

제조영상 불량 검출 및 분류

인공지능 학습모델 개발



저자 1 서 지 훈

저자 2 김 무 영

지도교수 감 진 규

목 차

| | |
|------------------------|----|
| 1. 서론 | 1 |
| 1.1. 연구 배경 | 1 |
| 1.2. 연구 목표 | 1 |
| 2. 연구 배경 | 1 |
| 2.1. 데이터 수집 | 1 |
| 2.2. 데이터 분석 | 2 |
| 2.3. 개발 환경 | 3 |
| 3. 연구 내용 | 4 |
| 3.1. 데이터 전처리 | 4 |
| 3.2. 모델 생성 | 5 |
| 3.3. 모델 학습 | 6 |
| 3.4. 데이터 증강 및 학습 | 7 |
| 4. 결과 분석 및 평가 | 8 |
| 5. 결론 및 향후 연구 방향 | 9 |
| 6. 개발 일정 및 역할 분담 | 10 |
| 6.1. 개발 일정 | 10 |
| 6.2. 역할 분담 | 11 |
| 7. 참고 문헌 | 11 |

1. 서론

1.1. 연구 배경

최근 롤마크(알파벳으로 표시된 원산지·제조사)를 위조해 국산으로 둔갑한 중국산 철근이 잇달아 발견되면서 수입산 부적합 철강재에 대한 우려가 커지고 있다. 실제로 대한제강의 위조 롤마크가 찍혀 유통된 철근의 경우 제품의 중량이 기준치 대비 13%나 미달해 건설공사에 쓸 수 없는 불량 철근이었다[1]. 불량 철근이 건설에 적용될 경우 건축물의 안정성에 심각한 영향을 미칠 수 있다. 예를 들어, 2014년 경주 마우나 리조트 붕괴 사고는 불량 철강 자재가 원인으로 지목된 사고였다[2].

1.2. 연구 목표

한국은 세계 철강생산 6위에 달하는 많은 양의 철강을 생산하고 있다[3]. 이렇게 많은 철강의 결함을 눈으로 모두 탐지하기에는 큰 어려움이 따르므로, 본 연구에서는 철강 표면의 결함을 검출 및 분류하는 인공지능 모델을 설계하고자 했다. 하지만 이미 이러한 모델들이 존재함을 확인했기에[4], 우리는 모델을 개선하여 기존 모델보다 더 우수한 성능을 보여주는 모델을 개발하고자 한다.

2. 연구 배경

2.1. 데이터 수집

본 연구에는 러시아 최대의 철강 기업 Severstal가 Kaggle에서 개최한 대회 Kaggle: Steel Defect Detection의 철강 표면 결함 데이터[5]를 사용했다. 이 대회에서 Severstal의 데이터베이스에 존재하는 실제 철강 데이터가 제공되었다.

2.2. 데이터 분석

데이터 세트는 5,506개의 테스트 데이터, 12,586개의 학습 데이터, train.csv 파일로 구성된다. train.csv 파일은 학습 데이터에 대한 정보를 담고 있는데, 결함에 대한 ImageId, ClassId, EncodedPixels의 세 가지 정보를 얻을 수 있다. ImageId는 결함이 존재하는 이미지들의 이름을 의미한다. ImageId에는 동일한 이름을 가지는 데이터가 있으므로 여러 개의 결함을 가지는 이미지가 존재한다. 또한 전체 학습 데이터 개수는 12,568개이지만 ImageId의 중복되지 않는 이미지의 이름은 6,666개이므로 결함이 존재하지 않는 이미지 또한 존재한다. ClassId는 결함의 종류를 1~4로 나타내며, EncodedPixels는 RLE 형식으로 인코딩 된 결함의 위치를 의미한다. 아래의 그림 1, 2는 train.csv를 분석하여 그래프로 표현한 형태이다.

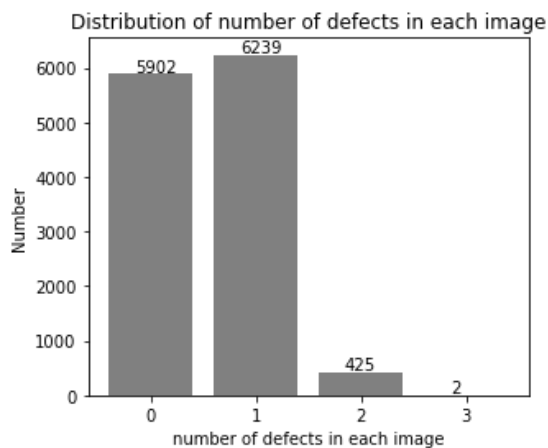


그림 1-결함 개수에 따른 이미지 개수

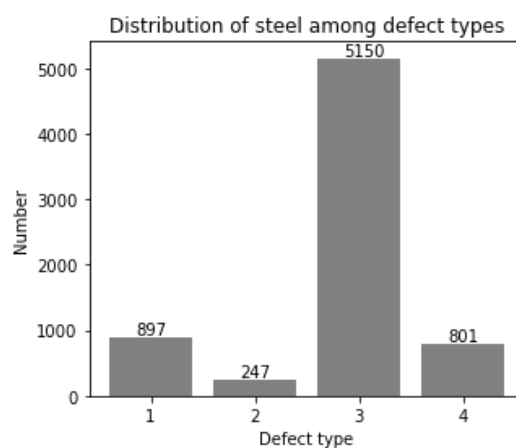


그림 2-결함 종류에 따른 이미지 개수

그림 1은 결함 개수에 따른 데이터 수를 나타낸 그래프로, 결함이 0개인 이미지는 5,902개, 결함이 1개의 이미지는 6,239개, 결함이 2개의 이미지는 425개, 결함이 3개인 이미지는 2개가 존재한다. 단순 CNN으로는 여러 개의 결함을 가진 이미지의 결함을 완벽하게 분류할 수 없으므로, 한 이미지를 여러 영역으로 분할하여 클래스로 분류하는, image segmentation을 지원하는 모델이 필요하다. 그림 2는 결함 종류에 따른 데이터 개수를 나타낸 그래프로, 네 종류의 결함은 각각 897, 247, 5150, 801개가 존재한다. 3번 결함의 수가 다른 세 클래스를 모두 합친 것의 세 배에 가까울 정도로 많으므로, 이 데이터 세트는 불균형이 크다.

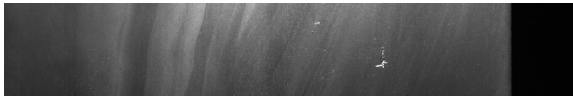


그림 3 – 데이터 세트 이미지의 예시



그림 4 – 철강 부분이 거의 존재하지 않는 예시

데이터 세트 이미지의 크기는 모두 256 x 1600으로 동일하다. 하지만 이미지의 크기가 통상적인 이미지에 비해 매우 큰 편이어서 학습이 오래 걸린다. 그림 3의 이미지는 철강이 대부분을 차지하고 있다. 반면, 그림 4는 철강이 거의 관측되지 않은 이미지이다. 이러한 이미지는 크기에 비해서 얻을 수 있는 정보가 매우 작다. 따라서 전처리 과정에서 이미지의 크기를 조절할 필요가 있다.

2.3. 개발 환경

개발 언어는 Python 3.6.9를 사용했다. OS는 Ubuntu 20.04.5 LTS를, CPU는 Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz를, GPU는 NVIDIA Corporation GP102 TITAN Xp 7대를 이용했다. 라이브러리는 TensorFlow 2.4.0을 사용하였으며, 이를 위하여 CUDA 11.0과 cuDNN 8.0을 사용했다.

3. 연구 내용

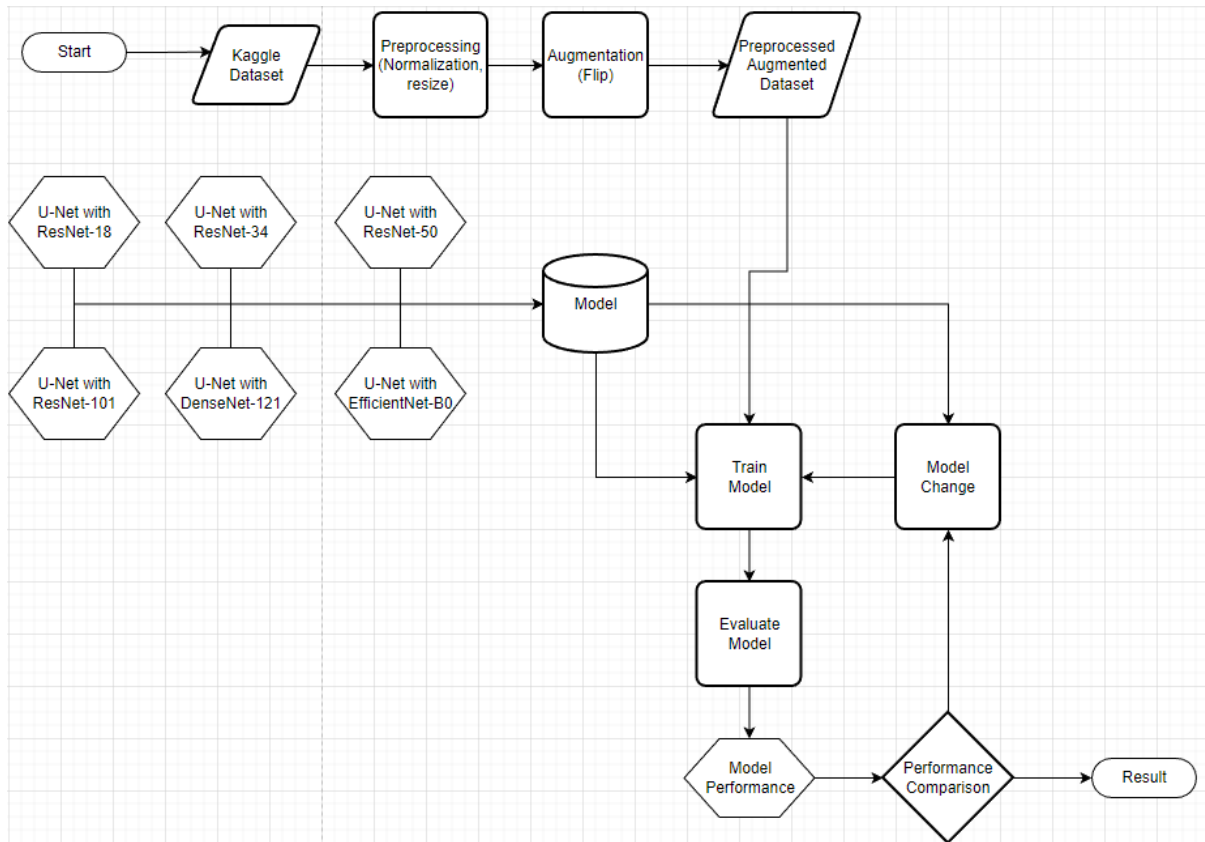


그림 5 – 연구의 전체적인 흐름도

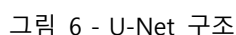
연구의 전체적인 흐름은 위의 그림 5와 같다.

3.1. 데이터 전처리

전처리는 Rescaling(픽셀 값 재조정) 및 Resize(이미지 크기 조정)를 사용했다. 픽셀은 0~255 사이의 값을 갖는데 이 차이가 너무 커서 딥러닝 모델의 학습할 때 개선되는 속도가 느리다. 따라서 픽셀 값이 0~1 사이의 값을 갖도록 정규화를 해준다.

데이터 세트의 이미지는 모두 256 x 1600으로 크기가 동일하지만, 이미지의 크기가 매우 커 학습에 걸리는 시간이 크다. 또한 그림 4와 같이 철강이 일부분만 촬영된 이미지의 경우 의미 없는 흑백 부분이 계산 과정에 포함된다. 또한 대부분의 CNN 모델은 224 x 224 크기의 작은 이미지에서 최적화되도록 개발되어, 큰 이미지보다 작은 이미지에서 더 우수한 성능[6]을 보인다. 따라서 이를 원본 이미지를 크게 훼손하지 않는 범위인 128 x 800으로 조정하여 학습 속도 개선을 시도하였다.

그림 1에서 알 수 있듯이, 사용하기로 한 데이터에는 다중 결함을 갖는 이미지가 존재한다. 이러한 이미지는 단순 CNN으로만 구성된 모델만을 사용하면 여러 개의 결함을 한 번에 분류해내는 게 어렵다. 그래서 이를 위해 Image Segmentation이 가능한 모델이 필요하고 본 연구에서는 Image Segmentation 모델 중 U-Net[7]을 사용하기로 했다.



하지만 그림 6에서 알 수 있듯, U-Net의 인코더에서는 이미지의 특징을 추출하는 convolution block이 단계마다 두 개씩 총 10개가 존재한다.



5

3.3. 모델 학습

U-Net과 함께 사용할 CNN 모델은 많은 종류가 있으므로 이를 다양하게 사용하여 성능을 비교하기로 하였다. 이때 과적합을 방지하기 위하여 학습 에폭(epoch)은 30으로 통일하였고, 학습시킬 데이터의 배치 사이즈(batch size)는 32로 설정하였다.

focal loss는 그림 4와 같이 학습에 이용되지 않을, 결함이 없는 부분이 대다수를 차지하는 데이터가 학습에 미치는 영향을 줄이기 위해 만들어진 손실 함수이다. tversky loss는 본 연구에 사용된 데이터 세트처럼 클래스 간의 데이터 수 불균형이 심한 경우, 수가 적은 클래스를 잘못 예측한 경우에 더 높은 손실을 주어 클래스 불균형의 영향을 줄이는 손실 함수이다. focal tversky loss[10]는 focal loss에 tversky loss를 적용하여 만들어진 손실 함수로, 이 두 손실함수의 특성을 모두 가지고 있다. 우리는 이 focal tversky loss를 손실 함수로 사용한다.

dice coefficient (주사위 계수)는 image segmentation에서 가장 널리 사용되는 평가 지표이다. 그림 8과 같이, 이 함수는 두 영역의 겹치는 정도의 두 배를 두 영역의 넓이로 나누어 계산된다. 이 dice coefficient를 성능 척도로 사용한다.

$$\text{Dice} = \frac{2|A \cap B|}{|A| + |B|}.$$

그림 8 – dice coefficient의 계산

우선 전처리와 증강을 시도하지 않고, 손실 함수 또한 가장 기본적인 이진 교차 엔트로피를 사용하였다. 백본 네트워크로는 ResNet-18을 사용하였다. 이렇게 설계된 모델을 학습시킨 후, 전처리를 적용한 모델과 비교해보았다.

| Preprocessing | Binary cross entropy | Dice coefficient |
|---------------|----------------------|------------------|
| None | 0.0139 | 0.4419 |
| Rescaling | 0.0138 | 0.4560 |

표 1 – U-Net with ResNet-18의 데이터 전처리 전후의 모델의 성능 비교

표 1은 데이터 전처리의 적용 여부에 따른 모델의 성능을 보여준다. 표 1에서 데이터 전처리는 모델의 성능을 증가시킬 수 있음을 확인하였다.

이후, 전처리를 적용하여 손실 함수를 이진 교차 엔트로피에서 focal tversky loss로 변경시켜 모델의 성능을 비교하였다.

| Loss function | Dice coefficient |
|---------------------------|------------------|
| Binary cross entropy | 0.4560 |
| Focal <u>tversky</u> loss | 0.4929 |

표 2 – U-Net with ResNet-18 모델의 손실함수에 따른 모델의 성능 비교

표 2는 손실함수의 변화에 따른 모델의 성능을 보여준다. 표 2에서 focal tversky loss를 사용하면 더 좋은 성능을 얻을 수 있음을 확인할 수 있었다. 하지만 아직 dice coefficient의 결과가 너무 낮는데, 이는 데이터의 대부분이 결함이 없는 픽셀과 결함 3으로 이루어져 결함 1, 2, 4에 대한 예측이 잘 이루어지지 않기 때문이라고 판단했다. 따라서 데이터 증강을 적용하기로 하였다.

3.4. 데이터 증강 및 학습

앞서 데이터 분석에서 언급했듯이, 데이터의 불균형을 데이터 증강을 통해 해결하고자 한다. 이 데이터 세트에서 결함은 여러 개가 될 수도, 매우 작을 수도 있으므로 기본 증강인 Crop과 같은 기법도 이미지의 큰 변형이 될 수 있다. 따라서 이미지의 원형을 충분히 유추할 수 있는 기본 대칭만 증강으로 사용하였다. 아래 그림 9, 10, 11은 원본 이미지와 증강을 가한 이미지의 차이점을 보여준다.



그림 9 – 증강을 가하지 않은 원본 이미지

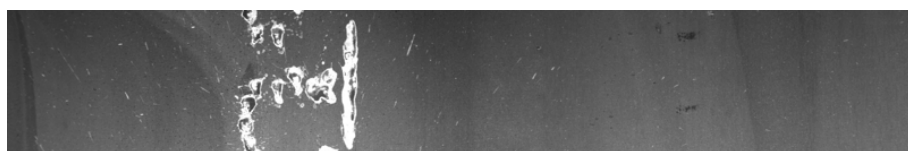


그림 10 – 좌우 대칭을 가한 이미지

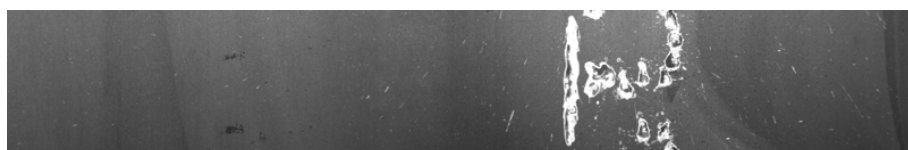


그림 11 – 상하 대칭을 가한 이미지

데이터 증강을 적용하고, U-Net의 백본 네트워크에 사용된 모델을 바꿔가며 실험한 결과는 아래의 표 3과 같다.

| Model | Focal <u>tversky</u> loss | Dice coefficient |
|-----------------|---------------------------|------------------|
| ResNet-18 | 0.4940 | 0.6049 |
| ResNet-34 | 0.4971 | 0.6008 |
| ResNet-50 | 0.5487 | 0.5425 |
| ResNet-101 | 0.5849 | 0.4686 |
| DenseNet-121 | 0.5410 | 0.5474 |
| EfficientNet-B0 | 0.4874 | 0.5909 |

표 3 – 데이터 증강 후 U-Net의 백본 네트워크 모델의 변경에 따른 성능 비교

표 2와 표 3에서, ResNet-18은 데이터 증강 후 기존보다 성능이 개선되었음을 확인할 수 있다. 또한, 백본 네트워크로 사용된 CNN 모델 중 가장 간단한 모델인 ResNet-18이 가장 높은 dice coefficient를 얻었음을 볼 수 있다.

4. 결과 분석 및 평가

아래 그림 12는 원본 이미지의 결함 위치와 실제로 예측된 결함의 위치를 나타낸 것이다.



그림 12 – 실제 결함의 위치와 예측된 결함의 위치

실제 결함을 상당한 정확도로 찾아내고 있음을 알 수 있다. 또한 연구에 사용된 CNN 모델 중 가장 작은 규모의 네트워크를 가진 ResNet-18 모델을 사용했을 때 가장 높은 dice coefficient를 얻었으므로, U-Net 성능 개선을 위해 백본 네트워크의 규모를 증가시키는 것이 반드시 전체 모델의 성능을 증가시키지 않음을 알 수 있다.

5. 결론 및 향후 연구 방향

철강의 불량률을 줄이기 위해 철강의 표면 결함을 탐지하는 딥러닝 모델을 설계하는 것을 목표로 하였다. 이를 위해 Severstal 의 철강 데이터를 정규화 및 resize 하여 전처리하였고, 모델을 학습시켰다. 하지만, 데이터 세트의 불균형으로 인해 만족할 만한 성능이 나오지 않았고, 이를 개선하기 위해 데이터 기본 증강을 적용하여 불균형을 해소하려고 하였다. 그 결과 dice coefficient 가 0.4929 에서 0.6049 로 향상된 것을 볼 수 있었다. 또한, U-Net 의 백본 네트워크에 다양한 모델을 적용해 실험한 결과, ResNet-18 이 0.6049 로 가장 성능이 좋았다.

향후에는 더 우수한 백본 네트워크를 사용하는 대신 GAN 모델과 같이 강력한 데이터 증강 기법을 사용함으로써, 더 우수한 성능을 낼 수 있을 것이다. 현재는 철강 표면의 결함 검출에서만 사용되지만 향후에는 의학 분야처럼 필요한 분야에 전반적으로 사용될 수 있을 것으로 생각한다.

6. 개발 일정 및 역할 분담

6.1. 개발 일정

| 5 월 | | 6 월 | | | | 7 월 | | | | 8 월 | | | | 9 월 | | | |
|--------|---------|-----|----------|-----|-----|------------|-----|-----|-----|-------------|-----|-----|------------|-----|-----|-------|-----|
| 3 주 | 4 주 | 1 주 | 2 주 | 3 주 | 4 주 | 1 주 | 2 주 | 3 주 | 4 주 | 1 주 | 2 주 | 3 주 | 4 주 | 1 주 | 2 주 | 3 주 | 4 주 |
| 데이터 분석 | | | | | | | | | | | | | | | | | |
| | 데이터 전처리 | | | | | | | | | | | | | | | | |
| | | | 기본 모델 작성 | | | | | | | | | | | | | | |
| | | | | | | 모델 학습 및 평가 | | | | | | | | | | | |
| | | | | | | 데이터 증강 | | | | | | | | | | | |
| | | | | | | | | | | 모델 재학습 및 평가 | | | | | | | |
| | | | | | | | | | | | | | 모델 변경 및 학습 | | | | |
| | | | | | | | | | | | | | | | | 모델 평가 | |

6.2. 역할 분담

| | |
|-----|--|
| 서지훈 | 철강 결함 검출 모델 초기 형태 개발 모델 학습 파라미터 값 조정 및 검증 완성된 모델 최종 검증 및 최적화 |
| 김무영 | 훈련용 데이터 분석 및 전처리 훈련용 데이터 증강 모델 평가 및 결과 분석 |

7. 참고 문헌

- [1] 장시형 기자, "기준 미달 불량 철근으로 부실공사 우려", 이코노미 조선, Oct 1 2014 [Online].
http://economychosun.com/client/news/view.php?boardName=C05&t_num=7907
- [2] 김윤호 기자, "95만원 아낀 불량 H빔, 경주 참사 불렀다 ", 중앙일보, March 28 2014 [Online]. <https://www.joongang.co.kr/article/14283424#home>
- [3] 윤보람 기자, "철강 호황에 한국 조강 생산 7천만t 회복...올해도 '훈풍' 기대", 연합뉴스, Jan 2 2022 [Online]. <https://www.yna.co.kr/view/AKR20220101041600003>
- [4] Praveen Damacharla, Achuth Rao M. V., Jordan Ringenberg, Ahmad Y Javaid, "TLU-Net: A Deep Learning Approach for Automatic Steel Surface Defect Detection", pp 1-6, Jan 18 2021
- [5] Data. Kaggle:Severstal-steel-defect-detection [Online]. Available:
<https://www.kaggle.com/c/severstal-steel-defect-detection/data>
- [6] Jerubbaal John Luke, Rajkumar Joseph, Mahesh Balaji, "IMPACT OF IMAGE SIZE ON ACCURACY AND GENERALIZATION OF CONVOLUTIONAL NEURAL NETWORKS", PP 70-80, Feb 2019
- [7] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation", pp 01-08, 18 May 2015
- [8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian sun, "Deep Residual Learning for Image Recognition", pp770-778, 10 Dec 2015
- [9] 신석용, 이상훈, 한현호, "딥러닝 기반의 Semantic Segmentation을 위한 Residual U-Net에 관한 연구, pp 251-257, 27 June 2021 (in Korean)
- [10] Nabila Abraham, Naimul Mefraz Khan, "A Novel Focal Tversky loss function with improved Attention U-Net for lesion segmentation", pp 1-5, Oct 18 2018