.569**
LeNet-5	0.995	0.816	0.625

따라서, 단순 CNN 네트워크인 LeNet-5를 활용한 모델링으로 80%의 검증 정확도에 60%의 평가 정확도의 성능에 머물렀다면, 대량의 데이터를 학습한 VGG 네트워크로 90%의 검증 정확도와 0.7의 평가 정확도에서, 설명가능한 인공지능 기술을 활용하여 그 결과를 해석했고, 그 해석을 기반으로 한 최적의 특징 레이어를 찾고, 80%(0.79)의 평가 정확도를 달성했다. 또한, 나머지 구역별로 전이 학습을 실시하는 추가적인 실험을 통해 그 수준을 찾는데 있어서 해석의 효과성을 적절히 검증했다. 검증에 대한 자세한 논의는 다음 절에서 확인해 볼 것이다.

4.8 결과 분석

전반적인 시각화의 성능 개선에 대한 효용성을 확인하기 위해 다른 Block의 레이어 튜닝도 함께 실험하였다. Grad-CAM의 해석력을 확인하기 위해 각 Block을 기준으로 앞부분을 동결하고 뒷부분을 학습했다.

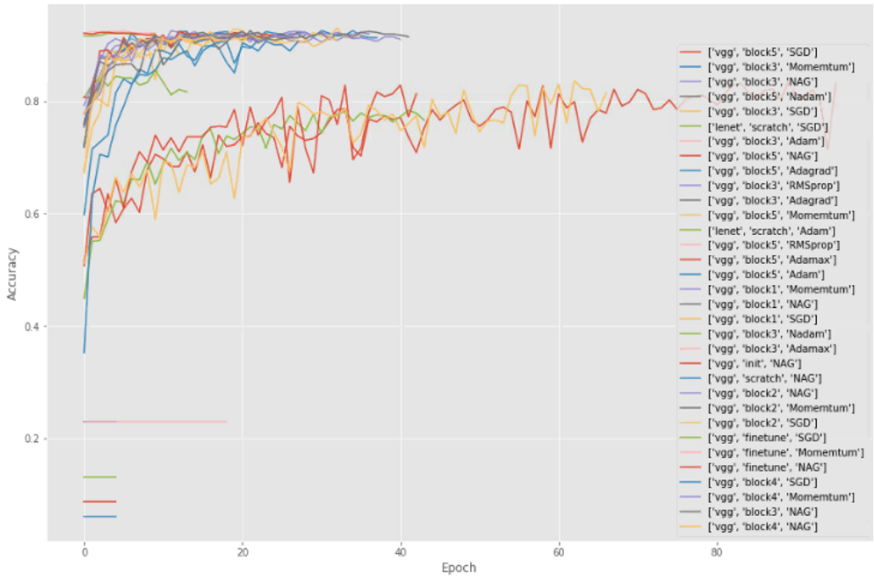


그림 20: 실험 결과(전체 모델의 훈련 추이)

위 그림에서 좌측 하단의 짧은 에폭에서 끝나는 정확도의 변화가 없는 모델들의 경우, 모두 Adam, RMSProp 등 어댑티브 방식의 최적화 기법이다. 세밀한 손실 업데이트가 필요한 미세조정이 아닌 네트워크를 처음부터 학습시키는 경우에도, 위와 같이, 어댑티브 방식의 최적화 기법으로 수렴이 잘 되지 않았다. 앞서 관련 연구에서 밝힌 바와 같이 모멘텀 방식의 최적화 기법들로 학습하여, 좋은 결과를 얻어낼 수 있었다. 특별한 고민없이 Adam 최적화를 자주 써왔지만, 데이터와 모델에 따라 지금도 여전히 SGD와 모멘텀 방식은 여전히 훌륭한 방식임을 확인하였다.

또한, 위 그래프의 중간에 위치한 추이들은 대부분 특징 추출 모델들로 오랜 학습 시간이 필요하고, 낮은 성능에 머물 수 밖에 없었고, 상단의 Block3까지 동결한 모델은 40 에폭 내에서 안정적으로 수렴하고 최적의 성능을 보였지만, 상단의 짧은 경향성들은 나머지 구간의 동결 모델들로

20에서 30 에폭 내에서 빠르게 수렴했으나, 상대적으로 낮은 성능을 보였다. 레이어 튜닝의 과정으로 동결 구간별로 나누어 학습한 결과는 다음과 같다.

표 9: 동결 구간에 따른 성능 비교

VGG-16 Layer Updated	SGD	Momentum	NAG
Block1-5	0.639	0.653	0.625
Block1-4	0.639	0.722	0.653
Block1-3	0.736	**0.778**	**0.778**
Block1-2	0.764	0.708	0.764
Block1	0.694	0.736	0.764

미세조정과 전이학습을 기반으로 하여 최초 모델링에서 평가 정확도의 경우, 20% 가량의 성능 향상을 확보할 수 있었다. IML 기술을 활용함으로써, 모델의 관심 영역을 추적하고 레이어의 재사용 수준을 결정하여 성능 향상까지 확인할 수 있었다.

자연 이미지에 대해 훈련된 많은 심층 신경망은 공통적으로, 첫 번째 레이어에서 가버(Gabor) 필터 및 색상 부분과 유사한 기능을 학습한다. 이러한 첫 번째 계층 기능은 특정 데이터 세트에 한정되지 않고 많은 데이터 세트와 작업에 적용할 수 있다. 대량 데이터로 미리 학습된 네트워크로 초기화하면 대상 데이터 세트에 대한 미세 조정 후에도 지속되는 일반화를 향상시킬 수 있다 [62]. 앞 절에서 보여준 Grad-CAM의 층별 시각화를 확인하여, 이미지넷의 데이터 그룹과 실험 데이터의 그룹은 이미지의 여러가지 특성이 매우 다름에도 불구하고, 이러한 전이학습 기술을 기반으로 한 Block3 까지의 특징 추출은 매우 유효함을 추출된 특징으로 모델의 설명력을 확인해 볼 수 있었다.

결론적으로, 특징을 활용한 반복적인 학습의 효율성을 개선하기 위

해, 데이터의 가공이나 투입없이, 기존 모델의 일부 계층을 재사용하는 전이 학습의 계층 재사용(Layer Reuse) 방법을 적용하였다. 전이 학습은 특징 학습(Feature Learning)과 학습 파라미터(Training Parameter)는 유지하고, 미세 조정을 통한 분류 학습(Classifier Learning)을 수행한다 [63][64]. Grad-CAM의 특징 맵을 해석한 결과를 미세 조정 설계에 활용하여 freezing Layer의 범위를 결정하여 최적의 성능에 도달했고, 그 검증을 위해 다른 범위의 동결 구간을 실험함으로써 적합성을 확인하였다.

물론, 이 연구의 실험 결과를 볼 때, 개별 블럭 단위의 정확도 차이가 크지 않기 때문에, 다른 레벨에서 아마도 부가적으로 정밀한 미세 조정[2] 기술을 활용하여 정확도를 더 개선할 수 있을 것이다. 또한, VGG-16의 계층을 더 늘리거나 또는 줄이는 방법[3]도 모델의 성능을 향상시키는 것이 충분히 가능할 것이다. 그러나, 제조 현장은 여러가지 요인이 제품의 품질에 영향을 크게 끼치게 되므로, 정교하게 설계한 모델이 얼마나 오래 활용이 될 지는 미지수다. 한두달 이후에 투입된 새로운 평가 데이터는 더이상 기대 이상의 성능을 유지하기 어려운게 현실이다. 추가로 수집된 데이터들을 학습시킬 때, 기존에 학습된 모델의 계층별 특징맵을 확인하여, 재학습의 수준을 빠르게 결정하고 부가적인 미세 조정의 여러 기법들을 적절하게 활용할 수 있다면, 모델 개발의 생산성을 높이는데 기여할 수 있다고 생각한다.

[2] 최적화, 학습률, 배치 사이즈, 가중치 감소, 정규화 및 드롭아웃 등
[3] VGG-19, VGG-12 등

제 5 장

결론

많은 연구들이 ILSVRC에서 주어진 대규모 시각적 데이터베이스인 이미지넷을 활용하여 딥러닝의 효과성을 증명하고, 모델의 손실을 감소시켜 정확도를 향상시키기 위해 최적의 모델 학습 기술들에 집중해왔다. 또한, 캐글(Kaggle)과 여러 블로그에서조차 ILSVRC의 우승 모델을 활용하여 개와 고양이의 신규 이미지를 분류하는 모델의 성능을 향상시키고자 하는 과제들을 쉽게 찾을 수 있고, 실제로 그 결과들은 매우 훌륭하다. 그러나, 제조 현장에서 수집되는 시각적 데이터들은 이미지넷의 그것과는 매우 다른 성격을 가진다. 따라서, 현장의 과제들은 기존의 성공적인 모델들을 활용하나 그 일반화 과정에서 많은 어려움을 겪고 있다. 이 연구는 실제 현장에서 수집된 이미지 데이터를 효율적으로 분류해내기 위해 합성곱 신경망과 특히, ILSVRC에서 좋은 성능을 보여준 VGG 모델을 활용하여, 조금 더 간결한 절차로 정확도를 개선할 수 있는 방법을 알아보고, 그러한 기술들을 활용한 실험을 실시한 후, 그에 대한 성과를 확인했다.

5.1 고찰

이 논문에서는 실제 데이터 샘플을 입력하고, 간단한 CNN 네트워크인 LeNet-5를 학습하는 것으로 실험을 시작했다. 그리고, 더 나은 정확도에 도달하기 위해, 대용량 데이터로 사전에 잘 훈련된 모델인 VGG-16의 네트워크와 모델을 활용하여, 밑바닥 학습부터 가중치 초기화를 적용하는

방법을 통해 모델의 성능을 일정 정도 개선했다. 그런 다음, 더욱 정교한 정확도의 개선을 위해 전이 학습 기반의 특징 모델을 학습시켰고, 2장에서 소개한 세가지 IML 기법들을 활용하여 실험 대상 모델의 예측을 들여다 보았다. LIME은 앞선 연구들에서 지역적인 해석을 위해 샘플링 과정의 반복으로 인한, 설명의 불안정성이 우려되었듯이 [25][13], 현장의 비즈니스적 시각과는 해석력에서 큰 차이를 보였다. CAM은 마지막 계층의 결과만을 들여다 보는 한계가 있었지만, Grad-CAM을 통해 각 컨볼루션 계층의 특징을 시각화함으로써 전이 학습의 수준을 확인할 수 있었고, 이를 통하여 평가 성능지표 또한 훌륭히 개선할 수 있었다. 더불어, Grad-CAM의 해석력을 평가하기 위해 각 Block의 컨볼루션 계층을 동결과 훈련을 나누어 학습하여 실제 가장 좋은 성능의 기준을 Grad-CAM을 통해 찾아냈음을 검증할 수 있었다. 이로써 IML의 슈퍼 벡터 표현 또는 오 탐지 간의 해석을 통해 품질을 향상시키는 정도를 확인하고, 이러한 특징의 해석을 통한 모델의 구현이 새로운 데이터의 정확성을 높인다는 것을 확인했다. 즉, IML의 장점은 해석의 품질을 개선하고 과도하게 적합된 모델로 인해 고통받는 분석가의 작업부하를 줄이는 데 충분한 도움이 될 수 있다는 것이다. 따라서, 이 연구는 해석 가능한 머신 러닝에서 얻은 이점이 고단한 분석가의 작업들에서 미세 조정의 품질을 해결할 수 있음을 보여준다. 결과적으로 이 접근법이 데이터에 과적합할 수 있는 모델을 일반화하는 데 효과적인 것을 알 수 있었다.

5.2 연구 제한

그럼에도 불구하고, 몇 가지 한계는 남아있다. 이 연구를 통해 IML의 효과성은 어느 정도 확인했지만, 기본적인 데이터 보강과 적절한 매개변

수 파악 등의 반복적인 절차는 여전했다. 또한, 현장의 실제 문제에 대한 기술적인 안목과 비즈니스적인 전문성이 필요하기 때문에, 인적 편차에 대한 우려도 남아있는 것이 현실이다. 서론에서 강조했듯이, 아직까지 설명가능한 인공지능 기술들은 모델의 관점에서 설명을 제공하는 것이 아니라 인간의 관점에서 해석이 가능한 특징을 시각적으로 나타내는 것으로, 그 해석에 기반하여 자동적으로 데이터를 수집한다거나 재학습을 하는 부분에 있어 작업자의 개입은 필수적이다.

5.3 향후 계획

본 논문은 IML을 활용하여 실제 제조 현장에서 발생한 결함 이미지의 패턴을 분류하고 그 특징을 추출하여, VGG-16의 정확도를 유지하면서 모델이 의사결정을 하는데 있어서 주요하게 바라보는 관점이 현업의 그것과 유사하게 해석가능함을 확인하였다. 특히, Grad-CAM의 특징 맵을 해석하여, 전이 학습을 통한 미세 조정 과정에서 Freezing Layer와 Training Layer를 설계하는 데에 도움을 줌으로써, 성능 향상의 가능성도 충분히 확인하였다. 따라서, 요즘 분석 솔루션의 핵심 기술이 된 자동화된 머신러닝(AutoML, Automated Machine Learning)을 적용할 때, 자동학습으로 제시된 초모수들의 최적화된 결과에 대하여 전이학습과 미세조정의 근거와 검증으로 활용성이 매우 높다. 그러나, 해석 이후에 수퍼 카테고리의 데이터를 (+)보강하여 모델의 정확도를 높이고, 오탐 카테고리의 데이터를 (-)보강하여 데이터의 편향성을 제거하는 반복적인 절차는 여전히 분석가들에게 과제로 남아있다. 향후 연구에서는 이러한 분석가들의 작업 부하를 줄이면서 모델의 일반화 오류를 줄일 수 있는 해석의 질(Quality)을 양(Quantity)적으로 평가하기 위한 기법에 대해 탐구할 계획이다.

참고 문헌

[1] M. T. Ribeiro, S. Singh, and C. Guestrin, "" why should i trust you?" explaining the predictions of any classifier," pp. 1135–1144, 2016.

[2] A. Karpaty, "Cs231n convolutional neural networks for visual recognition course," 2017.

[3] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[4] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*, vol. 1. MIT press Cambridge, 2016.

[5] D. Sarkar, R. Bali, and T. Ghosh, *Hands-On Transfer Learning with Python: Implement advanced deep learning and neural network models using TensorFlow and Keras*. Packt Publishing Ltd, 2018.

[6] I. O. for Standardization, *Accuracy (Trueness and Precision) of Measurement Methods and Results-Part 2: Basic Method for the Determination of Repeatability and Reproducibility of a Standard Measurement Method*. International Organization for Standardization, 1994.

[7] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[8] G. Calzoni, "Transcendence: Wally pfister," *Cineforum*, no. 536, pp. 35–36, 2014.

[9] K. Gödel, "Über formal unentscheidbare sätze der principia mathematica und verwandter systeme i," *Monatshefte für mathematik und physik*, vol. 38, no. 1, pp. 173–198, 1931.

[10] A. H. Maslow, "A dynamic theory of human motivation.," 1958.

[11] B. Burke, D. Cearley, N. Jones, D. Smith, A. Chandrasekaran, C. Lu, and K. Panetta, "Gartner top 10 strategic technology trends for 2020-smarter with gartner," 2019.

[12] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (xai)," *IEEE Access*, vol. 6, pp. 52138–52160, 2018.

[13] C. Molnar, "Interpretable machine learning," *Lulu. com*, 2019.

[14] W. J. Murdoch, C. Singh, K. Kumbier, R. Abbasi-Asl, and B. Yu, "Interpretable machine learning: definitions, methods, and applications," *arXiv preprint arXiv:1901.04592*, 2019.

[15] S. Russell and P. Norvig, "Artificial intelligence: a modern approach," 2002.

[16] D. Castelvecchi, "Can we open the black box of ai?," *Nature News*, vol. 538, no. 7623, p. 20, 2016.

[17] D. Gunning, "Explainable artificial intelligence (xai)," *Defense Advanced Research Projects Agency (DARPA), nd Web*, vol. 2, no. 2, 2017.

[18] P. Hall, "On the art and science of machine learning explanations," *arXiv preprint arXiv:1810.02909*, 2018.

[19] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A survey of methods for explaining black box models," *ACM computing surveys (CSUR)*, vol. 51, no. 5, pp. 1–42, 2018.

[20] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," *arXiv preprint arXiv:1702.08608*, 2017.

[21] D. Doran, S. Schulz, and T. R. Besold, "What does explainable ai really mean? a new conceptualization of perspectives," *arXiv preprint arXiv:1710.00794*, 2017.

[22] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, *et al.*, "Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai," *Information Fusion*, vol. 58, pp. 82–115, 2020.

[23] F. Santosa and W. W. Symes, "Linear inversion of band-limited reflection seismograms," *SIAM Journal on Scientific and Statistical Computing*, vol. 7, no. 4, pp. 1307–1330, 1986.

[24] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[25] D. Alvarez-Melis and T. S. Jaakkola, "On the robustness of interpretability methods," *arXiv preprint arXiv:1806.08049*, 2018.

[26] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2921–2929, 2016.

[27] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.

[28] M. V. Valueva, N. Nagornov, P. A. Lyakhov, G. V. Valuev, and N. I. Chervyakov, "Application of the residue number system to reduce hardware costs of the convolutional neural network implementation," *Mathematics and Computers in Simulation*, vol. 177, pp. 232–243, 2020.

[29] G. T. Toussaint, "The use of context in pattern recognition," *Pattern Recognition*, vol. 10, no. 3, pp. 189–204, 1978.

[30] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 11, pp. 3212–3232, 2019.

[31] P. Druzhkov and V. Kustikova, "A survey of deep learning methods and software tools for image classification and object detection," *Pattern Recognition and Image Analysis*, vol. 26, no. 1, pp. 9–15, 2016.

[32] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

[33] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[34] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

[35] H.-C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. M. Summers, "Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning," *IEEE transactions on medical imaging*, vol. 35, no. 5, pp. 1285–1298, 2016.

[36] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, 2020.

[37] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 12, pp. 2935–2947, 2017.

[38] Y. Guo, H. Shi, A. Kumar, K. Grauman, T. Rosing, and R. Feris, "Spottune: transfer learning through adaptive fine-tuning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4805–4814, 2019.

[39] H. B. Curry, "The method of steepest descent for non-linear minimization problems," *Quarterly of Applied Mathematics*, vol. 2, no. 3, pp. 258–261, 1944.

[40] S. Sra, S. Nowozin, and S. J. Wright, *Optimization for machine learning*. Mit Press, 2012.

[41] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[42] N. Qian, "On the momentum term in gradient descent learning algorithms," *Neural networks*, vol. 12, no. 1, pp. 145–151, 1999.

[43] Y. Nesterov, "A method for unconstrained convex minimization problem with the rate of convergence o (1/kˆ 2)," in *Doklady an ussr*, vol. 269, pp. 543–547, 1983.

[44] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *International conference on machine learning*, pp. 1139–1147, 2013.

[45] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization.," *Journal of machine learning research*, vol. 12, no. 7, 2011.

[46] G. Hinton, N. Srivastava, and K. Swersky, "Lecture 6e: Neural networks for machine learning," *Coursera: Neural Networks for Machine Learning*, vol. 4, 2014.

[47] T. Tieleman and G. Hinton, "Divide the gradient by a running average of its recent magnitude. coursera: Neural networks for machine learning," *Technical Report.*, 2017.

[48] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[49] T. Dozat, "Incorporating nesterov momentum into adam," 2016.

[50] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.

[51] N. S. Keskar and R. Socher, "Improving generalization performance by switching from adam to sgd," *arXiv preprint arXiv:1712.07628*, 2017.

[52] C. Xing, D. Arpit, C. Tsirigotis, and Y. Bengio, "A walk with sgd," *arXiv preprint arXiv:1802.08770*, 2018.

[53] Y. Wang, *A Generalized Framework for Machine Re-learning of Complex Process and Kinetic Models*. PhD thesis, 2019.

[54] S.-S. Lee and T.-S. Jang, "Understanding of laser-based ultrasonics," *Journal of the Korean Society for Nondestructive Testing*, vol. 22, no. 1, pp. 74–87, 2002.

[55] H.-D. Jeong, H.-J. Shin, and J. L. Rose, "Detection of defects in a thin steel plate using ultrasonic guided wave," *Journal of the Korean Society for Nondestructive Testing*, vol. 18, no. 6, pp. 445–454, 1998.

[56] S. Behnke, *Hierarchical neural networks for image interpretation*, vol. 2766. Springer, 2003.

[57] P. Y. Simard, D. Steinkraus, J. C. Platt, *et al.*, "Best practices for convolutional neural networks applied to visual document analysis.," in *Icdar*, vol. 3, 2003.

[58] S. V. Stehman, "Selecting and interpreting measures of thematic classification accuracy," *Remote sensing of Environment*, vol. 62, no. 1, pp. 77–89, 1997.

[59] C. Van Rijsbergen, "Information retrieval: theory and practice," in *Proceedings of the Joint IBM/University of Newcastle upon Tyne Seminar on Data Base Systems*, pp. 1–14, 1979.

[60] J. Opitz and S. Burst, "Macro f1 and macro f1," *arXiv preprint arXiv:1911.03347*, 2019.

[61] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[62] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," in *Advances in neural information processing systems*, pp. 3320–3328, 2014.

[63] W. Rawat and Z. Wang, "Deep convolutional neural networks for image classification: A comprehensive review," *Neural computation*, vol. 29, no. 9, pp. 2352–2449, 2017.

[64] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Ieee, 2009.

Abstract

The improving accuracy of classifying an image using interpretable machine learning (IML)

Seonghyeon Kim
Graduate School of Practical Engineering
Seoul National University

Artificial Intelligence illustrate not the causality, but the highest probability of the pattern in data unless the data has a bias because many algorithms depend on the data. This paper aims to tackle human interpretability focused on the trade-off between model complexity and reliability. This study first emphasized several theoretical perspectives among IML methods taxonomy. Based on the comprehension of IML approaches, it highlighted the implementation of modeling using real data sample. The research results give better understanding of generalization error in the interpretation of Machine Learning. Consequently, the interpretation of a model is to clarify the bias of the analysts whether data has a wrong answer or question.

Keywords : IML, XAI, visualization, interpretability, transfer learning
Student Number : 2019-22523